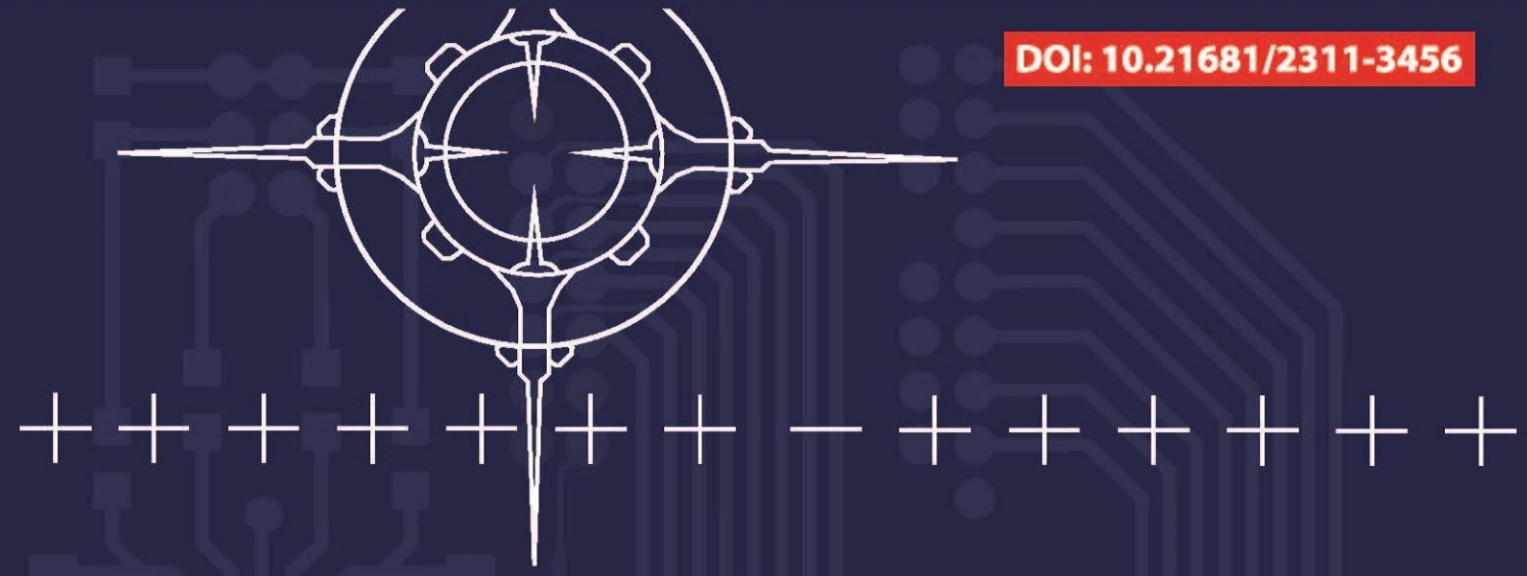
ВОПРОСЫ КИБЕРБЕЗОПАСНОСТИ

№6 2024 (64)

DOI: 10.21681/2311-3456



Управление рисками информационной безопасности

Безопасный искусственный интеллект

Безопасность программных ресурсов

Эшелон
комплексная безопасность

{KOMRAD}
Enterprise SIEM

ВЫСОКАЯ ПРОИЗВОДИТЕЛЬНОСТЬ И МИНИМАЛЬНЫЕ ТРЕБОВАНИЯ К АППАРАТНОМУ ОБЕСПЕЧЕНИЮ

KOMRAD Enterprise SIEM позволяет осуществлять централизованный сбор событий ИБ, выявлять инциденты ИБ и оперативно на них реагировать. Применение комплекса позволяет эффективно выполнять требования, предъявляемые регуляторами к защите персональных данных, к обеспечению безопасности государственных информационных систем и контролю критической информационной инфраструктуры предприятия. KOMRAD позволяет отправлять данные о событиях и инцидентах ИБ во внешние системы (например, ГосСОПКА).

- Визуальный конструктор запросов и директив корреляции
- Высокая производительность
- Гибкая интеграция с нестандартными источкниками событий
- Широкий спектр поддержки источников событий
- Ролевая модель управления доступом
- Оперативное оповещение об инциденте
- Масштабируемость

Чтобы получить демо-версию KOMRAD Enterprise SIEM или заказать пилот у наших партнеров в вашем регионе, свяжитесь с нашим отделом продаж по e-mail: sales@npo-echelon.ru.

ВОПРОСЫ КИБЕРБЕЗОПАСНОСТИ

НАУЧНЫЙ РЕЦЕНЗИРУЕМЫЙ ЖУРНАЛ

№6 (64) 2024 г.

Выходит 6 раз в год

Журнал выходит с 2013 г. (Свидетельство о регистрации ПИ № ФС77-75239). Перерегистрировано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций 07.03.2019.

Журнал входит в рейтинг научных изданий ВАК в категории К1, индексируется в RSCI, публикует статьи по специальностям 1.2.4 и 2.3.6 – физ-мат. науки; 2.2.15, 2.3.1, 2.3.5, 2.3.6 – техн.науки

Учредитель и издатель

АО «Научно-производственное объединение «Эшелон»

Над номером работали:

Г. И. Макаренко – *шеф-редактор*, И. Ю. Шахалов – *отв. секретарь*, С. С. Игнатов – *верстка*, Ю.С. Логинова – *зам. шеф-редактора*

Подписано к печати 20.12.2024 г.
Общий тираж 120 экз. Цена свободная

Адрес: 107023, Москва, ул. Электрозаводская, д. 24, стр. 1.
E-mail: editor@cyberrus.info, тел.: +7 (985) 939-75-01.
Требования, предъявляемые к рукописям, размещены на сайте: **https://cyberrus.info/**

СОДЕРЖАНИЕ

Подписка на журнал осуществляется в почтовых отделениях по каталогу «Пресса России». Подписной индекс 40707

МЕТОДИЧЕСКИЕ ПОЛОЖЕНИЯ ПО ВЕРОЯТНОСТНОМУ ПРОГНОЗИРОВАНИЮ КАЧЕСТВА ФУНКЦИОНИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ. Часть 2. МОДЕЛИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ «ЧЕРНЫХ ЯЩИКОВ»

Костогрызов А. И.[1], Нистратов А. А.[2], Голосов П. Е.[3]

DOI: 10.21681/2311-3456-2024-6-2-27

Продолжение[4]

Цель всей работы: *помочь системным аналитикам, участвующим в оценке качества функционирования информационных систем (ИС) при их создании, эксплуатации, модернизации, развитии, сформировать облик комплексной методики вероятностного прогнозирования, применимого в интересах обеспечения качества и безопасности, обоснования допустимых рисков, выявления существенных угроз и поддержки принятия научно обоснованных системных решений для упреждающего противодействия угрозам в жизненном цикле ИС.*

Цель 2-й части работы: *детализировать в интересах вероятностного анализа свойств, характеризующих качество функционирования ИС, общие методические положения, укрупненно изложенные в 1-й части статьи, путем предложения вероятностных моделей, представимых в виде «черных ящиков».*

Методы исследования *включают: методы теории вероятностей, методы системного анализа. В качестве моделируемой системы формально выступают «черные ящики», когда известны исходные данные для моделирования и выходные результаты, но неизвестно внутреннее устройство системы. Получаемые результаты математического моделирования используются в интерпретации к исходной ИС, в интересах которой проводятся соответствующие расчеты.*

Результаты 2-й части работы: *предложены модели, представимые в виде «черных ящиков», для вероятностного анализа составных свойств качества ИС согласно ГОСТ Р 59341-2021 «Системная инженерия. Защита информации в процессе управления информацией системы».*

Научная новизна работы: *предлагаемые модели ориентированы на достижение общей цели функционирования ИС различного функционального приложения (сформулированной в 1-й части статьи) – обеспечения надежности и своевременности предоставления необходимой информации, полноты, достоверности и безопасности используемой информации для последующего применения по назначению. Использование моделей позволяет осуществлять оценки по единой вероятностной шкале качества функционирования рассматриваемых системы и ее составных элементов, представимых в виде «черных ящиков».*

Ключевые слова: *вероятность, модель, прогнозирование, риск, система, системный анализ, угроза.*

1. Введение

Методические положения настоящей части статьи предназначены для вероятностного прогнозирования качества функционирования рассматриваемой ИС (далее по тексту – «Системы» с заглавной буквы) с использованием понятия «моделируемой системы». Под «моделируемой системой» понимается система, для которой решение задач системного анализа осуществляется с использованием ее формализованной модели и, при необходимости, формализованных моделей учитываемых сущностей в условиях их применения. Например, при проведении системного анализа в принимаемых допущениях, ограничениях и предположениях модель может формально описывать процесс, функциональные действия, множество активов и/или выходных результатов или множество этих или иных сущностей в их целенаправленном применении в задаваемых условиях.

В 1-й части статьи обоснована актуальность работы, предложен общий подход к вероятностному прогнозированию качества функционирования ИС, основанный на использовании моделей и методов ГОСТ Р 59341. Подход представлен в виде основных методических положений, раскрывающих базовые термины и определения, рассматриваемые объекты

1 Костогрызов Андрей Иванович, доктор технических наук, профессор. Федеральный исследовательский центр «Информатика и управление» Российской академии наук. Москва, Россия. E-mail: Akostogr@gmail.com

2 Нистратов Андрей Андреевич, кандидат технических наук. Федеральный исследовательский центр «Информатика и управление» Российской академии наук. Москва, Россия. E-mail: Andrey.nistratov@gmail.com

3 Голосов Павел Евгеньевич, кандидат технических наук, директор Института общественных наук Российской академии народного хозяйства и государственной службы (РАНХиГС). Москва. Россия. E-mail: pgolosov@gmail.com

4 Часть 1 настоящей работы опубликована в журнале «Правовая информатика», 2024, №3, с. 13–31.

ИС, цель и задачи прогнозирования, принятые предположения, условия и допущения, оцениваемые показатели, перечень вероятностных моделей, порядок проведения моделирования, интерпретация и системный анализ результатов расчетов [1–24]. Тем самым по-крупному описан облик комплексной методики вероятностного прогнозирования качества функционирования ИС, который подлежит наполнению моделями и методами.

В настоящей 2-й части работы общие методические положения 1-й части детализированы путем предложения вероятностных моделей, позволяющих проведение исследований «моделируемых систем» в виде «черного ящика». Это: «Модели для оценки надежности предоставления информации и выполнения операций»; «Модели для оценки своевременности предоставления информации и выполнения операций»; «Модели для оценки полноты используемой информации»; «Модели для оценки актуальности используемой информации»; «Модели для оценки безошибочности информации после контроля»; «Модели для оценки корректности информации после обработки»; «Модели для оценки безошибочности действий пользователей и персонала»; «Модели для оценки защищенности системы от опасных программно-технических воздействий»; «Модели для оценки защищенности активов от несанкционированного доступа»; «Модели для оценки конфиденциальности используемой информации». Также приводится «Метод использования универсальной вспомогательной модели показателя для определения исходных данных в расчетах», используемый при формировании исходных данных.

Примечание. Детализированный в статье перечень не исчерпывает всего множества существующих вероятностных моделей и методов, практически приемлемых для достижения поставленных целей в анализе качества функционирования ИС.

Тем самым представленные модели охватывают практическое воплощение идеи оценки качества функционирования ИС – см. рис. 1 и пояснения в 1-й части статьи. Во 2-й части статьи также приведены некоторые примеры, иллюстрирующие варианты применения предложенных моделей [1–12, 15–30].

2. Вероятностные модели

2.1. Общие положения по использованию «черных ящиков»

За основу предлагаемого подхода к математическому моделированию с использованием «черных ящиков» принят подход, изложенный в разные годы в приложении к различным системам [2–12, 15–24] и доведенный до реализации на уровне ГОСТ Р 59341. С учетом неопределенностей расчет вероятностных показателей делается в принятых предположениях, условиях и допущениях, описанных в 1-й части статьи.

Предлагаемые математические модели ориентированы на противодействие возможным угрозам качеству функционирования Системы. Перечень возможных разнородных угроз может включать:

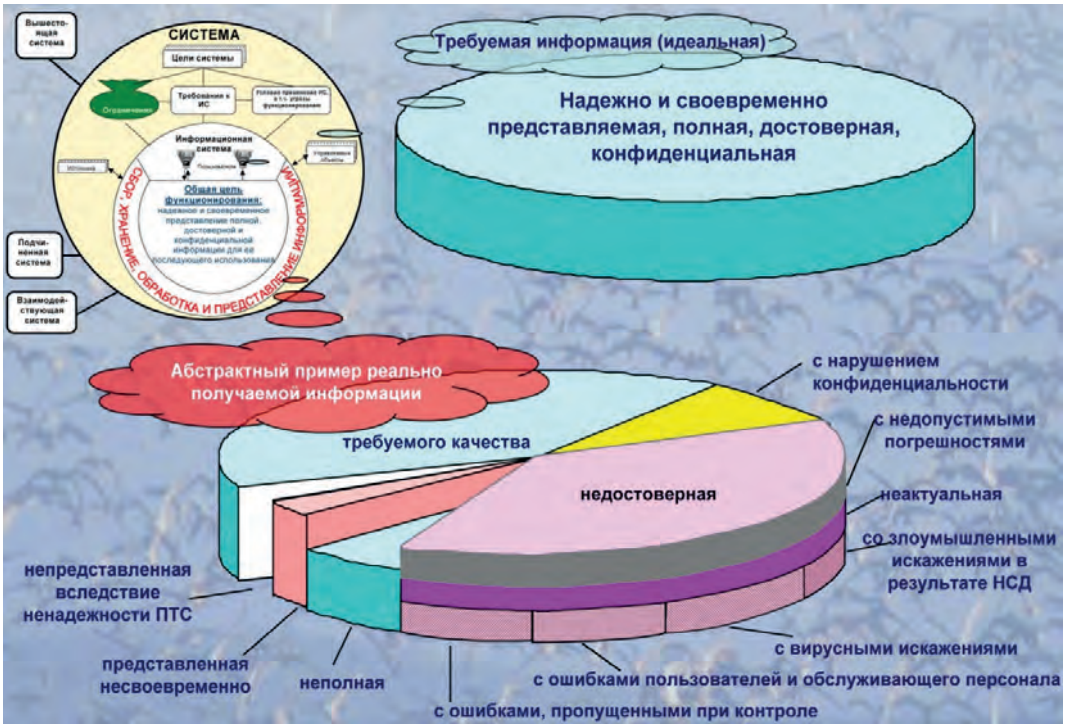


Рис. 1. Абстрактная иллюстрация качества функционирования ИС

- природные и природно-техногенные угрозы – по ГОСТ Р ИСО 13381-1, ГОСТ Р ИСО 17359, ГОСТ Р ИСО/МЭК 27002, ГОСТ Р 51901.1, ГОСТ Р 54124, ГОСТ Р МЭК 61069-5, ГОСТ Р МЭК 61069-6, ГОСТ Р МЭК 61069-7;
- угрозы со стороны человеческого фактора – по ГОСТ Р МЭК 62508;
- угрозы безопасности информации, качеству программного обеспечения, безопасности оборудования и коммуникаций, используемых в процессе работы – по ГОСТ Р ИСО/МЭК 15026, ГОСТ Р ИСО/МЭК 16085, ГОСТ Р ИСО/МЭК 27002, ГОСТ Р 51275, ГОСТ Р 51583, ГОСТ Р 54124, ГОСТ Р 56939, ГОСТ Р 58412, ГОСТ Р 59329 – ГОСТ Р 59357, ГОСТ Р 59989 – ГОСТ Р 59994;
- угрозы возникновения ущерба репутации и/или потери доверия к конкретному заказчику или поставщику, системы которого были скомпрометированы;
- прочие соответствующие угрозы качеству функционирования Системы.

Для оценки риска нарушения качества функционирования моделируемой системы (подсистемы, элемента), представимой как «черный ящик», необходимо учитывать, что в общем случае существует зависимость от надежности и своевременности предоставления используемой информации и выполнения операций, полноты оперативного отражения в системе новых объектов и явлений, актуальности обновляемой информации, безошибочности информации после контроля, корректности обработки информации, безошибочности действий пользователей и персонала системы, обеспечения безопасности информации – см. рис. 1. При этом понятие обеспечения безопасности информации включает сохранение целостности моделируемой системы в условиях опасных программно-технических воздействий, защищенность активов от несанкционированного доступа и сохранение конфиденциальности используемой информации. Под целостностью моделируемой системы понимается такое состояние моделируемой системы, которое отвечает целевому назначению модели системы в течение задаваемого периода прогноза. Полное формализованное описание всех предлагаемых моделей приведено в ГОСТ Р 59341, а ранее – в [2–12, 15–24]. По этой причине для оценки предлагаемых показателей качества функционирования моделируемой системы (подсистемы, элемента) в представляемых ниже в 2.2.–2.12. описаниях и примерах перечисляются лишь необходимые исходные данные. В отдельных случаях даются иллюстративные примеры и методические рекомендации с указанием соответствующих ссылок. Общий порядок проведения моделирования, интерпретации и системного анализа получаемых результатов расчетов приведен в 1-й части статьи.

2.2. Модели для оценки надежности предоставления информации и выполнения операций

Под надежностью предоставления информации в системе и выполнения операций понимается

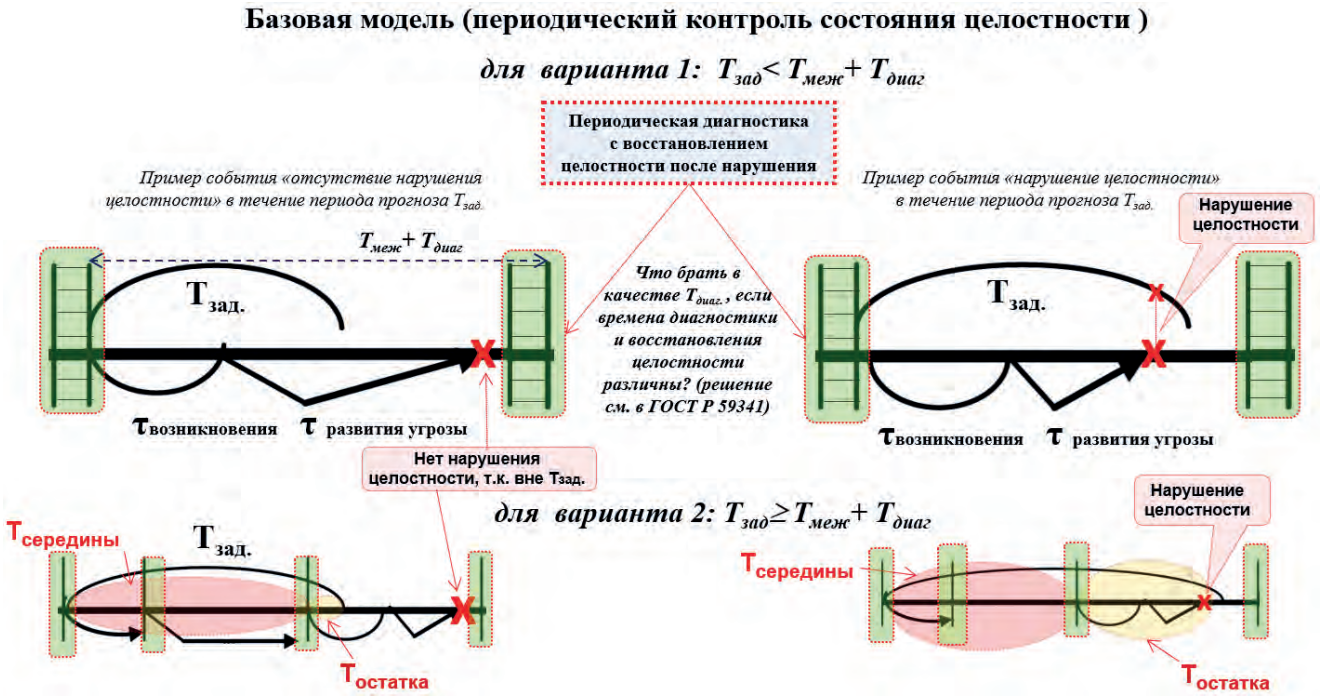


Рис. 2. Примеры элементарных событий, связанных с нарушением целостности моделируемой системы по ГОСТ Р 59341, приложению В.3.2

свойство системы обеспечивать прием, автоматическую обработку запроса или команды и предоставление или принудительную выдачу выходной информации согласно функциональному алгоритму при соблюдении эксплуатационных условий применения и технического обслуживания системы.

Для оценки надежности предоставления информации и выполнения операций в моделируемой системе (подсистеме, элементе) в течение заданного периода прогноза применяются «Модели для оценки надежности предоставления информации» из ГОСТ Р 59341, приложения В.3.2. Примеры рекомендуемой базовой модели в части элементарных событий, связанных с нарушением целостности моделируемой системы, и с привязкой к обозначениям исходных данных в ГОСТ Р 59341 представлены на рис. 2.

В качестве исходных данных для моделирования используются:

σ – частота возникновения источников угроз (например, ведущих к отказам и сбоям программно-технических и технологических средств);

β – среднее время развития угроз (активизации источников угроз) с момента их возникновения до нарушения целостности моделируемой системы;

$T_{\text{меж}}$ – среднее время между окончанием предыдущей и началом очередной диагностики целостности моделируемой системы;

$T_{\text{диаг}}$ – среднее время системной диагностики целостности моделируемой системы;

$T_{\text{восст}}$ – среднее время восстановления нарушаемой целостности моделируемой системы;

$T_{\text{зад}}$ – задаваемая длительность периода прогноза.

В результате моделирования рассчитываются частные показатели: вероятность $P_{\text{над предст}}(T_{\text{зад}})$ надежного предоставления информации и выполнения операций в системе в течение заданного периода прогноза $T_{\text{зад}}$ и вероятностный показатель надежности предоставления информации и выполнения операций $Z_{\text{над предст}}(T_{\text{зад}})$, учитывающий рассчитываемое значение $P_{\text{над предст}}(T_{\text{зад}})$ и соответствующие условия α из ГОСТ Р 59341, приложения В.3.2. Условие α касается надежности предоставления запрашиваемой или выдаваемой принудительно информации и формулируется в виде ограничений:

$$P_{\text{над предст}}(T_{\text{зад}}) \geq P_{\text{доп над}}(T_{\text{зад}}),$$

и возможный ущерб от нарушения не превышает допустимого уровня (это – формулировка условия α). Учет результатов моделирования в оценках интегрального риска осуществляют с использованием индикаторного показателя надежности предоставления информации $Z_{\text{над предст}}(T_{\text{зад}})$

$$Z_{\text{над предст}}(T_{\text{зад}}) = \begin{cases} 1, & \text{если условие надежности предоставления информации } \alpha \text{ выполнено,} \\ P_{\text{над предст}}(T_{\text{зад}}), & \text{если условие } \alpha \text{ не выполнено или не задано.} \end{cases} \quad (1)$$

Сопоставление с возможным ущербом позволяет рассматривать дополнение до единицы этого показателя $[1 - Z_{\text{над предст}}(T_{\text{зад}})]$ в качестве вероятностного выражения риска нарушения надежности предоставления запрашиваемой или выдаваемой принудительно информации в системе. Его равенство нулю при несущественном ущербе означает пренебрежимо малый риск.

Некоторые возможности применения модели продемонстрированы в 3-й части статьи. С целью избегания трудностей с формированием исходных данных для моделирования ниже излагается метод использования универсальной вспомогательной модели показателя элементарных состояний эксплуатируемой системы.

2.3. Метод использования универсальной вспомогательной модели показателя для определения исходных данных в расчетах

Для расчета разных по смысловому пониманию показателей могут быть использованы одни и те же математические модели. Например, упомянутые выше в 2.2. «Модели для оценки надежности...» из ГОСТ Р 59341, приложения В.3.2, применимы для анализа безошибочности действий пользователей и персонала системы, а также для анализа защищенности системы от опасных программно-технических воздействий в течение заданного периода прогноза – см. в том же стандарте приложения В.3.8, В.3.9. В качестве исходных выступают те же данные для моделирования (адаптированные по контексту), обозначаемые одинаковым образом как σ, β, $T_{\text{меж}}$, $T_{\text{диаг}}$, $T_{\text{восст}}$, $T_{\text{зад}}$.

В общем случае, если для таких исходных данных, как $T_{\text{меж}}$ и $T_{\text{диаг}}$ на практике не возникает каких-либо трудностей, а $T_{\text{зад}}$ задается аналитиком, то для определения σ, β, $T_{\text{восст}}$ может возникать вопрос – откуда их брать? В этом случае рекомендуется использование универсальной вспомогательной модели показателя (УВМП) – см., например, ГОСТ Р 59349 «Системная инженерия. Защита информации в процессе системного анализа». Дело в том, что в любой момент времени у ответственных лиц, принимающих решение, должно быть формальное представление о том, какое состояние эксплуатируемой системы «приемлемо», а какое «неприемлемо» и требует управляющей реакции для улучшения. То есть в каждый отчетный момент времени по каждому из критичных показателей (или по их совокупности) можно с однозначной уверенностью определить, что его (их) значения находятся в состоянии, которое может быть

охарактеризовано как «Приемлемое» или «Приемлемое с отклонением» (когда требуются определенные организационные или обычные технические усилия по улучшению значения показателя) или как «Неприемлемое» состояние (когда требуются кардинальные решения по восстановлению условий и/или ресурсов, которые в существующем виде уже не обеспечивают требуемый уровень качества функционирования системы или в ближайшее время при бездействии не будут его гарантировать) – см. рис. 3.

Состояния «Приемлемое», «Приемлемое с отклонением», «Неприемлемое» – это элементарные состояния, в которые может переходить во времени каждый из учитываемых показателей, используемых при моделировании. Под приемлемыми условиями и/или ресурсами системы понимается такое ее состояние (характеризуемое этими условиями и ресурсами), при котором обеспечивается достижение целей функционирования системы. Такое состояние называют состоянием целостности системы.

Рис. 3. Элементарные состояния контролируемого показателя УВМП во времени и временные исходные данные для моделирования

Из фрагмента состояний на рисунке 3 частота возникновения источника угроз для моделируемой системы $\sigma = 1 / [(\tau_{возн.1} + \tau_{возн.2} + \tau_{возн.3} + \tau_{возн.4}) / 4]$, среднее время развития угроз с момента возникновения источника угроз до нарушения нормальных условий функционирования моделируемой системы $\beta = (\tau_{разв.1} + \tau_{разв.2} + \tau_{разв.3} + \tau_{разв.4} + \tau_{разв.5}) / 5$, среднее время восстановления нарушаемой целостности моделируемой системы $T_{восст} = (\tau_{восст.1} + \tau_{восст.2} + \tau_{восст.3}) / 3$, где $\tau_{возн.i}$ – i-й интервал времени между возникновениями источника угроз, $\tau_{разв.j}$ – j-й интервал времени развития угроз с момента возникновения источника угроз до нарушения нормальных условий, $\tau_{восст.m}$ – m-й интервал времени восстановления нарушаемой целостности.

Значения σ, β, $T_{восст}$, получаемые по результатам анализа данных мониторинга (или их пересчета на уровне УВМП), являются исходными данными для формального описания моделируемой системы с учетом возможности прогнозирования динамики разнородных событий. Роль в УВМП каждого из учитываемых критичных показателей сводится к их количественным значениям при формировании значений исходных данных σ, β, $T_{восст}$ для последующего моделирования.

Примечание. Этот способ также применим для случая, когда в качестве критичного показателя выступает неколичественная оценка состояния с градациями «Приемлемое», «Приемлемое с отклонением», «Неприемлемое», что аналогично понятиям «допустимого», «значимого» и «критического» рисков, используемых для экспертных оценок.

Учитывая математическую идентичность моделей для оценки надежности предоставления информации, безошибочности действий пользователей и персонала системы, а также защищенности системы от опасных воздействий в течение заданного периода прогноза, некоторые возможности моделирования приведены в 3-й части статьи с использованием понятий сложной «моделируемой системы».

2.4. Модели для оценки своевременности предоставления информации и выполнения операций

2.4.1. Общее

Под своевременностью предоставления требуемой информации в системе понимается свойство системы обеспечивать предоставление запрашиваемой или выдаваемой принудительно (автоматически) выходной информации в задаваемые сроки, гарантирующие выполнение соответствующей функции согласно целевому назначению системы. Аналогичное определение – в приложении к выполнению операций.

Для оценки своевременности предоставления информации и выполнения операций в моделируемой системе достаточно высокую степень адекватности обеспечивают модели и методы теории массового обслуживания. В реальности могут использоваться различные технологии обслуживания. Например, это бесприоритетное и приоритетное обслуживание одним или несколькими приборами, многофазное обслуживание, обслуживание в режиме разделения времени и т.п. Для оценки некоторых из этих технологий при различного рода ограничениях уже существуют методические разработки, в том числе в приложении к анализу вычислительных систем и сетей. Применительно к системам массового обслуживания с ожиданием термин «технология обслуживания» совпадает с термином «дисциплина обслуживания» или «технология диспетчеризации», определяющим

порядок выборки очередного запроса из буфера для обработки на приборе – см., например, характерные свойства технологий в примере 1-й части статьи, описывающем динамический метод рациональной диспетчеризации запросов различной срочности. Под запросами понимаются не только запросы пользователей на получение выходных документов, но и задачи на пересылку файлов или ввод информации в базу данных (БД), а также некоторые технологические операции по управлению вычислительным процессом, администрированию доступа к передающей среде в компьютерных сетях, обеспечению безопасности информации и пр.

В общем случае процессы предоставления информации и выполнения операций формализуются как процессы массового обслуживания потоков запросов в надежно функционирующих системах с ожиданием и буфером бесконечного объема [3–13, 19–21] – см. рис. 4.

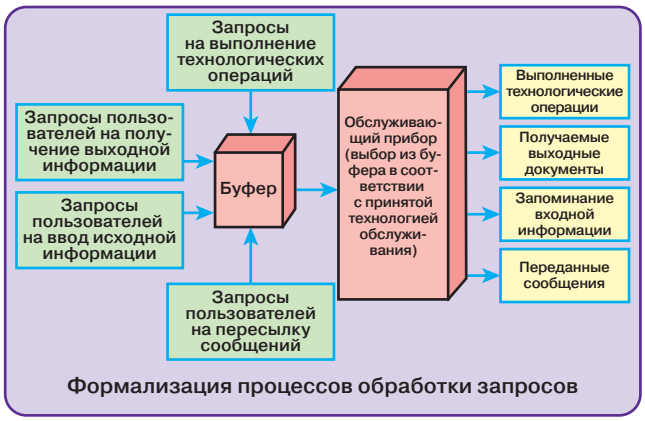


Рис. 4. Пример формального описания процессов обработки запросов

Требования к своевременности обработки запросов определяются формально с помощью следующих критериев.

Определение критерия 1 (оценка по среднему времени обработки). Обработка запросов i–го типа считается выполненной в срок, если среднее время их обработки $T_{\text{полн}.i1}$ с учетом задержек в очередях не превышает заданного уровня $T_{\text{зад}..i}$, т.е. если $T_{\text{полн}.i1} \le T_{\text{зад}..i}$.

Определение критерия 2 (оценка по вероятности своевременной обработки). Обработка запросов i–го типа считается выполненной в срок, если вероятность своевременной обработки $P_{\text{св}\,i}(T_{\text{зад}\,i})$ за время $T_{\text{зад}..i}$ не ниже заданной вероятности $P_{\text{св. зад}.i}$, т.е. $P_{\text{св}\,i}(T_{\text{зад}\,i}) = P(t_{\text{полн}.i} \le T_{\text{зад}..i}) \ge P_{\text{св.зад}.i}$, где случайная величина $t_{\text{полн}.i}$ характеризует полное время обработки запросов i–го типа с учетом задержек в очередях. Критерий 2 задает более жесткие временные рамки и используется для компьютерных систем жесткого реального времени (как правило, при этом $T_{\text{зад}.i} \ge 0{,}8$).

Для каждого из значимых типов обрабатываемой информации с привязкой к выполняемым функциональным задачам, источникам и получателям информации требования к своевременности обработки запросов в системе указываются в форме одного из двух упомянутых выше критериев своевременности 1 или 2.

В результате расчетов оцениваются такие показатели, как вероятность своевременной обработки запросов i-го типа в системе $P_{\text{св}\,i}(T_{\text{зад}\,i})$ и, исходя из них – относительная доля своевременно обработанных в системе запросов $C_{\text{своевр}}$, для которых выполняются требования к своевременности.

Вероятность своевременной обработки запросов определяется с использованием табулируемой неполной гамма-функции:

$$P_{\text{св}\,i}(T_{\text{зад}\,i}) = \int_0^{\theta_i} \exp(-\tau)\,\tau^{\gamma_i}\,d\tau/\Gamma(\gamma_i), \qquad (2)$$

где $\Gamma(\gamma) = \int_0^{\infty} \exp(-\tau)\,\tau^{\gamma}\,d\tau$ – гамма-функция,

$$\gamma_i = \frac{T_i}{\sqrt{|\,T_{i2} - T_i^2\,|}},\ \theta_i = T_{\text{зад}\,i} \cdot \frac{\gamma_i^2}{T_i};$$

γ_i, θ_i – рассчитываемые параметры неполной гамма-функции; T_i, и $\sqrt{|\,T_{i2} - T_i^2\,|}$ – соответственно среднее время и среднеквадратичное отклонение времени реакции системы при обработке запросов i-го типа (т. е. полного времени пребывания на обработке с учетом ожидания в очередях), T_{i2} – второй момент времени реакции. Чаще в качестве исходных данных формируют целиком именно среднеквадратичное отклонение, не опускаясь до отдельных измерений второго момента. В свою очередь, упомянутые значения T_i, и $\sqrt{|\,T_{i2} - T_i^2\,|}$ сами могут быть получены в результате математического моделирования – см., например, приемлемые модели в [2, 4–6, 15–23]. Поскольку полностью детерминированный режим поступления и обработки из рассмотрения исключен, то среднеквадратичные отклонения никогда не обращаются в 0 (т.е. отсутствуют случаи деления на 0).

Относительная доля своевременно обработанных в системе запросов $C_{\text{своевр}}$ охватывает лишь те типы запросов, для которых выполнены требования заказчика, этот показатель вычисляют по формуле

$$C_{\text{своевр}} = \frac{\sum_{i=1}^{I} \lambda_i\, P_{\text{св}\,i}(T_{\text{зад}\,i})[Ind(\alpha_1) + Ind(\alpha_2)]}{\sum_{i=1}^{I} \lambda_i} \qquad (3)$$

где λ_i – частота поступления на обработку запросов i-го типа; критерии α своевременности обработки каждого типа запросов устанавливают с использованием индикаторной функции $Ind(\alpha)$:

$$Ind(\alpha) = \begin{cases} 1, \text{ если условие } \alpha \text{ истинно,} \\ 0, \text{ если условие } \alpha \text{ ложно.} \end{cases}$$

При этом для i-го типа запросов условие своевременности α с учетом возможных ущербов определяют одним из условий α_1 или α_2:

α_1 – условие, когда для i-го типа запросов задан критерий 1 своевременности по среднему времени обработки (реакции системы) и $T_i \le T_{\text{зад } i}$.

α_2 – условие, когда для i-го типа запросов задан вероятностный критерий 2 своевременности и $P_{\text{св } i}(T_{\text{зад } i}) \ge P_{\text{св зад } i}$.

В общем случае расчетные показатели $P_{\text{св } i}(T_{\text{зад } i})$ зависят не только от частоты поступления различных запросов на обслуживание и времени их обслуживания, но и от использования конкретных технологий диспетчеризации (обслуживания) запросов и критериев своевременности (см. пример в 1-й части статьи).

Для оценки интегрального риска рассчитывается $Z_{\text{своевр}}$ – вероятностный показатель своевременности обработки запросов, учитывающий относительную долю своевременно обработанных в системе запросов $C_{\text{своевр}}$, и соответствующие условия α из ГОСТ Р 59341, приложения В.3.3.

2.4.2. Пример для оценки своевременности

Настоящий пример поясняет логику подхода к оценке относительной доли своевременно обработанных запросов лишь тех типов, для которых выполняются требования по своевременности $C_{\text{своевр}}$ согласно ГОСТ Р 59341. Пример призван продемонстрировать извлечение прагматических эффектов от применения предложенного подхода не только для решения корпоративных проблем с использованием ИС, но и для выработки научно обоснованных подходов к решениям задач по оптимизации функционирования и совершенствованию ИС межгосударственного значения. В настоящем примере область исследования охватывает острую проблематику обеспечения доверия к информационному обслуживанию пользователей с использованием распределенных реестров, основанных на блокчейн-технологии. Сами сообщения могут иметь произвольную природу, в частности, иметь технологический, информационный, финансовый, управленческий характер. При этом основным требованием к информационным сообщениям выступает требование наличия биективного отображения с элементом блокчейн-записей (реестровых записей).

Например, в инфраструктуре КНР используется национальная блокчейн-платформа «Xinghuo» («Искра», разработчик – компания Буби)[5], обеспечивающая единую среду взаимодействия государственных и частных организаций. На сегодняшний день она насчитывает около 20 опорных узлов и более 40 региональных блокчейн-хабов, включая отраслевые. Кроме того, сформированы шлюзы для доверенного обмена данными с рядом стран (Казахстан, Филиппины, Малайзия) Пример взаимодействия, основанный на цифровых сертификатах, представлен на рис. 5.

5 Информационное сообщение о платформе: https://wap.cinn.cn/p/303392.html

Рис. 5. Интеграционное решение для реализации международной платформенной торговли Китай-Малайзия

Особенностью построения подобного рода сложных ИС является неотчуждаемость информации от ее создателя, при этом обеспечивается возможность проверки достоверности сведений для произвольного сообщения, в том числе за счет одновременного хранения нескольких его копий на распределенных узлах. Принцип организации блокчейн-сетей не обязательно реализуется на основе подтверждения работы, однако для ряда случаев его применение, несмотря на высокую энергозатратность, является безальтернативным.

Что может быть известно о рассматриваемой Системе опорных узлов и региональных блокчейн-хабов гипотетической информационной среды взаимодействия государственных и частных организаций при ее создании, модернизации и развитии? Стараясь не перегружать статью техническими деталями, полагаем, что известно следующее: по условиям функционирования Системы в период наивысшей нагрузки все множество запросов (технологических, информационных, финансовых, управленческих и др.), связанных с предоставлением необходимой информации и выполнением операций в каждом из опорных узлов и региональных блокчейн-хабов рассматривается как самостоятельная «моделируемая Система» с исходными информационными данными, подразделяется на 4 типа. Это могут быть запросы, каждый из которых может быть обработан в режиме распараллеливания процессов обработки. Для примера, предположим, что запросы 1-го типа ($i = 1$) поступают в среднем через 4 секунды (т.е. частота поступления на обработку запросов 1-го типа $\lambda_1 = 0{,}25$ сек$^{-1}$), 2-го типа – через 5 секунд (т.е. $\lambda_2 = 0{,}2$ сек$^{-1}$), 3-го типа – через 6 секунд ($\lambda_3 = 0{,}167$ сек$^{-1}$), 4-го типа – через 7 секунд ($\lambda_4 = 0{,}143$ сек$^{-1}$). При этом, учитывая повышенные требования к надежности всей информационной среды, задаваемые требованиями бизнес-сообщества, выглядят следующим образом: риск несвоевременной обработки запросов не должен превышать 10^{-6} при том, что предельная длительность обработки запросов 1-го типа не должна превышать 64 секунд, 2-го типа – 96 секунд, 3-го типа – 164 секунд, 4-го типа – 180 секунд. Дополнительно с учетом собираемой статистики положим, что соотношения средних времен обработки запросов 1:2:3:4-го типов в автономном режиме (т.е. без очередей) составляет приблизительно 3:4:6:10.

Главные практические задачи для каждого из опорных узлов и региональных блокчейн-хабов гипотетичной информационной среды взаимодействия государственных и частных организаций (наподобие приведенной на рис. 5 интеграционной среды), подлежащих разрешению с использованием предлагаемой «Модели для оценки своевременности предоставления информации и выполнения операций», формулируются в виде четырех вопросов:

1. Сколько серверов необходимо для своевременной обработки всех запросов, поступающих в «моделируемую Систему»?
2. Как изменятся требования к количеству серверов, необходимых для своевременной обработки запросов в «моделируемой Системе», при смягчении требований к допустимому риску несвоевременной обработки запросов с уровня 10^{-6} до уровня 0.05?
3. Какое количество серверов необходимо иметь для обеспечения гарантированной своевременности обработки запросов при некоторой фиксированной степени распараллеливания процессов в их обработке в «моделируемой Системе» и заданных ограничениях?
4. Как оптимизировать степень распараллеливания процессов при обработке запроса каждого типа в «моделируемой Системе», чтобы избежать излишних затрат и при этом обеспечить требуемую своевременность? (т.е. как определить рациональное число параллельно выполняемых заданий для обработки запроса каждого типа?)

Методические рекомендации по приемлемым подходам к решению сформулированных выше задач с использованием предложенной в 2.4.1 «Модели...» состоят в следующем.

В качестве объектов анализа выступают технологии диспетчеризации и временные задержки при обработке запросов в «моделируемой Системе» (см. 1-ю часть статьи, где показано, что эффекты могут быть достигнуты за счет рациональной динамической настройки параметров технологий диспетчеризации ограничений в специфических условиях ограничений на своевременность обработки запросов различных типов). Каждый из опорных узлов и региональных блокчейн-хабов рассматривается как самостоятельная «моделируемая Система» с исходными информационными потоками, обеспечивающая предоставление необходимой информации и выполнение операций.

Применяя модель 2.4.1., предлагается вариант решения 1-й задачи.

Сначала требования бизнес-сообщества к своевременности обработки запросов переформулируются к виду критерия 2, рекомендуемому ГОСТ Р 59341, а именно:

- время обработки запросов 1-го типа не должно превышать 64 сек. с вероятностью не ниже 0,999999, т.е. задается условие $P(t_{\text{полн.1}} \le 64 \text{ сек.}) \ge 0{,}999999$;
- время обработки запросов 2-го типа не должно превышать 96 сек. с вероятностью не ниже

0,999999, т.е. задается условие $P(t_{\text{полн.2}} \le 96 \text{ сек.}) \ge \ge 0{,}999999$;

- время обработки запросов 3-го типа не должно превышать 164 сек. с вероятностью не ниже 0,999999, т.е. задается условие $P(t_{\text{полн.3}} \le 164 \text{ сек.}) \ge \ge 0{,}999999$;
- время обработки запросов 4-го типа не должно превышать 180 сек. с вероятностью не ниже 0,999999, т.е. задается условие $P(t_{\text{полн.3}} \le 180 \text{ сек.}) \ge \ge 0{,}999999$.

Здесь случайная величина $t_{\text{полн.}i}$ характеризует полное время обработки запросов i–го типа с учетом задержек в очередях, а допустимая вероятность «успеха» 0,999999 получается как дополнение допустимого риска 10^{-6} до 1, т.е. 0,999999 = 1 – 10^{-6}.

Решение задачи будем искать на множестве четырех технологий диспетчеризации, описанных в 1-й части статьи, где разъяснены свойства этих технологий относительно задержек в очередях и возможности извлечения эффектов при решении задачи повышения пропускной способности компьютерной сети путем максимизации своевременно обработанных в системе запросов. Краткая характеристика сравниваемых технологий диспетчеризации: технология 1 заключается в бесприоритетном обслуживании запросов (БПО) в порядке «первый пришел — первый обслужился»; технология 2 заключается в обслуживании запросов с относительными приоритетами (ОП); технология 3 заключается в обслуживании запросов с абсолютными приоритетами с дообслуживанием с прерванного места (АП); технология 4 заключается в пакетном обслуживании запросов с естественным формированием пакетов и относительными приоритетами внутри пакета (Пак.).

Для этих технологий значения среднего времени T_i и среднеквадратичного отклонения времени обработки запросов i-го типа в системе $\sqrt{|T_{i2} - T_i^2|}$ и, соответственно, $P_{\text{св }i}(T_{\text{зад }i})$ по формуле (2) рассчитываются с помощью моделей массового обслуживания, подробно описанных в [2, 4–6, 15–23].

Результаты расчетов показали, что только для 4-й технологии пакетной обработки (Пак.) требования по своевременности обработки всех типов запросов будут выполнены с вероятностью не ниже 0,999999 (i-й тип – это i-й приоритет внутри пакета) – см. рис. 6.

При этом среднее время обработки запросов 1-го типа составит 3.84 сек. (с учетом задержек в очередях) при максимально задаваемой длительности 64 сек., 2-го типа – 4,91 сек. при максимально задаваемой длительности 96 сек., 3-го типа – 6,33 сек. при максимально задаваемой длительности 164 сек., 4-го типа – 8,50 сек. при максимально задаваемой длительности 180 сек. А относительная доля своевременно обработанных в системе запросов, для которых выполнены требования бизнес-сообщества $C_{\text{своевр}}$, составит в процентном выражении более 99,9999 %. Такой эффект достигается, когда исходные средние времена обработки запросов в автономном режиме (т.е. без очередей) составляют: по 1-му типу – не более 0.6 сек. (сравните: с очередями среднее время обработки составит 3,84 сек.); по 2-му типу – не более 0,8 сек. (с очередями 4,91 сек.); по 3-му типу – не более 1,2 сек. (с очередями 6,33 сек.); по 4-му типу – не более 2,0 сек. (с очередями 8,50 сек.)

Таким образом, в результате моделирования принципиально найдено решение 1-й задачи. Т.е. ответ на 1-й вопрос «Сколько серверов необходимо для своевременной обработки всех запросов поступающих в «моделируемую Систему»?» таков: «Серверов необходимо столько, чтобы после распараллеливания процессов времена обработки запросов в автономном режиме составляли: по 1-му типу (приоритету) – не более 0,6 сек.; по 2-му типу – не более 0,8 сек.; по 3-му типу – не более 1,2 сек.; по 4-му типу – не более 2,0 сек. При этом должна применяться технология пакетной обработки запросов с естественным формированием пакетов и относительными приоритетами внутри пакета.

По 2-й задаче (вопрос: Как изменятся требования к количеству серверов, необходимых для своевременной обработки запросов в «моделируемой Системе», при смягчении требований к допустимому риску несвоевременной обработки запросов с уровня 10^{-6} до уровня 0,05? Т.е. смягчение требований к допустимому риску – в 50000 раз!). Для критерия 2, рекомендуемого ГОСТ Р 59341, требования стали такими (вероятность «успеха» 0,95 задается как дополнение до 1 допустимого риска 0.05):

- время обработки запросов 1-го типа не должно превышать 64 сек. с вероятностью не ниже 0,95, т.е. задается условие $P(t_{\text{полн.1}} \le 64 \text{ сек.}) \ge 0{,}95$;

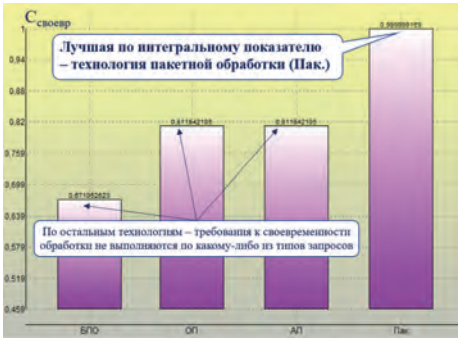


Рис. 6. Относительная доля своевременно обработанных в системе запросов, для которых выполнены требования бизнес-сообщества $C_{\text{своевр}}$ при допустимом риске 10^{-6}

- время обработки запросов 2-го типа не должно превышать 96 сек. с вероятностью не ниже 0,95, т.е. задается условие $P(t_{\text{полн.2}} \le 96 \text{ сек.}) \ge 0{,}95$;
- время обработки запросов 3-го типа не должно превышать 164 сек. с вероятностью не ниже 0,95, т.е. задается условие $P(t_{\text{полн.3}} \le 164 \text{ сек.}) \ge 0{,}95$;
- время обработки запросов 4-го типа не должно превышать 180 сек. с вероятностью не ниже 0,95, т.е. задается условие $P(t_{\text{полн.4}} \le 180 \text{ сек.}) \ge 0{,}95$.

При приблизительно задаваемом соотношении средние времена обработки запросов подбирались так, чтобы хотя бы для одной из рассматриваемых технологий диспетчеризации все видоизмененные требования к своевременности выполнялись.

По результатам расчетов по интегральному показателю $C_{\text{своевр}}$ опять оказывается лучшей технология пакетной обработки (Пак.) – см. рис. 7.

Рис. 7. Относительная доля своевременно обработанных в системе запросов, для которых выполнены требования бизнес-сообщества $C_{\text{своевр}}$ при допустимом риске 0,05

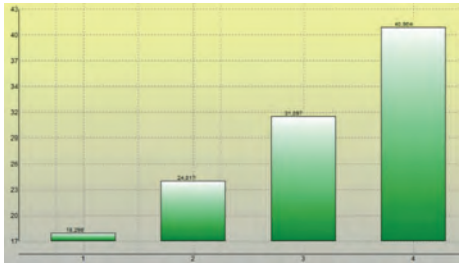


Рис. 8. Средние времена обработки запросов 1–4-го типов в секундах (с учетом очередей)

При этом по сравнению с допустимым риском 10^{-6} с учетом задержек в очередях среднее время обработки запросов 1-го типа составит 18,30 сек. (т.е. возрастет в 4,8 раза), 2-го типа – 24,02 сек. (возрастет в 4,9 раза), 3-го типа – 31,10 сек. (возрастет в 4,9 раза), 4-го типа – 40,96 сек. (возрастет в 4,8 раза) – см. рис. 8. А относительная доля своевременно обработанных в системе запросов, для которых выполнены заданные требования бизнес-сообщества, $C_{\text{своевр}}$ составит в процентном выражении 97,51 %. Такой эффект при решении задачи 2 достигается, когда исходные средние времена обработки запросов в автономном режиме (т.е. без очередей) составляют: по 1-му типу – не более 0,67 сек. (сравните: для задачи 1 – 0,6 сек. при допустимом риске 10^{-6}); по 2-му типу – не более 0,96 сек. (для задачи 1 – 0,8 сек.); по 3-му типу – не более 1,46 сек. (для задачи 1 – 1,2 сек.); по 4-му типу – не более 2,42 сек. (для задачи 1 – 2,0 сек.). Это очень несущественное смягчение требований к производительности серверов при смягчении требований к допустимому риску несвоевременной обработки запросов с уровня 10^{-6} до уровня 0,05, т.е. в 50000 раз (!).

Таким образом, в результате моделирования принципиально найдено решение 2-й задачи. Т.е. ответ на 2-й вопрос таков: «Серверов необходимо столько, чтобы после распараллеливания процессов времена обработки запросов в автономном режиме составляли: по 1-му типу – не более 0,67 сек.; по 2-му типу – не более 0,96 сек.; по 3-му типу – не более 1,46 сек.; по 4-му типу – не более 2,42 сек.». При этом был выявлен важный скрытый эффект: смягчение допустимого риска с высокого уровня 10^{-6} до 0,05 (применимого для обычных ИС организационного типа) приведет лишь к послаблениям по производительности обслуживающих серверов на 10–20 %. Т.е. способ снизить затраты на производительность и количество серверов путем смягчения допустимых рисков несвоевременной обработки запросов до уровня 0,05 является абсолютно бесперспективным.

После найденных решений для 1-й и 2-й задач становятся понятными возможные подходы к решениям по 3-й и 4-й задачам.

Поставленный вопрос по 3-й задаче примера: «Какое количество серверов необходимо иметь для обеспечения гарантированной своевременности обработки запросов при некоторой фиксированной степени распараллеливания процессов в их обработке в «моделируемой Системе» и заданных ограничениях? Предлагаемый подход к решению 3-й задачи таков: «Гарантии должны быть связаны с задаваемой вероятностью своевременной обработки (не менее 0,999999, что эквивалентно допустимому риску 10^{-6}) и применением пакетной технологии диспетчеризации. Для искомого ответа достаточно оценить, сколько серверов при распараллеливании процессов обеспечат временные требования к обработке запросов в автономном режиме, обоснованные выше при исследованиях по вопросам 1 и 2. Если решение приемлемо при существующих ограничениях, следует остановиться, т.е. ответ найден. Если ожидаемые затраты и условия неприемлемы (т.к. производительные серверы – это суть затраты денег, энергии и пр.), это означает, что требования при задаваемых ограничениях невыполнимы, т.е. в принятой постановке вопроса задача не имеет решения».

Поставленный вопрос по 4-й задаче: «Как оптимизировать степень распараллеливания процессов при обработке запроса каждого типа в Системе, чтобы избежать излишних затрат и при этом обеспечить требуемую своевременность?». Понимая, что в общем случае алгоритмическое выполнение программы обработки запросов состоит из составных распараллеливаемых заданий, предлагаемый подход к решению 4-й задачи следующий: «Оптимизация должна заключаться в формальном решении такой постановки задачи: для задаваемой вероятности своевременной обработки не менее 0,999999 определить такое минимальное число составных распараллеливаемых заданий, при котором на выделенном множестве серверов с принятой пакетной технологией диспетчеризации будут обеспечены следующие временные характеристики обработки запросов после распараллеливания (на незагруженных серверах при отсутствии очередей): по 1-му типу (приоритету) – не более 0,6 сек.; по 2-му типу – не более 0,8 сек.; по 3-му типу – не более 1,2 сек.; по 4-му типу – не более 2,0 сек. При этом предполагается наличие иных задаваемых ограничений (стоимостных, технических, ресурсных, климатических и пр.)».

Примечание. Для других технологий диспетчеризации решения рассмотренных в 2.4.2. задач 1–4 будут иными.

2.5. Модели для оценки полноты используемой информации

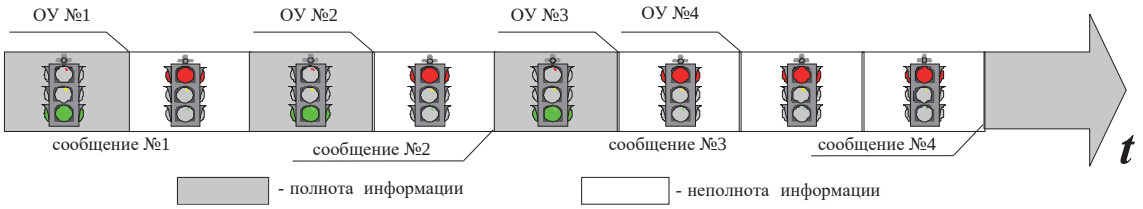
2.5.1. Общее

Под полнотой выходной информации в системе понимается свойство выходной информации отражать состояния всех требуемых объектов учета предметной области системы. Слагается из полноты реализации функций системы, полноты ввода первоначальной информации и полноты оперативного отражения объектов учета в системе.

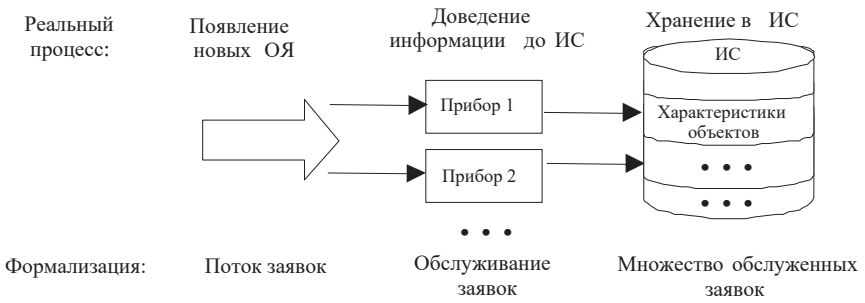
Анализ функционирования ИС показывает, что при решении некоторых задач нередко оказывается необходимым учет множества объектов и явлений, первоначальное возникновение которых в реальности имеет случайный характер. Примерами такого рода задач служат задачи слежения за состоянием местности в условиях чрезвычайной ситуации (например, пожара или наводнения), учета грузов на таможне и др. Выходную информацию будем называть полной, если в ней отражены состояния всех существующих в реальности объектов учета и явлений, необходимых для эффективного выполнения должностными лицами ИС своих функциональных обязанностей. Необходимо отличать полноту представляемой информации от ее достоверности: полнота относится лишь к вновь появляющимся объектам учета и явлениям, а достоверность – как к новым, так и к уже отраженным в ИС. Следовательно, информация может быть полной, но недостоверной.

Сущность влияния неполноты информации на принятие решения состоит в невозможности учета всех объектов учета и явлений (ОУ), характеризующих формальное состояние реальной действительности и влияющих на принимаемые решения. В результате логика принятия решения может оказаться неадекватной в сложившейся ситуации, т.е. решение может оказаться просто неверным – см. рис. 9.

Под полнотой оперативного отражения объектов учета в системе понимается свойство системы



а) Процессы появления новых объектов учета (ОУ) и доведения информации о них до ИС



б) Моделирующая система массового обслуживания M/G/∞.

Рис. 9. Формализация процессов отражения в ИС информации о новых появляющихся объектах учета

отражать требуемые состояния реально существующих объектов учета, в том числе впервые появляющихся в процессе функционирования системы и подлежащих учету в системе согласно ее функциональному назначению. Для оценки полноты оперативного отражения в системе новых объектов и явлений применяется «Модель для оценки полноты оперативного отражения в системе новых объектов и явлений (ОЯ)» из ГОСТ Р 59341, приложения В.3.4. Облик рекомендуемой базовой модели массового обслуживания M/G/∞ с привязкой к обозначениям исходных данных в ГОСТ Р 59341 представлен на рис. 9б. При использовании этой модели отсутствие очереди означает, что все объекты, подлежащие учету, отражены в базе данных ИС.

В качестве исходных данных для моделирования используются:

λ — частота появления новых ОУ в процессе функционирования системы;

$T_{\text{база данных}}$ — среднее время подготовки, передачи и ввода новых ОУ в БД системы.

В результате моделирования рассчитываются частные показатели: вероятность того, что в системе полностью отражены состояния всех реально существующих критичных объектов и явлений $P_{\text{полн}}$ и вероятностный показатель полноты оперативного отражения в системе новых объектов и явлений $Z_{\text{полн}}$, учитывающий эту вероятность и соответствующие условия α из ГОСТ Р 59341, приложения В.3.4.

2.5.2. Пример для оценки полноты

Этот пример демонстрирует подход к оценке полноты оперативного отражения объектов учета в такой ИС, как система дистанционного контроля (СДК) гипотетичной угольной шахты – см. подробнее ГОСТ Р 58494-2019 «Оборудование горно-шахтное. Многофункциональные системы безопасности угольных шахт. Система дистанционного контроля опасных производственных объектов».

Объектами анализа являются информационные сообщения, впервые поступающие в базу данных (БД), и технологии сбора таких данных от источников в режиме реального времени функционирования СДК угольной шахты. Требования заказчика сформулированы следующим образом: должна быть обеспечена полнота отражения информации в СДК обо всех реальных событиях и явлениях, в частности, вероятность того, что в СДК полностью отражены состояния всех реально существующих критичных объектов и явлений (это – условия α по ГОСТ Р 59341):

- для чрезвычайных происшествий, угрожающих безопасности людей и среды их обитания, должна быть не ниже 0,98;
- для оперативной информации об обстановке (в т. ч. по условиям функционирования шахты) – не ниже 0,95;
- для статистической информации при управлении СДК – не ниже 0,9;
- для команд и приказов с условиями их выполнения – не ниже 0,95.

Положим, согласно выданным главному конструктору СДК постановкам функциональных задач и принятым неблагоприятным сценарием возникновения и развития возможных аварийных ситуаций установлена ожидаемая частота появления новых объектов учета:

- для информации о чрезвычайных происшествиях – до трех раз в сутки (расчетные варианты $i = 1, 2, 3$);
- для оперативной информации об обстановке – в среднем до одного раза в час ($i = 4, 5, 6$);
- для статистической информации при управлении СДК – 2 раза в неделю ($i = 7, 8, 9$);
- для единожды вводимой информации (команд и приказов с условиями их выполнения) – 6 раз в сутки ($i = 10$).

Для проведения требуемых оценок технические решения главного конструктора в части сбора информации описаны следующими исходными данными, позволяющими охарактеризовать существенные различия в среднем времени подготовки, передачи и ввода новых объектов учета в БД СДК (именно для анализа этих различий анализируемые варианты снабжены индексом $i = 1,\ldots,10$).

Для информации о чрезвычайных происшествиях предусмотрена ее подготовка человеком. При этом для детальной информации и ее визуального контроля подготовка занимает в среднем 20 мин ($i = 1$), для детальной информации с программным контролем – 10 мин ($i = 2$), для укрупненной информации – до 5 мин ($i = 3$).

Для оперативной информации об обстановке возможна подготовка ее человеком с визуальным контролем в среднем около 20 мин ($i = 4$) либо с программным контролем до 10 мин ($i = 5$), а для некоторых видов информации, формируемой с помощью автоматических датчиков, в среднем за 30 с ($i = 6$). Т. е. результаты для $i = 4$ характеризуют существующую систему ручного контроля на местах, а результаты для $i = 5, 6$ характеризуют СДК.

Для статистической информации возможна подготовка информации человеком в течение 20 мин ($i = 7$), для детальной информации с программным контролем – до 10 мин ($i = 8$), для укрупненной информации с программным контролем – до 5 мин ($i = 9$).

Для команд и приказов информация готовится человеком в среднем в течение 5 мин ($i = 10$).

Согласно предложенным техническим решениям предусмотрены:

– передача информации от источников по телефону за среднее время до 10 мин ($i = 1, 4, 7$) или

автоматизированно с использованием СДК – до 1 мин (i = 2, 3, 5, 6, 8, 9, 10);
- ввод поступившей информации в БД за среднее время человеком от 1 мин (i = 4) до 10 мин (i = 1, 2, 5, 7, 8) или автоматически в СДК за 20 с (i = 3, 6, 9, 10).

С использованием модели 2.5.1. осуществлена количественная оценка полноты оперативного отражения в СДК состояния всех реально существующих критичных объектов и явлений. Результаты расчетов для сравнения вариантов приведены на рисунке 10.

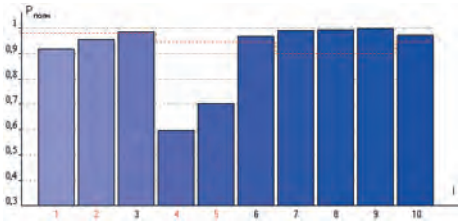


Рис. 10. Сравнительные оценки вариантов по вероятности того, что в СДК полностью отражены состояния всех реально существующих критичных объектов и явлений

Анализ результатов расчетов показывает (см. рис. 10):

- для информации о новых чрезвычайных происшествиях требованиям заказчика удовлетворяет лишь вариант оперативного обнаружения и подготовки укрупненной информации у источника с программным контролем, передачей через СДК с автоматическим вводом в БД (i = 3);
- для оперативной информации об обстановке требованиям заказчика отвечает лишь вариант с автоматическими датчиками СДК (i = 6). При этом другие варианты обнаружения и подготовки информации в течение 10–20 мин и длительного ввода ее в БД при передаче сколь угодно быстро не позволят обеспечить требуемую полноту оперативного отражения информации;
- для статистической информации при управлении СДК (i = 7, 8, 9) любой способ обнаружения и подготовки информации человеком, передачи любым из выбранных способов обеспечит полноту оперативного отражения в БД информации о реальных объектах учета и явлениях. Это объясняется относительной редкостью появления новой статистической информации;
- требуемая полнота оперативного отражения в системе реальных команд и приказов, поступающих через средства связи СДК (i = 10), будет обеспечена, что гарантируется быстротой передачи.

По результатам системного анализа сделан вывод: из множества сравниваемых технических решений лишь варианты 3, 6–10 отвечают задаваемым требованиям. Их реализация позволит обеспечить выполнение изначальных требований заказчика. Вместе с тем, заказчик, осознавая привычность работы в условиях неполноты информации на шахте, а также дороговизну технических изменений в проекте, вполне может согласиться на снижение изначальных требований к полноте оперируемой информации до такого уровня, что предъявляемые условия α по результатам моделирования выполняются. Эти результаты будут учтены при расчете интегрального риска в 3-й части статьи, где используется значение вероятностного показателя полноты оперативного отражения в системе новых объектов и явлений $Z_{\text{полн}}$, учитывающего рассчитываемое значение $P_{\text{полн}}$ и соответствующие условия α из ГОСТ Р 59341, приложения В.3.4 по допустимой вероятности того, что в СДК полностью отражены состояния всех реально существующих критичных объектов и явлений. Поскольку все требования к полноте оперируемой информации выполнены, в примере 3-й части статьи соответствующий показатель $Z_{\text{полн}}$ полагается равным 1.

2.6. Модели для оценки актуальности используемой информации

2.6.1. Общее

Под актуальностью информации понимается свойство безошибочной информации отражать текущее состояние прикладной области системы со степенью приближения, достаточной для получения на ее основе достоверной выходной информации в интересах конечного пользователя. Рассогласование реальной и хранимой в БД информации вызвано устареванием информации в результате какого-либо значимого изменения до следующего обновления этого изменения в БД – см. рис. 11. Т.е. актуальность характеризует старение информации во времени.

Для оценки актуальности обновляемой информации применяется «Модель для оценки актуальности обновляемой информации» из ГОСТ Р 59341, приложение В.3.5.

В качестве исходных данных используются:

ξ — среднее время между значимыми изменениями реальной информации относительно информации, хранимой в системе (т. е. ξ^{-1} — частота значимого изменения);

$T_{\text{база данных}}$ – среднее время подготовки, передачи и ввода в БД данных от источников;

q – среднее время между соседними обновлениями данных (т. е. q^{-1} – частота обновления данных) в системе при обновлении ее по регламенту;

В результате моделирования рассчитываются частные показатели: вероятность сохранения актуальности информации на момент ее использования $P_{\text{акт}}$

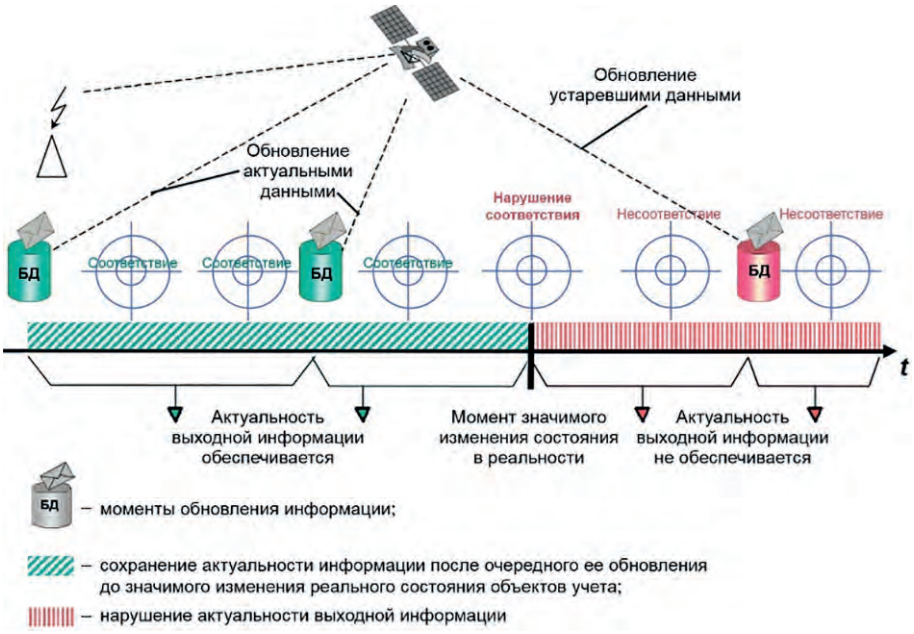


Рис. 11. Иллюстрация формирования актуальности выходной информации

и вероятностный показатель актуальности информации в системе $Z_{акт}$, учитывающий и соответствующие условия α из ГОСТ Р 59341, приложения В.3.5.

2.6.2. Пример для оценки актуальности

Важной практической задачей при создании и организации эффективного функционирования СДК является определение научно обоснованного периода обновления данных о состоянии параметров контролируемого оборудования и среды эксплуатации угольной шахты (например, давления, температуры, напряжения, загазованности и др.). С одной стороны, обновление данных по мере значимого изменения состояния оборудования необходимо для обеспечения достоверности информации с последующим ее применением по назначению. С другой стороны, слишком частое обновление этих данных необоснованно перегружает каналы связи и компьютерную память, приводит к программным сбоям, создает недопустимые временные задержки, может рассинхронизировать информационные процессы в СДК, нарушая тем самым режим реального времени функционирования самой СДК и лишая необходимой информационно-аналитической поддержки должностных лиц в процессе управления информацией на шахте. Учитывая результаты примера 2.5.2. о достижимости полноты отражения оперативной информации об обстановке с вероятностью не ниже $P_{полн} = 0{,}95$, задача формализована главным конструктором следующим образом: определить такой рациональный период обновления информации в СДК, при котором актуальность используемой информации будет не ниже, чем 0,95.

Анализ совокупности обновляемой информации при круглосуточной загрузке оборудования позволил выявить два варианта условий:

- обычные условия загрузки оборудования, характеризуемые частотой значимого изменения состояния оборудования 36 раз в сутки;
- условия наивысшей загрузки, возникающие для некоторого оборудования случайным образом (например, для вентиляторных установок или модульных дегазационных установок при повышенных скоплениях на местах газа метана), продолжающиеся несколько часов в сутки и характеризуемые частотой значимого изменения состояния оборудования 3 раза в час.

Среднее время съема, передачи и ввода в БД СДК телеметрических данных от оборудования составляет в среднем 16 с. Еще несколько секунд уходит на аналитическую обработку и доведение результатов обработки до пользователей. Это означает, что обновление чаще 25–30 с нецелесообразно из-за перегрузки и вычислительной неспособности своевременно обработать такую часто обновляемую информацию от сотен источников.

Моделирование для определения искомого периода обновления информации в СДК осуществлено по этим исходным данным с использованием модели 2.6.1. Сравнительные результаты расчетов приведены на рисунках 12–15.

Анализ результатов расчетов показывает, что для обеспечения актуальности информации в СДК с вероятностью не ниже 0,95, период обновления может быть выбран следующим образом: для обычных

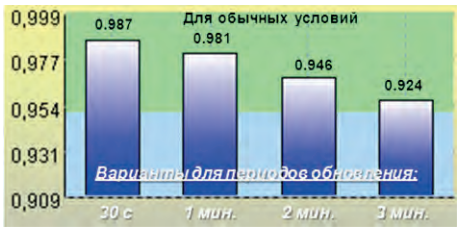


Рис. 12. Вероятность сохранения актуальности информации для обычных условий загрузки оборудования

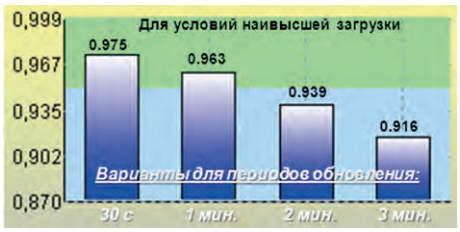


Рис. 13. Вероятность сохранения актуальности информации для условий наивысшей загрузки оборудования

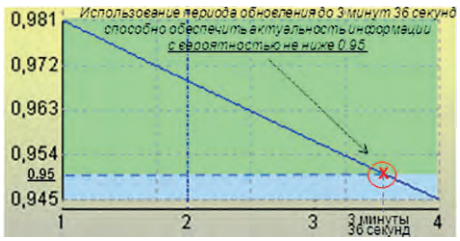


Рис. 14. Зависимость вероятности сохранения актуальности информации для обычных условий загрузки оборудования от периода обновления (в минутах)

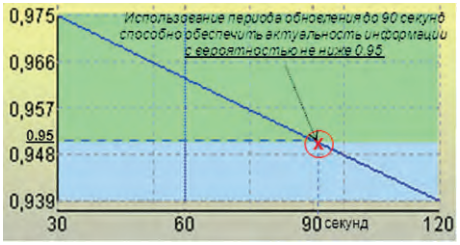


Рис. 15. Зависимость вероятности сохранения актуальности информации для условий наивысшей загрузки оборудования от периода обновления (в секундах)

условий загрузки оборудования – до 3 мин 36 с; для условий наивысшей загрузки – до 90 с. В результате системного анализа определено: из соображений недопущения вычислительной перегрузки СДК наиболее рациональным в условиях неопределенности функционирования шахты признан период обновления информации в СДК, равный 90 с.

Для определенности при расчете интегрального риска нарушения реализации процесса управления информацией системы в части 3 статьи использована достигаемая вероятность сохранения актуальности информации $P_{акт} = 0{,}95$. Эти результаты будут учтены при определении значения вероятностного показателя $Z_{акт}$, учитывающего условия и соответствующие им α из ГОСТ Р 59341, приложения В.3.5. Поскольку все требования к актуальности оперируемой информации выполнены, в примере 3-й части статьи этот показатель $Z_{акт}$ полагается равным 1.

2.7. Модели для оценки безошибочности информации после контроля

2.7.1. Общее

Под безошибочностью информации понимается свойство информации не иметь явных или скрытых ошибок и/или искажений. Понятие ошибки должно быть определено в эксплуатационной документации для каждой конкретной задачи ИС в зависимости от целевого назначения информации.

Модель процессов анализа каких-либо объектов (например, информации, образцов материала, событий, результатов работы и др.) поясним на примере.

Модель может использоваться для оценки безошибочности информации в результате контроля и для оценки корректности информации в результате ее обработки.

Определение: информация считается безошибочной в результате контроля, если в процессе контроля до истечения заданного срока контроля все наличествующие ошибки выявлены (и, соответственно, исправлены) и новые ошибки не внесены.

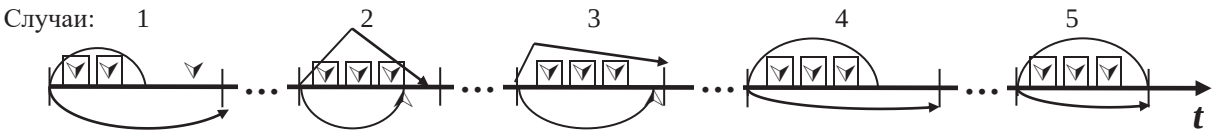
Определение: информация считается корректно обработанной, если в процессе ее анализа до истечения заданного срока обработки все принципиальные моменты учтены и алгоритмические ошибки не допущены.

Поскольку содержание модели для оценки корректности информации в результате ее обработки отличается лишь формулировкой исходных данных, приведенных в подразделе 2.8., то ниже ограничимся изложением содержания модели в приложении к контролю информации. Суть формализации отражена на рис. 16:

Случаи 1, 2, 3 характеризуют наличие ошибок после контроля, для случаев 4, 5 безошибочность после контроля обеспечена.

Случай 1 – наработка на ошибку или допустимое время контроля истекли раньше, чем закончился проверяемый документ, и после этого осталась хотя бы одна наличествующая ошибка. Ошибки контроля 1-го рода при этом не допускались.

Случай 2 – допустимое время контроля истекло раньше, чем закончился проверяемый документ, однако в непроверенной части ошибок не осталось. Вместе с тем, во время работы были допущены ошибки контроля 1-го рода.



- наличие изначальной ошибки в проверяемом документе;
- выявленная и исправленная ошибка;
- ошибка, случайно внесенная контролером (ошибка контроля 1-го рода);
- минимальное из двух времен:
а) периода непрерывной работы контролера (например, для контролера-человека по истечении этого периода следует восстановительный отдых, для программных средств контроля - время их непрерывного использования);
б) времени реального контроля документа, определяемого его объемом и скоростью контроля;
- задаваемое допустимое время контроля (истекает в момент пересечения оси *t* со стрелкой);
- минимальное из двух времен:
а) времени наработки контролера на ошибку, по истечении которого наличествующая ошибка пропускается (ошибка контроля 2-го рода);
б) задаваемого допустимого времени контроля;
- время до свершения первой ошибки контроля 1-го рода.

Рис. 16. Иллюстрация формальных процессов контроля безошибочности информации (фрагмент)

Случай 3 – допустимое время контроля не истекло раньше, чем закончился проверяемый документ, вследствие чего все изначальные ошибки выявлены и исправлены. Вместе с тем, во время работы были допущены ошибки контроля 1-го рода.

Случаи 4 и 5 — все ошибки выявлены и исправлены, и новые не внесены. При этом случай 4 аналогичен случаю 2, а случай 5 — случаю 3 с тем отличием, что ошибки 1-го рода не были допущены.

Для оценки безошибочности информации после контроля применяется «Модель для оценки безошибочности информации после контроля» из ГОСТ Р 59341, приложение В.3.6.

В качестве исходных данных используются:

V – объем контролируемой информации;

μ – доля первоначальных ошибок в контролируемой информации в объеме V (до контроля), т. е. произведение $V \cdot \mu$ принимает безразмерное значение от 0 до 1;

ν – средняя скорость контроля информации;

n — частота ошибок контроля 1-го рода (когда реальное отсутствие ошибки истолковывается как наличие ошибки);

$T_{нар}$ – среднее время наработки контролера на ошибку 2-го рода, после истечения которого первая же реальная ошибка в контролируемом объеме информации оказывается пропущенной (для программно-технических средств — это время наработки на отказ);

$T_{непр}$ – период непрерывной работы контролера;

$T_{зад}$ – задаваемое время на контроль информации.

В результате моделирования рассчитываются частные показатели: вероятность отсутствия ошибок в информации после ее контроля $P_{безош}$ и вероятностный показатель безошибочности информации в системе $Z_{безош}$, учитывающий и соответствующие условия α из ГОСТ Р 59341, приложения В.3.6.

2.7.2. Пример для оценки безошибочности

Объектами анализа являются информационные ресурсы, вводимые в СДК, и технологии контроля их безошибочности. При проектировании СДК необходимо обосновать технологию контроля информации, обеспечивающую безошибочность входной информации. Требования заказчика сформулированы следующим образом: используемые технологии контроля входной формализованной и неформализованной информации должны обеспечивать ее безошибочность, в частности, вероятность отсутствия ошибки во входном сообщении, вводимом в БД СДК, должна быть не ниже 0,95, при этом допустимое время контроля не должно превышать 10 мин для графических документов и входных обобщенных документов до 10000 знаков и 1 ч для детальных документов объемом до 50000 знаков.

Для проведения требуемых оценок технические решения главного конструктора в части контроля информации описаны следующими исходными данными, позволяющими охарактеризовать существенные различия в рассматриваемых вариантах технологий контроля (именно для анализа этих различий анализируемые варианты снабжены индексом $i = 1,\ldots,10$). Согласно постановкам функциональных

задач информация, подлежащая контролю, обладает следующими характеристиками: средний объем коротких документов составляет в среднем 20 контролируемых объектов для графической информации (расчетные варианты $i = 1, 2$), 10000 текстовых знаков для обобщенных документов ($i = 3, 4, 7–10$) и 50000 знаков для детальных документов ($i = 5, 6$).

Для оценки безошибочности информации в СДК технические решения главного конструктора предусматривают осуществление лишь визуального контроля всей информации. С применением настоящей методики осуществляется количественная оценка ожидаемой безошибочности используемой информации и выявление необходимости создания вспомогательных средств программного контроля и обоснования системных требований к ним.

По результатам сравнения с аналогами установлено, что частота ошибок в документах может составлять одну ошибку на 100 графических объектов ($i = 1, 2$), одну ошибку на 100 знаков ($i = 3, 5, 7, 8$) или 200 знаков ($i = 4, 6$) неформализованной информации. В результате натурных экспериментов и сравнения с аналогами установлено, что технология контроля информации характеризуется следующими исходными данными:

- скорость контроля равна 20 объектам в минуту для графической информации ($i = 1, 2$), 2000 табличным знакам в минуту ($i = 3–6$) без программной поддержки и 6000 знакам в минуту (i = 7–10) с использованием средств программного контроля;
- частота ошибок контроля 1-го рода составляет одну ошибку на 100 мин работы для высококвалифицированного контролера ($i = 1, 3, 5$) и одну ошибку на 50 мин для контролера средней квалификации ($i = 2, 4, 6$). Кроме того, при поддержке программными средствами контроля частота ошибок 1-го рода может быть снижена на порядок, т. е. для высококвалифицированного контролера она составит одну ошибку на 16 ч ($i = 7, 9$), а для контролера средней квалификации – одну ошибку на 8 ч ($i = 8, 10$) работы;
- среднее время наработки на ошибку 2-го рода соответственно составляет 1 ч для высококвалифицированного контролера ($i = 1, 3, 5, 7, 9$) и 40 мин для среднеквалифицированного ($i = 2, 4, 6, 8, 10$) контролера;
- среднее непрерывное время работы человека-контролера составляет 45 мин ($i = 1–10$), после чего следует необходимое восстановление концентрации внимания (вплоть до смены контролера);
- на однократный контроль короткого и обобщенного документа отводится в среднем 10 мин ($i = 1–4, 7–10$), а на однократный контроль детального документа – 1 ч ($i = 5, 6$).

При моделировании предусмотрено использование повторного визуального контроля ($i = 9, 10$), причем в качестве исходной доли ошибок после первого контроля выступают результаты расчетов по настоящей методике соответственно для вариантов $i = 7$ и $i = 8$.

С использованием модели В.3.6 по этим исходным данным проведена количественная оценка безошибочности информации после контроля. Сравнительные результаты расчетов приведены на рисунках 17 и 18.

Рис. 17. Вероятность отсутствия ошибок в информации без контроля для 10 вариантов сравнения

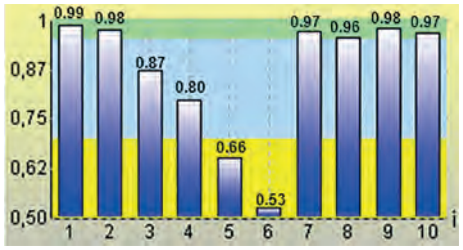


Рис. 18. Вероятность отсутствия ошибок в информации после контроля для 10 вариантов сравнения

Анализ результатов расчетов показывает:

- для коротких графических документов вероятность отсутствия ошибок после контроля специалистом средней и высокой квалификации превышает 0,98, причем она по-прежнему будет удовлетворять требованиям при возможном увеличении среднего объема контролируемой информации до 40 объектов ($i = 1, 2$);
- для документов объемом 10000–50000 знаков ($i = 3–6$) ошибки без контроля неизбежны. При контроле без поддержки программными средствами вероятность отсутствия ошибок ниже требуемой (от 0,53 до 0,87) независимо от квалификации проверяющих;
- применение поддерживающих программных средств контроля ($i = 7, 8$) позволяет повысить вероятность отсутствия ошибок в документах объемом 10000 знаков до уровня 0,96–0,97;
- применение повторного визуального контроля с использованием программных средств ($i = 9, 10$)

оказывается избыточным как для высококвалифицированных, так и среднеквалифицированных контролеров по сравнению с вариантами $i = 7, 8$.

Вывод: для выполнения заданных требований выявлена объективная необходимость разработки специальных программных средств поддержки контроля информации в СДК. Рекомендации: основными требованиями к разработке этих программных средств, а в последующем – и для эксплуатационной документации должны быть:

- требования к скорости контроля – не ниже 20 графических объектов в минуту и 6000 текстовых знаков в минуту;
- требования к допустимой частоте ошибок первого рода – не чаще одной ошибки за 500 мин работы;
- требования к допустимой наработке до первого пропуска ошибки – в среднем не менее 40 мин;
- регламентация времени работы человека-контролера, в частности непрерывное время контроля не должно превышать 45 мин.

Поскольку все требования к безошибочности информации в системе после контроля выполнены, в примере 3-й части статьи показатель $Z_{\text{безош}}$ полагается равным 1.

2.8. Модели для оценки корректности информации после обработки

2.8.1. Общее

Под корректностью обработки информации в системе понимается свойство системы обеспечивать получение правильных согласованных результатов или эффектов обработки информации. Информация считается корректно обработанной, если в процессе ее анализа до истечения заданного срока обработки все принципиальные моменты учтены и алгоритмические ошибки не допущены. Требуемая корректность обработки информации программно-аналитическими средствами в системе и выходной информации от системы пользователями обеспечивается на основе применения эффективных способов анализа информации (как с использованием, так и без использования прикладного программного обеспечения), позволяющих учесть важную для принятия решения информацию и не допустить алгоритмических ошибок при анализе всего объема информации. Корректность в обработке информации является следствием приемлемого соотношения между объемом анализируемой информации, частью важной для принятия решения информации, подлежащей учету, скоростью анализа информации, частотой ошибок аналитика, длительностью его непрерывной работы и ограничениями на допустимое время обработки.

Формализация процессов обработки информации в системе полностью аналогична формализации для модели 2.7.1. с точностью до переопределений исходных данных. Для оценки корректности обработки информации применяется «Модель для оценки корректности обработки информации» из ГОСТ Р 59341, приложение В.3.7.

В качестве исходных данных используются:

V – объем информации, подлежащий обработке (анализу);

μ – часть важной для принятия решения информации, которая должна быть объективно использована при обработке (анализе) информации объема V, т. е. произведение $V \cdot \mu$ принимает безразмерное значение от 0 до 1;

v – скорость обработки (анализа);

n – частота ошибок обработки (анализа) 1-го рода (когда несущественная для принятия решения информация ошибочно воспринимается в качестве важной);

$T_{\text{нар}}$ – среднее время наработки на алгоритмическую ошибку (когда объективно важная для принятия решения информация игнорируется, это аналог ошибки контроля 2-го рода);

$T_{\text{непр}}$ – период непрерывной работы аналитика (в качестве аналитика могут выступать программно-аналитические средства или пользователь системы);

$T_{\text{зад}}$ – задаваемое время на обработку (анализ) информации.

В результате моделирования рассчитываются частные показатели: вероятность получения корректных результатов обработки информации $P_{\text{корр}}$ и вероятностный показатель корректности обработки информации в системе $Z_{\text{корр}}$, учитывающий и соответствующие условия α из ГОСТ Р 59341, приложения В.3.7.

2.8.2. Пример для оценки корректности

Объектами анализа являются информационные активы СДК и технологии их обработки по назначению. При проектировании СДК главный конструктор оценивает целесообразность разработки вспомогательных экспертных систем для обработки информации. Заказчик использует настоящую методику для количественной оценки ожидаемой корректности обработки информации в режиме реального времени функционирования СДК, а главный конструктор – для дальнейшего выявления рациональных технических способов удовлетворения требований технического задания. Согласно постановкам функциональных задач аналитики (операторы) анализируют те же объемы информации, что и в примере 2.7.2., но уже с целями подготовки и принятия прагматических решений по обеспечению промышленной безопасности на предприятии. Для проведения требуемых оценок с учетом существенных различий в рассматриваемых вариантах технологий обработки информации

анализируемые варианты по-прежнему снабжены индексом $i = 1,...,10$. Т. е. обобщенная информация характеризуется объемом до 20 объектов ($i = 1, 2$), а детальная информация – объемом до 10000 знаков ($i = 3, 4, 7–10$) и до 50000 знаков ($i = 5, 6$). При анализе информации осуществляется не контроль, а семантическая обработка аналитиком. Примером малого объема анализируемой информации может служить обобщенное состояние контролируемых объектов на электронной карте с использованием мнемосхем.

Примером большего объема анализируемой информации может служить детальная информация о состоянии контролируемого оборудования шахты. Таковых объектов учета для СДК, охватывающих несколько шахт, могут быть тысячи и десятки тысяч.

Пусть в обобщенной информации малого объема (i = расчетные варианты 1, 2) вся информация является принципиальной, в детальной (для $i = 3–10$) процент принципиальной информации не превышает 50 %. В обязанности аналитика (оператора) входят корректные выделение и осмысление этой информации в режиме реального времени для последующего использования ее по назначению. Требуемый уровень корректности обработки информации по выбранному вероятностному показателю – не ниже 0,95.

В результате натурных экспериментов и сравнения с аналогами установлено, что технология обработки информации характеризуется следующими исходными данными. Скорость обработки информации составляет 20 объектов в минуту для аналитика как высокого ($i = 1, 3, 5, 7, 9$), так и среднего уровня квалификации ($i = 2, 4, 6, 8, 10$) и 2000 знаков в минуту для оператора-аналитика ($i = 3–5$). Использование специальной экспертной системы автоматической обработки данных (целесообразность создания которой оценивается Главным конструктором, $i = 6–10$) позволяет повысить скорость обработки детальной информации аналитиком до 6000 знаков в минуту. Частота ошибок анализа 1-го рода, среднее время наработки на алгоритмическую ошибку и непрерывное время работы человека (аналитика, оператора) сохраняются теми же, что и в примере 5 для контроля информации. Допустимое время оперативной обработки информации объемом 10000 знаков составляет 10 мин для $i = 1–4, 7, 8$, при детальной аналитической обработке документов объемом 50000 знаков – до одного часа ($i = 5, 6$).

Моделирование осуществлено по этим исходным данным с использованием рекомендаций 2.8.1. – см. результаты расчетов на рис. 19, 20.

Анализ обобщенных результатов расчетов показывает:

- вероятность получения корректных результатов обработки обобщенной информации составляет 0,96–0,97 для аналитика как среднего, так и высокого уровня квалификации ($i = 1, 2$) из-за сравнительно небольшого объема анализируемой информации. Часть неучтенной информации не превысит 5 %;
- для документов объемом 10000 знаков за счет применения специальной экспертной системы ($i = 7, 8$) корректность обработки информации оператором как среднего, так и высокого уровня квалификации составит 0,96–0,97 ($i = 7, 8$) против 0,80–0,88 (для $i = 3, 4$), характерных для варианта обработки информации без ее использования. При этом часть неучтенной информации составит для $i = 7, 8$ лишь 1,5–2.2 % против 6,2–10,1 % для $i = 3, 4$;
- для документов объемом 50000 знаков применение оператором экспертной системы ($i = 6$) позволит повысить вероятность корректной обработки до уровня 0,81 против 0,66 без ее использования ($i = 5$), но для корректности обработки информации этого явно недостаточно;
- использование в автоматическом режиме специальной экспертной системы обеспечит корректность обработки лишь на уровне 0.58–0,59 ($i = 9, 10$), что объясняется слабой производительностью применяемых программно-технических средств, не позволяющих за одну минуту автоматически обработать весь объем принципиальной информации. Часть неучтенной информации превысит 20 %.

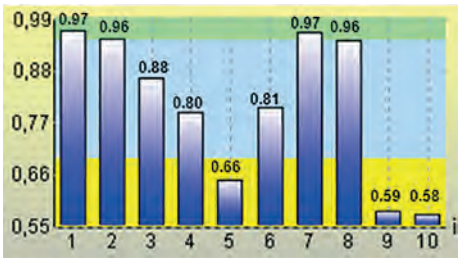


Рис. 19. Вероятность получения корректных результатов обработки информации для 10 вариантов сравнения

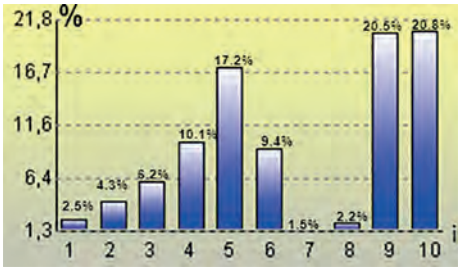


Рис. 20. Часть принципиальной информации, не учтенная в процессе обработки для 10 вариантов сравнения

Учитывая потенциальные возможности специальной экспертной системы поддержки принятия решений и ее осуществимость, при расчете интегрального риска в примере 3-й части статьи использован вероятностный коэффициент корректности обработки информации в системе $Z_{\text{корр.}} = 1$.

2.9. Модели для оценки безошибочности действий пользователей и персонала

Модель позволяет оценить воздействие «человеческого фактора» на уровне вероятности безошибочных действий пользователей и персонала системы в течение заданного периода прогноза $P_{\text{чел}}(_{\text{Тзад}})$.

Требуемая безошибочность действий пользователей и персонала системы в течение заданного времени обеспечивается на основе профессионального отбора, специальной подготовки пользователей и обслуживающего персонала системы, реализации и использования эффективных средств программной поддержки. Безошибочность является следствием приемлемого соотношения между частотой возможных ошибок, временем их обнаружения и исправления.

Для оценки безошибочности действий пользователей и персонала системы в течение заданного периода прогноза применяются одноименные модели из ГОСТ Р 59341, приложения В.3.8.

В качестве исходных данных используются:

σ – частота возникновения источников угроз из-за «человеческого фактора»;

β – среднее время развития угроз (активизации источников угроз) с момента их возникновения до нарушения целостности моделируемой системы;

$T_{\text{меж}}$ – среднее время между окончанием предыдущей и началом очередной диагностики целостности моделируемой системы;

$T_{\text{диаг}}$ – среднее время системной диагностики целостности моделируемой системы;

$T_{\text{восст}}$ – среднее время восстановления нарушаемой целостности моделируемой системы;

$T_{\text{зад}}$ – задаваемая длительность периода прогноза.

В результате моделирования рассчитываются частные показатели: вероятность безошибочных действий пользователей и персонала системы в течение заданного периода прогноза $P_{\text{чел}}(_{\text{Тзад}})$ и вероятностный показатель безошибочности действий пользователей и персонала системы в течение заданного периода прогноза $Z_{\text{чел}}(_{\text{Тзад}})$, учитывающего и соответствующие условия α из ГОСТ Р 59341, приложения В.3.8.

Учитывая математическую идентичность моделей для оценки надежности предоставления информации и безошибочности действий пользователей и персонала системы, некоторые сравнительные возможности моделирования приведены в 3-й части статьи с использованием понятий сложной «моделируемой системы».

2.10. Модель для оценки защищенности системы от опасных программно-технических воздействий

Для оценки безопасности информации в части сохранения целостности моделируемой системы в условиях опасных программно-технических воздействий применяется «Модель для оценки сохранения целостности моделируемой системы в условиях опасных программно-технических воздействий» из ГОСТ Р 59341, приложение В.3.9.2.

В качестве исходных данных используются:

σ – частота возникновения источников угроз в виде источников опасных программно-технических воздействий, ведущих к нарушению безопасности информации;

β – среднее время развития угроз (активизации источников угроз) с момента их возникновения до нарушения целостности моделируемой системы (т.е. выполняемого процесса или защищаемых активов, используемых при выполнении процесса) в результате опасных программно-технических воздействий;

$T_{\text{меж}}$ – среднее время между окончанием предыдущей и началом очередной диагностики целостности моделируемой системы;

$T_{\text{диаг}}$ – среднее время системной диагностики целостности моделируемой системы;

$T_{\text{восст}}$ – среднее время восстановления нарушаемой целостности моделируемой системы;

$T_{\text{зад}}$ – задаваемая длительность периода прогноза.

В результате моделирования рассчитывается частный показатель: вероятность отсутствия опасного программно-технического воздействия на систему $P_{\text{возд}}(T_{\text{зад}})$ в течение заданного периода прогноза $T_{\text{зад}}$.

Учитывая математическую идентичность моделей для оценки надежности предоставления информации, безошибочности действий пользователей и персонала системы с моделью для оценки защищенности системы от опасных программно-технических воздействий, некоторые сравнительные возможности моделирования приведены в 3-й части статьи с использованием понятий сложной «моделируемой системы».

2.11. Модели для оценки защищенности активов от несанкционированного доступа (НСД)

2.11.1. Общее

Настоящая вероятностная модель справедлива для оценки защищенности ресурсов без учета периода их объективной ценности, т.е. лишь исходя из реализуемой технологии защиты. Другими словами, защищаемые ресурсы полагаются априори ценными в течение бесконечного периода времени.

Построение вероятностного пространства для оценки отсутствия воздействий в результате НСД осуществляется в предположении реализации в системе

элементов защиты ресурсов от потенциального нарушителя. В приложении к ИС защищаемыми являются в первую очередь информационные и программные ресурсы. Однако, модель является более общей, в качестве защищаемых могут выступать людские, материальные, финансовые и др. ресурсы согласно вербальной модели угроз (см. также сайт ФСТЭК России https://bdu.fstec.ru/).

Для доступа к хранимым в системе ресурсам выстраивается последовательность преград от злоумышленника с тем, чтобы допущенный пользователь, зная и реализуя алгоритм преодоления этих преград, мог решать свои задачи в установленном штатном режиме. В качестве нарушителя рассматривается лицо, не посвященное в тайну преодоления защитных преград. Вскрывая каким-либо доступным образом алгоритм преодоления преград, злоумышленник вполне может получить доступ к ресурсам системы.

Рассматривается наиболее тяжелый режим функционирования системы защиты в ожидании постоянной угрозы ее вскрытия. Нарушитель в состоянии проникнуть в систему лишь при условиях, что

- во-первых, ему станет известна система защиты в части, необходимой для достижения его целей;
- во-вторых, он успеет получить доступ к информационным и/или программным ресурсам системы до того, как эта система защиты видоизменится (после чего перед нарушителем возникнет проблема повторного преодоления защитных преград).

При моделировании действия «умного» нарушителя, оснащенного возможными высокотехнологичными средствами вскрытия системы зашиты, могут быть охарактеризованы лишь большей скоростью преодоления защитных преград.

Для оценки безопасности информации в части защищенности активов от НСД применяется одноименная модель из ГОСТ Р 59341, приложения В.3.9.3.

Таблица 1.

Характеристики сценария угроз НСД и системы защиты

Преграда	Частота смены значения параметра преграды	Среднее время преодоления преграды нарушителем	Возможный способ преодоления преграды
1. Охраняемая территория со сменой охраны	через 2 ч	30 мин	Скрытое проникновение на территорию
2. Пропускная система на объект СДК (в т. ч. доступ к рабочим местам пользователей со сменой службы контроля)	через 1 сут	10 мин	Подделка документов, сговор, обман
3. Электронный ключ для включения компьютера	через 5 лет (наработка до замены)	1 нед	Кража, принудительное изымание ключа, сговор
4. Пароль для входа в систему	через 1 мес	1 мес	Подсматривание, принудительное выпытывание, сговор, подбор пароля
5. Пароль для доступа к программным устройствам	через 1 мес	10 сут	Подсматривание, принудительное выпытывание, сговор, подбор пароля
6. Пароль для доступа к требуемой информации	через 1 мес	10 сут	Подсматривание, принудительное выпытывание, сговор, подбор пароля
7. Зарегистрированный внешний носитель информации для записи	через 1 год	1 сут	Кража, принудительная регистрация, сговор
8. Подтверждение подлинности пользователя в процессе сеанса	через 1 мес	1 сут	Подсматривание, принудительное выпытывание, сговор
9. Телемониторинг	через 5 лет (наработка до замены)	2 сут	Имитация неисправности, ложные ролики, маскировка под персонал, сговор
10. Шифрование информации со сменой ключей	через 1 мес	2 года	Расшифровка, сговор

В качестве исходных данных используются:

M – количество преград, которое необходимо преодолеть нарушителю, чтобы получить доступ к ресурсам системы;

f_m – среднее время между соседними изменениями параметров защиты m-й преграды;

u_m – среднее время преодоления (вскрытия значений параметров защиты) m-й преграды.

В результате моделирования рассчитывается частный показатель: вероятность обеспечения защищенности активов системы от НСД – $P_{\text{НСД}}$..

2.11.2. Пример для оценки защищенности от НСД

Пример демонстрирует подход к оценке вероятности обеспечения защищенности активов СДК от несанкционированного доступа $P_{\text{НСД}}$.

Объектами анализа являются информационные и программные ресурсы СДК для построения на шахте эффективной защиты от НСД.

Анализируются возможности и целесообразность создания 10 преград для защиты от НСД. На основании вербальной модели угроз в таблице отражены предполагаемые характеристики сценария угроз НСД и системы защиты информации.

Моделирование осуществлено по этим исходным данным с использованием рекомендаций 2.11.1. Результаты расчетов отражены на рисунке 21.

Анализ полученных результатов расчета показывает следующее.

Первые 3 преграды преодолеваются с вероятностью около 0,745. Использование сменяемых паролей один раз в месяц для 4, 5 и 6 преград позволяет в три раза поднять защищенность с 0,255 до 0,872. Однако общая защищенность системы после введения первых шести преград остается слабой (0,872).

Введение 7, 8, 9 преград практически бесполезно, т.к. не обеспечивает заметного повышения защищенности системы для заданных значений исходных данных (0,877 по сравнению с 0,872).

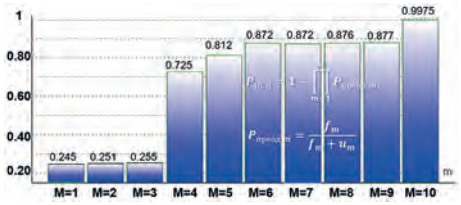


Рис. 21. Рост вероятности обеспечения защищенности активов СДК от НСД с увеличением количества и качества преград, m = 1,..., M

Использование криптографических средств защиты (10-я преграда) позволяет более существенно повысить защищенность информационных ресурсов от НСД – до уровня 0,9975. Это в 399 раз превышает вероятностный риск преодоления преград в системе защиты от НСД [0,9975/(1–0,9975)].

Для определенности при расчете интегрального риска с учетом требований по защите информации от НСД в примере 3-й части статьи использована достигаемая вероятность обеспечения защищенности активов СДК от НСД $P_{\text{НСД}} = 0{,}9975$.

2.12. Модели для оценки конфиденциальности используемой информации

2.12.1. Общее

Требуемая конфиденциальность информации обеспечивается на основе реализации мероприятий, гарантирующих защищенность информационных ресурсов системы от НСД до истечения периода объективной конфиденциальности (ПОК) данной информации. Моделируемые случаи соотношения между временем смены значений параметров преград системы защиты и их расшифровки (вскрытия) и периодом объективной конфиденциальности информации для одной преграды приведены на рисунке 22.

Случай 1 – НСД осуществлен до истечения ПОК. Случай 2 – НСД осуществлен после истечения ПОК. Случай 3 – НСД не состоялся. Случай 4 – НСД осуществлен, и период объективной ценности ресурсов

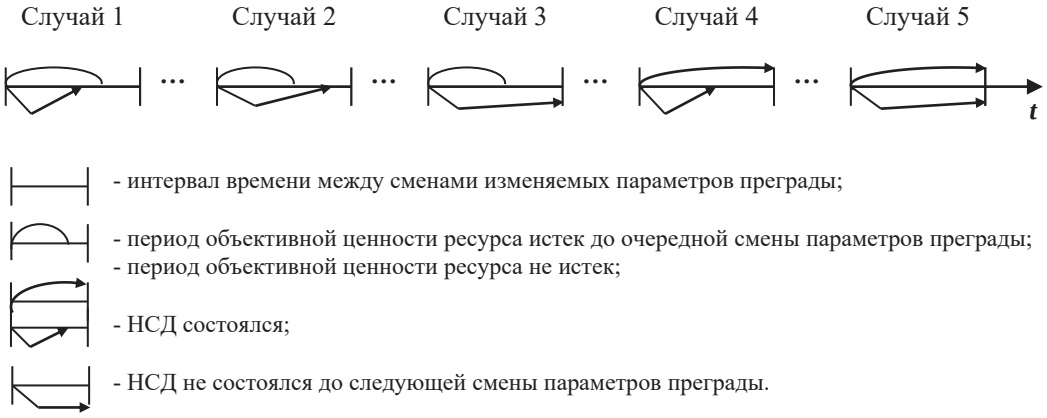


Рис. 22. Формализация процессов НСД с учетом ценности ресурсов

дольше, чем время между соседними сменами параметров системы защиты. Случай 5 – параметры системы защиты сменились раньше, чем истек ПОК и осуществлен НСД (для нарушителя требуется повторное преодоление преграды).

Для оценки безопасности информации в части сохранения конфиденциальности используемой информации применяются «Модель для оценки сохранения конфиденциальности используемой информации» из ГОСТ Р 59341, приложение В.3.9.3.

В качестве исходных данных используются:

M – количество преград, которое необходимо преодолеть нарушителю, чтобы получить доступ к ресурсам системы;

f_m – среднее время между соседними изменениями параметров защиты m-й преграды;

u_m – среднее время преодоления (вскрытия значений параметров защиты) m-й преграды;

$T_{\text{конф}}$ – период объективной конфиденциальности используемой информации.

В результате моделирования рассчитывается частный показатель: вероятность сохранения конфиденциальности используемой информации $P_{\text{конф}}(T_{\text{конф}})$ в течение периода объективной конфиденциальности $T_{\text{конф}}$ (период $T_{\text{конф}}$ может играть роль периода прогноза $T_{\text{зад}}$).

2.12.2. Пример для оценки конфиденциальности используемой информации

Пример демонстрирует подход к оценке вероятности сохранения конфиденциальности используемой информации в течение периода объективной конфиденциальности $P_{\text{конф}}(T_{\text{конф}})$.

Объектами анализа являются те же информационные и программные ресурсы СДК при тех же используемых преградах системы защиты от НСД. Дополнительно учтена длительность периода объективной конфиденциальности информации, характеризующего ценность ресурса. С учетом того, что большинство примеров в 3-й части статьи ориентированы на период прогноза 1 мес, в настоящем примере роль периода объективной конфиденциальности информации играет именно этот период прогноза. Характеристики десяти преград те же, что и в примере 2.11.2.

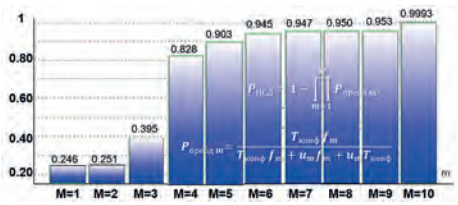


Рис. 23. Рост вероятности сохранения конфиденциальности используемой информации с увеличением количества и качества преград, m = 1,..., M

Моделирование осуществлено по исходным данным таблицы заданием $T_{\text{конф}}$ = 1 мес и использованием рекомендаций 2.12.1. Результаты расчетов отражены на рисунке 23.

Системный анализ полученных результатов расчета показывает следующее.

Использование первых 6 преград (охрана, пропускной режим, электронный ключ и различные системы паролей) обеспечит конфиденциальность информации с вероятностью не выше 0,945.

Использование всех 10 преград обеспечит требуемую конфиденциальность информации в системе: 0,9993, что более, чем в 1400 раз превышает вероятностный риск нарушения конфиденциальности информации [0,9993/(1— 0,9993)]. В условиях примера это может рассматриваться как более обоснованное значение показателя эффективности защиты информации.

Для определенности при расчете интегрального риска в 3-й части статьи использована достигаемая вероятность сохранения конфиденциальности используемой информации в течение месяца $P_{\text{конф}}(T_{\text{конф}}) = 0{,}9993$.

Выводы по 2-й части работы

1. Методические положения 1-й части статьи детализированы путем предложения следующих вероятностных моделей, позволяющих проведение исследований «моделируемых систем» в виде «черного ящика»: «Модели для оценки надежности предоставления информации и выполнения операций»; «Модели для оценки своевременности предоставления информации и выполнения операций»; «Модели для оценки полноты используемой информации»; «Модели для оценки актуальности используемой информации»; «Модели для оценки безошибочности информации после контроля»; «Модели для оценки корректности информации после обработки»; «Модели для оценки безошибочности действий пользователей и персонала»; «Модели для оценки защищенности системы от опасных программно-технических воздействий»; «Модели для оценки защищенности активов от несанкционированного доступа»; «Модели для оценки конфиденциальности используемой информации». Также разъяснен предложенный «Метод использования универсальной вспомогательной модели показателя для определения исходных данных в расчетах».
2. Использование некоторых возможностей предложенных моделей продемонстрировано на примерах:

➢ оценки своевременности предоставления информации и выполнения операций в приложении

к решению практических задач, связанных с производительностью опорных узлов и региональных блокчейн-хабов гипотетичной информационной среды взаимодействия государственных и частных организаций;

- оценки и обеспечения полноты и актуальности используемой информации, безошибочности информации после контроля, корректности обработки информации, защищенности ресурсов ИС от НСД и сохранения конфиденциальности используемой информации в приложении к системе дистанционного контроля гипотетичной угольной шахты.

Демонстрация оценок других свойств, характеризующих качество функционирования ИС с помощью предложенных моделей, охватывающих понятия сложной «моделируемой системы», будет проведена в 3-й части статьи.

(Окончание статьи следует в №1–2025 журнала «Вопросы кибербезопасности»)

Литература

1. *Костогрызов А. И., Нистратов А. А. Методические положения по вероятностному прогнозированию качества функционирования информационных систем. Часть 1. Общий подход // Правовая информатика, 2024, №3. С.13–31.*
2. *Костогрызов А. И., Петухов А. В., Щербина А. М. Основы оценки, обеспечения и повышения качества выходной информации в АСУ организационного типа. М.: Изд. «Вооружение. Политика. Конверсия», 1994. 278 с.*
3. *Костогрызов А. И., Липаев В. В. Сертификация качества функционирования автоматизированных информационных систем. – М. Изд. «Вооружение, политика, конверсия», 1996. 278 с.*
4. *Костогрызов А. И., Нистратов Г. А. Стандартизация, математическое моделирование, рациональное управление и сертификация в области системной и программной инженерии. – М. Изд. «Вооружение, политика, конверсия», 2004, 2-е изд. 2005. 395 с.*
5. *Костогрызов А. И., Степанов П. В. Инновационное управление качеством и рисками в жизненном цикле систем – М.: Изд. «Вооружение, политика, конверсия», 2008. – 404 с.*
6. *A. Kostogryzov, A. Nistratov, G. Nistratov SOME APPLICABLE METHODS TO ANALYZE AND OPTIMIZE SYSTEM PROCESSES IN QUALITY MANAGEMENT («Некоторые прикладные методы для анализа и оптимизации системных процессов в управлении качеством») // InTech, 2012, ISBN979-953-307-778-8, 2012, pp. 127–196. http://www.intechopen.com/books/total-quality-management-and-six-sigma/some-applicable-methods-to-analyze-and-optimize-system-processes-in-quality-management*
7. *Абросимов Н. В., Алешин А. В., Махутов Н. А. и др. /Под ред. Махутова Н. А./ Безопасность России. Правовые, социально-экономические и научно-технические аспекты. Научные основы техногенной безопасности. М.: МГОФ «Знание», 2015, 936 с.*
8. *Абросимов Н. В., Махутов Н. А. и др. / Под ред. Махутова Н. А./ Безопасность России. Правовые, социально-экономические и научно-технические аспекты. Техногенная, технологическая и техносферная безопасность. М.: МГОФ «Знание», 2018, 1016 с.*
9. *Probabilistic modeling in system engineering (Вероятностное моделирование в системной инженерии). InTechOpen, Edited by A. Kostogryzov, 2018, 279 p. URL: http://www.intechopen.com/books/probabilistic-modeling-in-system-engineering*
10. *A. Kostogryzov and V. Korolev, Probabilistic Methods for Cognitive Solving of Some Problems in Artificial Intelligence Systems (Вероятностные методы для когнитивного решения некоторых задач в системах искусственного интеллекта). Probability, combinatorics and control./ IntechOpen, 2020, pp. 3–34. URL: https://www.intechopen.com/books/probability-combinatorics-and-control*
11. *Нистратов А. А. Аналитическое прогнозирование интегрального риска нарушения приемлемого выполнения совокупности стандартных процессов в жизненном цикле систем высокой доступности. Часть 1. Математические модели и методы // Системы высокой доступности. 2021. Т.17 №3, с. 16–31, Часть 2. Программно-технологические решения. Примеры применения // Системы высокой доступности. 2022. Т.18 №2, с. 42–57.*
12. *Kostogryzov A., Makhutov N., Nistratov A., Reznikov G. Probabilistic predictive modeling for complex system risk assessments (Вероятностное упреждающее моделирование для оценок рисков в сложных системах). Time Series Analysis - New Insights. IntechOpen, 2023, pp. 73–105. http://mts.intechopen.com/articles/show/title/probabilistic-predictive-modelling-for-complex-system-risk-assessments*
13. *Хинчин А. Я. Работы по математической теории массового обслуживания. – М.: изд-во Физ. мат. лит., 1963.*
14. *Григолионис В. О сходимости сумм ступенчатых процессов к пуассоновскому // Теория вероятности и ее применения. Т.8, 1963, №2.*
15. *Гнеденко Б. В., Коваленко И. Н. Введение в теорию массового обслуживания. М.: Наука. 1987.*
16. *Матвеев В. Ф., Ушаков В. Г. Системы массового обслуживания. М.: МГУ, 1984.*
17. *Костогрызов А. И., Назаров Л. В. Пакетная обработка требований в системе с относительным приоритетом// Изв. АН СССР сер. Техническая кибернетика. 1981, №3, С. 183–187.*
18. *Балыбердин В. А. Методы анализа мультипрограммных систем. – М. Радио и связь, 1982. – 152 с.*
19. *Балыбердин В. А. Оценка и оптимизация характеристик систем обработки данных. – М.: Радио и связь, 1987. 176 с.*
20. *Костогрызов А. И., Матвеев В. Ф. Анализ применения комбинированной дисциплины обслуживания в системах реального времени // Изв. АН СССР сер. Техническая кибернетика. 1986, №6, С. 79–84.*
21. *Костогрызов А. И. Пакетная обработка заявок в режиме равномерного разделения процессора с прерыванием // Изв. АН СССР сер. Техническая кибернетика. 1987, №4, С. 88–93.*
22. *Костогрызов А. И. Класс приоритетных дисциплин с комбинированием принципов обслуживания в порядке приоритета и пакетной обработки заявок. Анализ их свойств и возможностей применения в АСУ// Анализ стохастических систем методами исследования операций и теории надежности. К.: Ин-т кибернетики им. В. М. Глушкова АН УССР, 1987. С. 52–55.*
23. *Безкоровайный М. М., Костогрызов А. И., Львов В. М. Инструментально-моделирующий комплекс для оценки качества функционирования информационных систем КОК. М.: Изд. «Вооружение. Политика. Конверсия», 2002. 304 с.*
24. *Kostogryzov A., Atakishchev O., Nistratov A., Nistratov G., Klimov S., Grigoriev L. The method of rational dispatching a sequence of heterogeneous repair works // Energetica. 2017. Vol.63, No 4, P. 154–162.*
25. *Гостев И. М., Голосов П. Е. Анализ эффективности облачной вычислительной системы, обслуживающей поток заданий с директивными сроками выполнения при множественных отказах серверов // Программная инженерия. 2023. Том 14, № 6. С. 278–284. DOI: 10.17587/prin.14.278-284*

26. Голосов П. Е., Гостев И. М. Анализ эффективности имитационных моделей облачных вычислений с использованием элементов искусственного интеллекта / Радиотехнические и телекоммуникационные системы. М. 2023. № 2. С. 29–39.
27. Golosov P. E., Ronzhin A. F. Approaches to execution of sets of tasks with random processing time in coherent computational systems / Proceedings of the International Conference on Modern stochasticity: theory and applications. Kyiv. 10–14.09.2012. С. 33.
28. Lyu, Siwei & Farid, Hany. (2005). How Realistic is Photorealistic?. Signal Processing, IEEE Transactions on. 53. 845–850. 10.1109/TSP.2004.839896.
29. Rahmouni, Nicolas & Nozick, Vincent & Yamagishi, Junichi & Echizen, I. (2017). Distinguishing computer graphics from natural images using convolution neural networks. 1–6. 10.1109/WIFS.2017.8267647.
30. Golosov P. E., Gostev I. M. Optimization of the Distribution of Hash Calculation Tasks Flow at a Priori Given Complexity / Информационные технологии. 2021. No 5. P. 242–248.

METHODOLOGICAL PROVISIONS ON PROBABILISTIC PREDICTION OF INFORMATION SYSTEMS OPERATION QUALITY. Part 2. MODELING USING «BLACK BOXES»

Kostogryzov A. I[6], Nistratov A. A.[7], Golosov P. E.[8]

Objective: *The purpose of the entire work is to help system analysts involved in assessing the quality of information systems (IS) operation during their creation, operation, modernization, development, to form the appearance of a comprehensive probabilistic prediction methodology applicable in the interests of ensuring quality and safety, justifying acceptable risks, identifying significant threats and supporting the adoption of scientifically rational system decisions to proactively counter threats in IS life cycle. The purpose of the 2nd part of the work is to detail, in the interests of probabilistic analysis of the properties characterizing information systems operation quality, the general methodological provisions (summarized in the 1st part of the article), by proposing probabilistic models represented in the form of «black boxes».*

Research methods include: *methods of probability theory, methods of system analysis. Formally, «black box» acts as a modeled system when the initial data for modeling and output results are known, but the internal detail structure of the system is unknown. The obtained results of mathematical modeling are used in the interpretation of the original IS, in the interests of which the corresponding calculations are carried out.*

Results of the 2nd part are: *models presented in the form of «black boxes» are proposed for the probabilistic analysis of the composite properties of the IS quality according to GOST R 59341-2021 «System engineering. Protection of information in system information management process».*

Scientific novelty: *The proposed models are aimed at achieving the general purpose of IS operation in various functional applications – to ensure the reliability and timeliness of providing the necessary information, completeness, validity and security (the purpose is formulated in the 1st part of the article). The use of models makes it possible to carry out assessments on a single probabilistic scale of IS operation quality under consideration and its constituent elements, represented as «black boxes».*

Keywords: *probability, model, prediction, risk, system, system analysis, threat.*

References

1. Kostogryzov A. I., Nistratov A. A. Metodicheskie polozhenija po verojatnostnomu prognozirovaniju kachestva funkcionirovanija informacionnyh sistem. Chast' 1. Obshhij podhod // Pravovaja informatika, 2024, №3. S. 13–31.
2. Kostogryzov A. I., Petuhov A. V., Shherbina A. M. Osnovy ocenki, obespechenija i povyshenija kachestva vyhodnoj informacii v ASU organizacionnogo tipa. M.: Izd. «Vooruzhenie. Politika. Konversija», 1994. 278 s.
3. Kostogryzov A. I., Lipaev V. V. Sertifikacija kachestva funkcionirovanija avtomatizirovannyh informacionnyh sistem. – M. Izd. «Vooruzhenie, politika, konversija», 1996. 278 s.
4. Kostogryzov A. I., Nistratov G. A. Standartizacija, matematicheskoe modelirovanie, racional'noe upravlenie i sertifikacija v oblasti sistemnoj i programmnoj inzhenerii. – M. Izd. «Vooruzhenie, politika, konversija», 2004, 2-e izd. 2005. 395 s.
5. Kostogryzov A. I., Stepanov P. V. Innovacionnoe upravlenie kachestvom i riskami v zhiznennom cikle sistem – M.: Izd. «Vooruzhenie, politika, konversija», 2008. – 404 s.
6. A. Kostogryzov, A. Nistratov, G. Nistratov SOME APPLICABLE METHODS TO ANALYZE AND OPTIMIZE SYSTEM PROCESSES IN QUALITY MANAGEMENT («Nekotorye prikladnye metody dlja analiza i optimizacii sistemnyh processov v upravlenii kachestvom») // InTech,

6 Andrey I. Kostogryzov, Dr.Sc., Professor, Chief Researcher, Federal Research Center «Informatics and Control» of the Russian Academy of Sciences. Moscow, Russia. E-mail: Akostogr@gmail.com
7 Andrey A. Nistratov, Ph.D., Senior researcher, Federal Research Center «Informatics and Control» of the Russian Academy of Sciences. Moscow, Russia. E-mail: Andrey.nistratov@gmail.com
8 Pavel E. Golosov, Ph.D., Director of the Institute of Social Sciences of the Russian Academy of National Economy and Public Administration, Moscow, Russia. E-mail: pgolosov@gmail.com

2012, ISBN979-953-307-778-8, 2012, pp. 127–196. http://www.intechopen.com/books/total-quality-management-and-six-sigma/some-applicable-methods-to-analyze-and-optimize-system-processes-in-quality-management
7. Abrosimov N. V., Aleshin A. V., Mahutov N. A. i dr. /Pod red. Mahutova N. A./ Bezopasnost' Rossii. Pravovye, social'no-jekonomicheskie i nauchno-tehnicheskie aspekty. Nauchnye osnovy tehnogennoj bezopasnosti. M.: MGOF «Znanie», 2015, 936 s.
8. Abrosimov N. V., Mahutov N. A. i dr. / Pod red. Mahutova N. A./ Bezopasnost' Rossii. Pravovye, social'no-jekonomicheskie i nauchno-tehnicheskie aspekty. Tehnogennaja, tehnologicheskaja i tehnosfernaja bezopasnost'. M.: MGOF «Znanie», 2018, 1016 s.
9. Probabilistic modeling in system engineering (Verojatnostnoe modelirovanie v sistemnoj inzhenerii). InTechOpen, Edited by A. Kostogryzov, 2018, 279 p. URL: http://www.intechopen.com/books/probabilistic-modeling-in-system-engineering
10. A. Kostogryzov and V. Korolev, Probabilistic Methods for Cognitive Solving of Some Problems in Artificial Intelligence Systems (Verojatnostnye metody dlja kognitivnogo reshenija nekotoryh zadach v sistemah iskusstvennogo intellekta). Probability, combinatorics and control./ IntechOpen, 2020, pp. 3-34. URL: https://www.intechopen.com/books/probability-combinatorics-and-control
11. Nistratov A. A. Analiticheskoe prognozirovanie integral'nogo riska narushenija priemlemogo vypolnenija sovokupnosti standartnyh processov v zhiznennom cikle sistem vysokoj dostupnosti. Chast' 1. Matematicheskie modeli i metody // Sistemy vysokoj dostupnosti. 2021. T. 17 № 3, s. 16–31, Chast' 2. Programmno-tehnologicheskie reshenija. Primery primenenija // Sistemy vysokoj dostupnosti. 2022. T. 18 № 2, s. 42–57
12. Kostogryzov A., Makhutov N., Nistratov A., Reznikov G. Probabilistic predictive modeling for complex system risk assessments (Verojatnostnoe uprezhdajushhee modelirovanie dlja ocenok riskov v slozhnyh sistemah). Time Series Analysis – New Insights. IntechOpen, 2023, pp. 73-105. http://mts.intechopen.com/articles/show/title/probabilistic-predictive-modelling-for-complex-system-risk-assessments
13. Hinchin A. Ja. Raboty po matematicheskoj teorii massovogo obsluzhivanija. – M.: izd-vo Fiz. mat. lit., 1963.
14. Grigolionis V. O shodimosti summ stupenchatyh processov k puassonovskomu // Teorija verojatnosti i ee primenenija. T. 8, 1963, № 2.
15. Gnedenko B. V., Kovalenko I. N. Vvedenie v teoriju massovogo obsluzhivanija. M.: Nauka. 1987.
16. Matveev V. F., Ushakov V. G. Sistemy massovogo obsluzhivanija. M.: MGU, 1984.
17. Kostogryzov A. I., Nazarov L. V. Paketnaja obrabotka trebovanij v sisteme s otnositel'nym prioritetom // Izv. AN SSSR ser. Tehnicheskaja kibernetika. 1981, №3, S.183–187.
18. Balyberdin V. A. Metody analiza mul'tiprogrammnyh sistem. – M. Radio i svjaz', 1982. – 152 s.
19. Balyberdin V. A. Ocenka i optimizacija harakteristik sistem obrabotki dannyh. – M.: Radio i svjaz', 1987. 176 s.
20. Kostogryzov A. I., Matveev V. F. Analiz primenenija kombinirovannoj discipliny obsluzhivanija v sistemah real'nogo vremeni // Izv. AN SSSR ser. Tehnicheskaja kibernetika. 1986, № 6, S.79–84.
21. Kostogryzov A. I. Paketnaja obrabotka zajavok v rezhime ravnomernogo razdelenija processora s preryvaniem // Izv. AN SSSR ser. Tehnicheskaja kibernetika. 1987, № 4, S.88–93.
22. Kostogryzov A. I. Klass prioritetnyh disciplin s kombinirovaniem principov obsluzhivanija v porjadke prioriteta i paketnoj obrabotki zajavok. Analiz ih svojstv i vozmozhnostej primenenija v ASU// Analiz stohasticheskih sistem metodami issledovanija operacij i teorii nadezhnosti. K.: In-t kibernetiki im. V.M.Glushkova AN USSR, 1987. S. 52–55
23. Bezkorovajnyj M.M., Kostogryzov A.I., L'vov V.M. Instrumental'no-modelirujushhij kompleks dlja ocenki kachestva funkcionirovanija informacionnyh sistem KOK. M.: Izd. «Vooruzhenie. Politika. Konversija», 2002. 304 s.
24. Kostogryzov A., Atakishchev O., Nistratov A., Nistratov G., Klimov S., Grigoriev L. The method of rational dispatching a sequence of heterogeneous repair works // Energetica. 2017. Vol.63, No 4, P. 154–162
25. Gostev I. M., Golosov P. E. Analiz jeffektivnosti oblachnoj vychislitel'noj sistemy, obsluzhivajushhej potok zadanij s direktivnymi srokami vypolnenija pri mnozhestvennyh otkazah serverov // Programmnaja inzhenerija. 2023. Tom 14, № 6. S. 278–284. DOI: 10.17587/prin.14.278-284.
26. Golosov P. E., Gostev I. M. Analiz jeffektivnosti imitacionnyh modelej oblachnyh vychislenij s ispol'zovaniem jelementov iskusstvennogo intellekta / Radiotehnicheskie i telekommunikacionnye sistemy. M. 2023. № 2. S. 29–39.
27. Golosov P. E., Ronzhin A. F. Approaches to execution of sets of tasks with random processing time in coherent computational systems / Proceedings of the International Conference on Modern stochasticity: theory and applications. Kyiv. 10–14.09.2012. S. 33
28. Lyu, Siwei & Farid, Hany. (2005). How Realistic is Photorealistic?. Signal Processing, IEEE Transactions on. 53. 845–850. 10.1109/TSP.2004.839896.
29. Rahmouni, Nicolas & Nozick, Vincent & Yamagishi, Junichi & Echizen, I.. (2017). Distinguishing computer graphics from natural images using convolution neural networks. 1–6. 10.1109/WIFS.2017.8267647.
30. Golosov P. E., Gostev I. M. Optimization of the Distribution of Hash Calculation Tasks Flow at a Priori Given Complexity / Informacionnye tehnologii. 2021. No 5. P. 242–248.

ПРЕДУПРЕЖДЕНИЕ КОМПЬЮТЕРНЫХ АТАК ТИПА MAN IN THE MIDDLE, СОВЕРШАЕМЫХ С ИСПОЛЬЗОВАНИЕМ ГЕНЕРАТИВНОГО ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Жарова А. К.[1], Елин В. М.[2], Аветисян Б. Р.[3]

DOI: 10.21681/2311-3456-2024-6-28-41

Целью статьи *является представление научному сообществу разработанной авторской методики выявления/предотвращения компьютерной атаки по типу «злоумышленник посередине» (MITM).*

Метод исследования: *для достижения поставленной цели авторы использовали методы математического моделирования, сравнительного анализа, табличный метод, а также методы экспериментально-теоретического уровня.*

Результат: *проведен сравнительный анализ программных решений, представленных в виде исходного кода на площадках по типу GITHUB, которые обеспечивают реализацию атаки злоумышленник посередине как в локальных, так и глобальных сетях, а также анализ некоторых методик предотвращения атаки по типу MITM, использующих сервисы искусственного интеллекта (ИИ). На основе данного анализа определены различные логические реализации атаки по типу MITM, а также представлены уязвимости информационных систем перед компьютерной атакой MITM. На основании проведенного анализа существующих методов противодействия этим атакам и выявленных слабых сторон этих методов, предложена авторская методика предотвращения атаки по типу MITM, которая включает обучение ИИ на дата-сетах, подключенных к библиотекам разных языков программирования и алгоритмизированных эвристических моделях, реагирующих на изменение логики поведения пользователей, либо активности персонального компьютера, сетевого оборудования.*

Практическая ценность *состоит в разработанной авторской методике выявления/предотвращения компьютерной атаки по типу MITM с использованием «предиктивных» сетевых технологий, которые основаны на применении нейронных сетей, обученных методами машинного обучения.*

Ключевые слова: *дата-сеты, MITM, методики предотвращения атаки, эвристические модели, поведение пользователей, предиктивные сетевые технологии.*

Введение

Развитие инженерной мысли позволило интегрировать достижения в сфере науки и техники в сферу жизнедеятельности человека, формируя тем самым новые общественные отношения. Цифровые технологии могут работать на благо человека, но нередко используются и в противозаконных целях [1]. Чем чаще человек пользуется информационными технологиями, тем больше он оставляет цифровых следов и становится более уязвимым перед злоумышленниками. Интерес злоумышленников представляют и организации, они также становятся уязвимыми перед различными информационными угрозами [2].

В соответствии с отчетом, составленным Kaspersky ICS CERT, за первое полугодие 2023[4] компьютерным атакам подверглись промышленные компании в следующих отраслях экономики.

С каждым годом в мире наблюдается рост интенсивности компьютерных инцидентов. По исследованию, проведенному компанией Positive Technologies, «утечка конфиденциальной информации стала одним из самых распространенных последствий кибератак. Ее доля в I квартале 2024 г. выросла до 72 %, тогда как за тот же период 2023 г. этот показатель составлял 59 %. За первые 2,5 месяца этого года также утекли данные 170 компаний — это 40 % от всего числа инцидентов[5] в 2023 г.»

Причем в руках злоумышленников находятся различные решения, которые позволяют им получить

1 Жарова Анна Константиновна, доктор юридических наук, профессор Финансового университета при Правительстве Российской Федерации, Москва, Россия. E-mail: anna_jarova@mail.ru

2 Елин Владимир Михайлович, кандидат юридических наук, доцент Финансового университет при Правительстве Российской Федерации; доцент кафедры информационной безопасности Московского университета МВД России имени В.Я. Кикотя, Москва, Россия. E-mail: elin_vm @mail.ru

3 Аветисян Борис Рафаелович, главный научный сотрудник, Научно-исследовательский институт образования и науки, Москва, Россия. E-mail: Boris.Avetisyan@gmail.com

4 Первое полугодие 2023 года — краткий обзор основных инцидентов промышленной кибербезопасности // https://ics-cert.kaspersky.ru/publications/reports/2023/10/05/h1-2023-a-brief-overview-of-main-incidents-in-industrial-cybersecurity/ (дата обращения 20.09.2024)

5 Информационная опасность: доля утечек личных данных выросла до 72% // https://iz.ru/1696887/elizaveta-krylova/informatcionnaia-opasnost-dolia-utechek-lichnykh-dannykh-vyrosla-do-72 (дата обращения 20.09.2024)

незаконный доступ к данным. Существуют различные методы проведения компьютерных атак, одним из таких методов является атака типа «злоумышленник посередине» (MITM). На данный момент невозможно привести статистику ущерба происходящего именно от компьютерных атак типа MITM, поскольку каждый случай атаки имеет свои уникальные характеристики и последствия. В научной литературе изучается специфика этой атаки и предлагаются различные методы ее предотвращения. Так, наиболее эффективным методом предотвращения атаки по типу MITM являются «предиктивные» сетевые технологии, которые основаны на применении нейронных сетей, обученных методами машинного обучения. Предиктивные сетевые технологии формируют прогнозы на основе анализа целого спектра данных: от анализа сетевых портов, используемых в рамках исследуемого сегмента сетевой инфраструктуры, производительности систем, до анализа доменной зоны (геолокации серверной части – «цифровой юрисдикции»). Полученные данные выступают базисом для анализа и прогнозировании инцидентов, а также для формирования вероятностной оценки угрозы утечки информации ограниченного доступа по техническим каналам связи и основаниями расчета ущерба от несанкционированного доступа третьих лиц.

Предиктивные сетевые технологии целесообразны при предотвращении атак типа MITM в компьютерных сетях. Одним из компонентов обеспечения безопасности обрабатываемых и передаваемых данных выступают правила, реализуемые в цифровых сертификатах и криптографических протоколах SSL/TLS шифрования. Решение SSL (Secure Sockets Layer) или его более современной версии TLS (Transport Layer Security) используется для защиты передаваемых данных между клиентом и сервером.

Искусственный интеллект (ИИ) используется не только для предотвращения/выявления компьютерных атак, но и в противоположных целях [3]. С применением ИИ [4] злоумышленники ускоряют процесс поиска уязвимостей.

Анализируя существующие методы противодействия компьютерным атакам и, в частности, атаке по типу MITM, авторы статьи поставили задачу разработки авторской методики выявления/предотвращения компьютерной атаки по типу MITM.

В статье проводится сравнительный анализ программных решений, представленных в виде исходного кода на площадках по типу GITHUB, которые обеспечивают реализацию атаки по типу MITM как в локальных, так и глобальных сетях, а также анализ некоторых методик предотвращения атаки по типу MITM, использующих сервисы ИИ. На основе данного анализа определяются различные логические реализации атаки по типу MITM, а далее в целях предотвращения атаки по типу MITM проводится обучение ИИ на дата-сетах, подключенных к библиотекам разных языков программирования и алгоритмизированных эвристических моделях, реагирующих на изменение как логики поведения пользователей, так и активности персонального компьютера и сетевого оборудования.

Понятие компьютерной атаки

Понятие компьютерной атаки раскрывается стандартом ISO/IEC 27000:2014[6] как «попытка уничтожения, раскрытия, изменения, блокирования, кражи, получения несанкционированного доступа к активу или его несанкционированного использования». Ученые относят компьютерную атаку к инструменту кибероперация [5], проводимых против конкретных лиц или организаций.

К классическим компьютерным атакам можно отнести атаки типа «отказ в обслуживании» (DoS) и распределенные атаки «отказ в обслуживании» (DDoS); атаки «злоумышленник посередине» (MITM); фишинг; целевые фишинговые атаки; атаки путем внедрения (SQLI и XSS); глушение, подслушивающие атаки; и атаки вредоносных программ.

Определенной новеллой выступают компьютерные атаки, основанные на технологиях ИИ, результатом которых является некорректная классификация данных, генерация синтетических данных, незаконный доступ к данным и их анализ [5]. Так, в компьютерной атаке по типу MITM субъектом может выступать ИИ, который перехватывает и анализирует передаваемый трафик. В целях противодействия такой атаке исследователи предлагают производить анализ и классификацию сетевого трафика и обнаружение аномалий, на основе которых формируются прогнозы о возможной атаке [6].

Понятие компьютерной атаки по типу «злоумышленник посередине» (MITM)

Понятие компьютерная атака по типу «злоумышленник посередине» (MITM) является собирательным, описывает ситуацию, когда субъект использует различные методики и технические решения, направленные на получение доступа к трафику и его декодировки. Данная компьютерная атака реализуется с применением алгоритмов перехвата трафика, передаваемого между двумя оконечными устройствами.

Корпорация по управлению доменными именами и IP-адресами (Internet Corporation for Assigned Names and Numbers, ICANN) в атаке «злоумышленник

6 ISO/IEC 27000:2014 Информационные технологии. Методы и средства обеспечения информационной безопасности. Системы менеджмента информационной безопасности. Общий обзор и терминология (Information technology. Security techniques. Information security management systems. Overview and vocabulary)

посередине» качестве посредника определяет как человека, так и устройство, которые имеют возможность перехватывать или модифицировать данные, пересылаемые между двумя абонентами системы связи. ICANN приводит два примера атак MITM в Интернете.

Первый — это «клонирование» или подмена точки доступа (иногда такой тип атаки именуется «злой двойник».

Второй тип атак называется «противник в браузере»[7].

ФСТЭК России связывает проведение MITM-атаки с уязвимостью реализации протокола инкапсуляции Ethernet, которая позволяет объединять заголовки. Эксплуатация данной уязвимости позволяет действовать удаленно и вызывать необходимые технические сбои с последующей реализацией атаки (MITM)»[8].

Исследователи Keeper Security считают, что MITM это тип компьютерной атаки, при «которой злоумышленник перехватывает данные, передаваемые между двумя устройствами, компьютером или мобильным терминалом, на котором запущен веб-браузер и главный сервер»[9].

Классическая MITM-атака проходит в два этапа. Первый – перехват данных, когда преступник интегрируется в среду передачи данных. Далее при помощи спуфинга реализует подмену IP-адресов, ARP[10]-сообщений, сервера доменных имен и т.д. Атаки MITM зачастую используют ARP-кэш, который представляет собой локальный кэш с назначенными IP-адресами и сопоставленными физическими уникальными идентификаторами устройств в сети (MAC-адресами). В результате реализуются задачи получения сведений о структуре исследуемой сети и сопоставления локальных идентификаторов и универсальных идентификаторов сети (MAC-адресов).

Второй этап – дешифрация, т.е. получение доступа к зашифрованным данным. Поскольку существует большое разнообразие методик проведения атаки, существуют и методики противодействия, одной из которых является анализ исходного кода (как устанавливаемых приложений, так и исследование передаваемых данных в рамках «песочницы» на виртуальной машине и без наличия выхода в открытую сеть). Такой анализ может осуществляться вручную специалистом, либо анализироваться при помощи решений, созданных на основе обученного по соответствующему направлению ИИ.

Требования по безопасности информации к средствам защиты информации от воздействий

ФЗ «О техническом регулировании»[11] в части обеспечения информационной безопасности не устанавливает требований об обязательной сертификации средств защиты информации. Требования об обязательной сертификации средств защиты информации (технические, криптографические, программные и другие средства, предназначенные для защиты сведений, составляющих государственную тайну, средства, в которых они реализованы, а также средства контроля эффективности защиты информации) определены Постановлением Правительства РФ «О сертификации средств защиты информации»[12]. Но, как мы можем отметить, эти требования касаются информационных технологий, обрабатывающих государственную тайну.

Условно, с 2018 года ФСТЭК России начала формировать требования к поставщикам средств защиты информации, при этом были определены требования к средствам защиты информации от воздействий, направленных на отказ в обслуживании информационных (автоматизированных) систем[13]. В 2020 г. ФСТЭК России сформулировала требования, устанавливающие уровни доверия к средствам технической защиты информации и средствам обеспечения безопасности информационных технологий[14].

С целью предотвращения перехвата трафика и обеспечения информационной безопасности пользователей сети ФСТЭК России определила требования о сертификации межсетевых экранов. В 2023 г. ФСТЭК России установила соответствия классов защиты многофункциональных межсетевых экранов уровня сети уровням доверия[15]. В 2024 г. в СМИ была опубликована информация, что «Росреестр рассматривает возможность заказать продукты в области межсетевых экранов нового поколения (NGFW). Стоимость проекта оценивается в 1 млрд руб. Другим крупным заказчиком NGFW является ВТБ»[16].

7 Что такое атаки типа «злоумышленник в середине» или, как их еще называют, атаки посредника (Man in the Middle Attack, MIMA)? // https://www.icann.org/ru/blogs/details/what-is-a-man-in-the-middle-attack-2-11-2015-ru (дата обращения 20.09.2024)

8 BDU:2022-05987: Уязвимость реализации протокола инкапсуляции Ethernet, связанная с возможностью объединения заголовков, позволяющая нарушителю вызвать отказ в обслуживании или реализовать атаку «человек посередине» (MITM) // https://bdu.fstec.ru/vul/2022-05987

9 Что такое атаки «злоумышленник в середине»? // https://www.keepersecurity.com/blog/ru/2023/10/16/how-to-detect-man-in-the-middle-attacks/ (дата обращения 20.09.2024)

10 ARP — протокол в компьютерных сетях, предназначенный для определения MAC-адреса другого компьютера по известному IP-адресу.

11 ФЗ «О техническом регулировании» от 27 декабря 2002 г. № 184-ФЗ // СЗ РФ 2002. № 52 (Ч. 1). Ст. 5140.

12 Постановление Правительства РФ «О сертификации средств защиты информации» от 26 июня 1995 г. № 608 // СЗ РФ 1995. № 27. Ст. 2579.

13 Требования по безопасности информации к средствам защиты информации от воздействий, направленных на отказ в обслуживании информационных (автоматизированных) систем (утв. приказом ФСТЭК России от 30.07.2018 N 132) (Документ опубликован не был) // СПС «КонсультантПлюс».

14 Требования по безопасности информации, устанавливающие уровни доверия к средствам технической защиты информации и средствам обеспечения безопасности информационных технологий (утв. приказом ФСТЭК России от 02.06.2020 N 76) (Документ опубликован не был) // СПС «КонсультантПлюс».

15 Требования по безопасности информации к многофункциональным межсетевым экранам уровня сети (утв. приказом ФСТЭК России от 07.03.2023 № 44) (Документ опубликован не был) // СПС «КонсультантПлюс».

16 Росреестр закажет средства защиты информации на миллиард // https://www.cnews.ru/news/top/2024-07-08_rosreestr_zakazhet_sredstva (дата обращения 20.09.2024)

Некоторые методики выявления атаки MITM

Поскольку атака MITM направлена на получение несанкционированного доступа, то для обнаружения этой атаки могут использоваться сигнатурные и эвристические анализаторы, входящие в состав систем обнаружения атак (IDS). Так, исследователи в 2017 г. предположили, что со временем может состояться переход к эвристическим решениям в средствах обнаружения/предотвращения вторжений (IDS/IPS). В настоящее время применяются как сигнатурные, так и эвристические анализаторы, поскольку обе модели обнаружения атак MITM имеют свои сильные и слабые стороны в зависимости от конкретной ситуации.

В основе работы эвристического анализатора заложена схема, в которой в режиме обучения формируются «правильные» шаблоны поведения системы, а в режиме анализа – обнаруживаются отклонения от этих шаблонов. За счет этого эвристический анализатор может обнаружить вредоносную активность, не попавшую ни под какую конкретную сигнатуру.

Исследования в области разработки современных систем обнаружения компьютерных атак показывают, что методы ИИ и машинного обучения могут быть применимы в области обнаружения/предотвращения атак по типу MITM [7]. Одним из главных преимуществ эвристических анализаторов IDS, использующих методы машинного обучения, является их способность выявлять новые виды атак, в отличие от сигнатурных анализаторов [8]. Основным компонентом эвристического анализа вредоносных программ является искусственная нейронная сеть в виде многослойного перцептрона с иммунным обучением [9]. Для решения задачи обучения авторы статьи использовали модель кодирования настраиваемых параметров в виде адаптивного структурированного мультиантитела, что позволило уменьшить количество нейронов в скрытом слое и устранить, таким образом, избыточность нейронной сети.

Другие исследователи [9] пришли к выводу, что большинство используемых методов глубокого обучения в области обнаружения вторжений показывают хорошие результаты, независимо от того, используется какой-то один вид нейронной сети (например, recurrent neural network – RNN) или их сочетание (например, convolutional neural network CNN-RNN). Сочетания призваны устранить недостатки конкретных методов или в целом улучшить степень автоматизации всего процесса выявления атак. Использование методов глубокого обучения с учетом всех предварительных и вспомогательных приемов является более эффективным, чем просто использование этих методов перед классическими методами машинного обучения. Нейронные сети, особенно при их комбинировании с другими, не относящимися к глубокому обучению методами, обычно демонстрируют хорошие результаты.

Авторы статьи [10] подчеркивают, что популярностью у исследователей пользуются RNN и CNN и их сочетания, но все чаще в новых исследованиях разработчики обращаются к таким архитектурам технологий как автокодировщики, графовые нейронные сети, трансформеры.

Сетевая система обнаружения вторжений с применением машинного обучения позволяет выявлять широкий спектр веб-атак, производимых на сетевом уровне [11]. Другие исследователи предлагают для решения этой задачи разработанный ими алгоритм выявления атак по типу MITM для статически назначаемых IP-адресов хоста, а также IP-адресов, назначаемых через DHCP. Этот алгоритм, как пишут авторы статьи, они реализовали с использованием асинхронного метода диспетчеризации для снижения затрат на производительность [12].

Однако, несмотря на то что предлагаются различные методики выявления/предотвращения атаки по типу MITM, наиболее эффективным методом предотвращения атаки по типу MITM являются предиктивные сетевые технологии, которые основаны на алгоритмах искусственного интеллекта и машинного обучения.

Обзор некоторых российских решений, анализаторов исходного кода

Идет постоянный поиск наиболее удачных решений, которые обсуждаются как на теоретическом, так и на практическом уровнях. Разнообразие атак по типу MITM эксплуатирует уязвимости информационных технологий, наиболее сложными для обнаружения являются программные закладки, и эффективность этого зависит от уровня их встраивания. Программные закладки, встроенные на этапе производства, практически не поддаются выявлению[17]. Существуют три основных типа анализаторов исходного кода программного обеспечения (ПО) на наличие уязвимостей и закладок:

1. Анализаторы кода веб-приложений, которые помогают предотвратить уязвимости на веб-сайтах.
2. Анализаторы встраиваемого кода, которые позволяют найти проблемы в исходных текстах модулей, расширяющих функциональность корпоративных систем, таких как 1С, CRM и SAP.
3. Анализаторы исходного кода на других языках программирования, не связанных с бизнес- и веб-приложениями.

Наибольшего результата в области анализа исходного кода, позволяет достичь применение двух

17 Кое-что о закладках, или Как АНБ следит за пользователями // https://www.cryptopro.ru/en/blog/2015/11/10/koe-chto-o-zakladkakh-ili-kak-anb-sledit-za-polzovatelyami (дата обращения 20.09.2024)

основных технологий анализа – динамический анализатор (DAST – Dynamic Application Security Testing) и статический анализатор (SAST – Static Application Security Testing), разновидностью которого является бинарный анализ.

Предлагаемое на российском рынке компанией Solar решение Solar appscreener, технологическая основа которого представлена на рисунке 1, позволяет применять технологии динамических и статических анализаторов кода.

Рис. 1. Представленная разработчиками технологическая основа «Solar appScreener»

В основе подхода, реализованного Solar, применяется единая технологическая платформа, обеспечивающая комплексный анализ безопасности приложений[18]. В нее входит ядро платформы, технологические модули, коннекторы, единый интерфейс для удобного управления сканированиями, корреляция результатов разных видов анализа и функция получения подробного отчета. В технологическом решении используется технология Fuzzy Logic Engine для сокращения ложных срабатываний.

Сканер уязвимостей в Yandex выступает еще одним российским решением в области анализа исходного кода. При этом он позволяет хранить и распространять Docker-образы[19], размещаемые в отказоустойчивом хранилище. Для всех данных настроена автоматическая репликация при редактировании, создании и удалении Docker-образа меняется каждая копия. Docker-образы передаются по протоколу HTTPS. Сканер уязвимостей анализирует Docker-образ и сравнивает его содержимое с базами уязвимостей CVE[20].

Следующим решением является система обнаружения вторжений, осуществляющая мониторинг и обработку событий внутри хоста – ViPNet IDS HS от INFOTECS, которое использует сигнатурный и эвристический методы анализа атак на основе правил и сигнатур, разработанных в России. За счет централизованного управления агентами, настройками и группами правил на хостах администраторы по информационной безопасности могут оперативно реагировать на события безопасности в сети[21].

Статическим анализатором исходного кода для поиска ошибок и уязвимостей в программах на языке C, C++ и C# выступает анализатор PVS-Studio разработанный компанией ООО «СиПроВер»[22].

На рынке также представлены анализаторы с открытым исходным кодом, например, SonarQube, как платформа для непрерывной оценки качества кода путем статического анализа и измерения качества программного кода. В возможности платформы входит анализ кода и поиск ошибок согласно правилам стандартов программирования некоторых языков[23].

Обзор вредоносных систем ИИ предназначенных для совершения компьютерных атак различного типа

В настоящее время разработаны и применяются ряд систем ИИ изначально предназначенных для совершения компьютерных атак различного типа, некоторые наиболее часто применяемые представлены в таблице 1. По каждой из систем проводились исследования, направленные на изучение возможностей и особенностей ее применения для осуществления незаконного анализа данных и нецелевого использования информационных систем.

Вредоносные системы используются для совершения компьютерных атак в различных сферах, в частности в различных сферах экономики. Подтверждение этому представлено в табл. 2.

Основным методом осуществления представленных атак является состязательное машинное обучение (Adversarial Machine Learning (AML) как метод, основанный на машинном обучении, суть которого заключается в использовании существующих «слепых зон» между обрабатываемыми в процессе обучения модели совокупности данных. При обучении вредоносного ИИ определяются слабые стороны защищаемой системы и вносятся небольшие изменения в ее массивы данных. В связи с этим, в защищаемой модели не формируются устойчивые связи между целевыми значениями, что в дальнейшем приводит к неправильным классификациям с пересечением границы принятия решения и ошибочного

18 SOLARAPPSCREENER // https://rt-solar.ru/products/solar_appscreener (дата обращения 20.09.2024)

19 Шаблон (исполняемый пакет), из которого создаются Docker-контейнеры. Образ содержит всё необходимое для запуска приложения, помещённого в контейнер: код, среду выполнения, библиотеки, переменные окружения и конфигурационные файлы.

20 Yandex Container Registry // https://yandex.cloud/ru/services/container-registry?utm_source (дата обращения 20.09.2024)

21 О продукте //https://infotecs.ru/products/vipnet-ids-hs-versiya-1/#:~:text (дата обращения 20.09.2024)

22 Как PVS-Studio ищет ошибки: методики и технологии // https://habr.com/ru/companies/pvs-studio/articles/319382/ (дата обращения 20.09.2024)

23 Keep AI generated code clean // https://www.sonarsource.com/products/sonarqube/ (дата обращения 20.09.2024)

Таблица 1.

Инструменты на базе искусственного интеллекта, использующие анализ данных для совершения компьютерных преступлений

Наименование	Область применения
DeepHack	Инструмент на базе искусственного интеллекта для создания шаблонов атак с инъекциями для приложений баз данных[24]
DeepLocker	Инструмент на базе искусственного интеллекта, который эмулирует APT для запуска сложных кибератак[25]
GyoiThon	Инструмент на базе искусственного интеллекта для сбора информации и автоматической эксплуатации[26]
EagleEye	Инструмент на базе искусственного интеллекта для разведки информации в социальных сетях с использованием алгоритмов распознавания лиц[27]
Malware-GAN	Инструмент на базе искусственного интеллекта, используемый для создания вредоносного ПО, которое может обходить механизмы обнаружения безопасности[28]
uriDeep	Инструмент на базе искусственного интеллекта, который генерирует поддельные домены для использования в различных сценариях атак[29]
Deep Exploit	Инструмент на базе искусственного интеллекта, который автоматизирует Metasploit для сбора информации, сканирования и последующей эксплуатации[30]
DeepGenerator	Инструмент на базе искусственного интеллекта для создания шаблонов атак с инъекциями для веб-приложений

Таблица 2.

Вредоносные алгоритмы, используемые для подмены данных и в целях обхода решений на основе ИИ

Объект воздействия	Способ воздействия
Дорожные знаки	Неправильная классификация дорожного знака алгоритмами ИИ может привести к дорожно-транспортным происшествиям на автономных автомобилях[31]
Данные медицинских изображений	Неправильная классификация медицинских отклонений алгоритмами ИИ может привести к ложной диагностике состояния здоровья [11].
Данные изображений лица	Неправильная классификация изображений лиц может привести к аутентификации [13].
Цифровая рекомендация системы	Внесение ложных данных алгоритмами ИИ может привести к неверным рекомендациям [14].
Данные КТ сканирования	Неправильная классификация подделанных 3D-изображения компьютерной томографии может привести к ложной диагностике [15].
Речевые аудиоданные	Состязательная атака на голосовую активацию персональной помощи может нарушить ее функциональность [16].
Системы обнаружения сетевых вторжений	Генерация вредоносного трафика для обхода защиты систем обнаружения сетевых вторжений на базе искусственного интеллекта [17].

отнесения данных к другому классу. В дальнейшем другой ИИ при анализе защищаемой системы, не покажет наличие вредоносных данных, поскольку произошла подмена классификации данных.

Ряд авторов [16], анализируя этап машинного обучения, выделяют четыре основных направления атак AML:

- атаки на решение классификатора, включая отравляющие (причинные) атаки на этапе обучения и исследовательские (уклоняющиеся) атаки обученной модели на этапах тестирования;
- атаки либо на целостность модели, приводящие к неправильной классификации, либо на пригодность модели при наличии высокой частоты неправильных классификаций;
- целенаправленные атаки, когда состязательные выборки нацелены на достижение определенного целевого значения, или неизбирательные атаки, когда выборки не нацелены на определенное целевое значение;
- атаки на конфиденциальность, целью которой выступает извлечение информации из классификатора.

Иной подход к классификации атак предлагается на основе:

- сложности по критерию последствий, варьирующихся от незначительного снижения достоверности прогнозов модели до неправильной классификации всех невидимых точек данных;

24 Bishopfox/deephack: Poc code from def con 25 presentation. (дата обращения 20.09.2024)

25 Cyberwarefare/deeplocker: Deeplocker – deep learning based malware. (дата обращения 20.09.2024)

26 Gyoisamurai/gyoithon: Gyoithon is a growing penetration test tool using machine learning. (дата обращения 20.09.2024)

27 Thoughtfuldev/eagleeye: Stalk your friends. find their instagram, fb and twitter profiles using image recognition and reverse image search. (дата обращения 20.09.2024)

28 Yanminglai/malware-gan: Realization of paper: «generating adversarial malware examples for black-box attacks based on gan» 2017. (дата обращения 20.09.2024)

29 Mindcrypt/urideep: Unicode encoding attacks with machine learning. (дата обращения 20.09.2024)

30 Machine learning security/deepexploit at master · 13o-bbr- bbq/machine learning security. (дата обращения 20.09.2024)

31 Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In Proceedings of the 2017 ACM on Asia conference on computer and communications security, pages 506–519. ACM, 2017.

- полученного противником знания, например по типу «атака белого ящика»[32], в целях получения злоумышленником знаний об обучающей модели (ее архитектуре, сетевом трафике, который она анализирует, и ее функциям, которые используются для поддержки обучения) [17].

AML атаки исследователи предлагают классифицировать как таргетированные, которые направлены на изменение предсказания классификатора к определенному классу. Исследование особенностей осуществления автоматически генерируемых AML атак, позволило прийти ученым к выводу, что система защиты от таких атак может быть разработана на основании анализа алгоритмов машинного обучения при применении состязательных выборок.

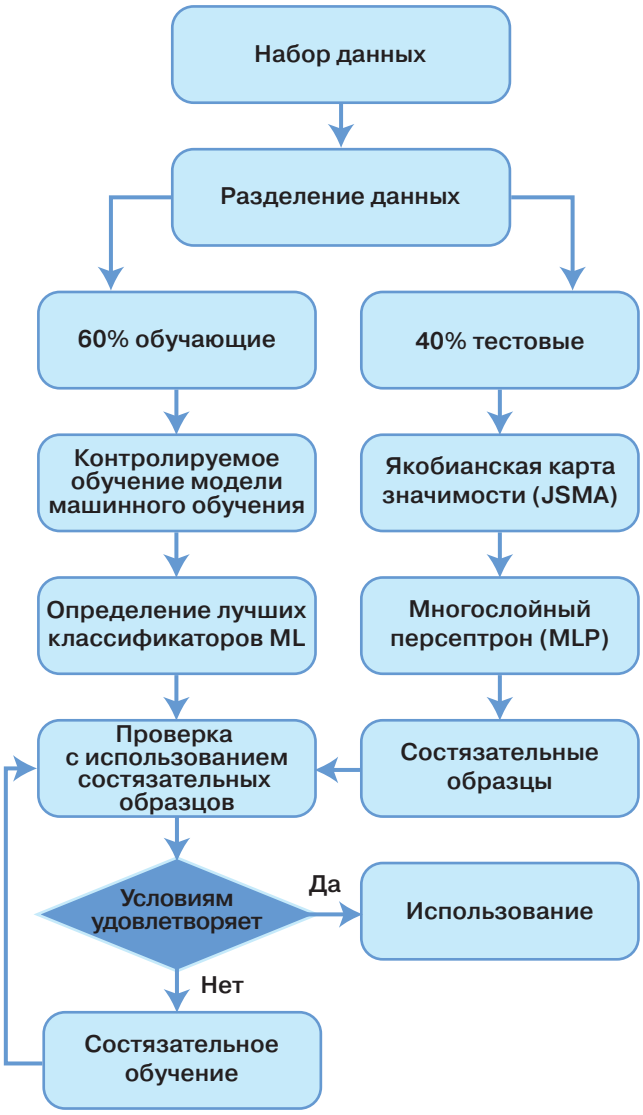


Рис. 2. Применение состязательных выборок для построения системы защиты от атак

32 Противоположностью выступает атака по типу черного ящика, когда у противника нет информации о внутренней работе целевой модели.

Для обнаружения вторжений в систему ученые предлагают разделять данные на обучающий и тестовый наборы, в соотношении – 60 % и 40 % соответственно (рис. 2). Далее производится оценка моделей машинного обучения под наблюдением с установлением наиболее эффективных моделей. Производится генерация состязательных выборок с использованием метода карты значимости на основе метода Якобиана, оценивается производительность системы, обученной на сгенерированных состязательных выборках. Процент состязательных выборок включается в обучающие данные и производится повторное обучение и оценка моделей [17].

Различные методы генерации состоятельных выборок целесообразно классифицировать по сложности, скорости генерации данных, и их производительности.

Наиболее простой (также наиболее трудоемкий и наименее точный) подход заключается в ручном изменении входных точек данных. К популярным методам автоматической генерации возмущенных выборок относят метод быстрого градиентного знака (Fast Sign Gradient Method, FGSM) и метод использования карты значимости на основе метода Якобиана (Jacobian saliency map, JSMA). Оба метода используют алгоритм, согласно которому при добавлении небольших изменений (δ) к исходной выборке (X) результирующая выборка X^* может демонстрировать состязательные характеристики $X^* = X + \delta$. Оба метода также обычно применяются при использовании предварительно обученного многослойного перцептрона (Multilayered perseptron, MLP) в качестве базовой модели для генерации состязательной выборки.

Метод FGSM воздействует на входные данные путем добавления определенного количества возмущения, когда знаки от функции градиента исходной функции потерь умножаются на некоторый ε. Шум возмущения вычисляется градиентом функции стоимости J по отношению к входным данным. Пусть θ представляет параметры модели, x — входные данные для модели, y — метки, связанные с входными данными, ε — значение, которое представляет степень применяемого шума, а $J(\theta,x,y)$ — функция стоимости, используемая для обучения целевой нейронной сети.

$$X' = X + e * sign(\nabla_x J(X, ytrue))$$

В JSMA методе считается прямая производная, на основании чего строится карта градиентов. На карте каждому параметру объекта соответствует критерий и его удельный вес, направленный на изменение конечного результата работы алгоритма. Тем самым, метод позволяет изменить как можно меньше параметров в атакуемом объекте.

Таблица 3.

Развитие функционала вредоносной GAN по некоторым отраслям экономики

Год	Название	Способ воздействия	Тип данных
2016	TextGAN	Синтетическая генерация текста посредством состязательного обучения	Текстовый
2017	FM-GAN	Генерация синтетического текста с помощью состязательных признаков	Текстовый
2017	MidiNet	Генерация синтетического звука	Аудио
2017	Age-cGAN	Предсказание возраста лица с помощью условных генеративных состязательных сетей	Визуальный
2017	CVAE-GAN	Генерация синтетического изображения лица	Визуальный
2017	SenseGen	Модель глубокого обучения для генерации синтетических данных датчиков	Текстовый
2018	WGAN	Генерация синтетического изображения МРТ мозга	Визуальный
2018	ACGAN	Генерация синтетического медицинского изображения печени	Визуальный
2018	Predestrian Synthesis GAN	Генерация синтетических данных пешеходов	Визуальный
2018	HP-GAN	Генерация синтетических данных для прогнозирования движения человека	Визуальный
2018	VAE-GAN	Генерация синтетического видео из текста	Визуальный
2018	WaveGAN	Состязательный синтез звука	Аудио
2019	DermGAN	Генерация синтетического изображения кожи	Визуальный
2019	CT-GAN	Генерация синтетического медицинского изображения МРТ	Визуальный
2019	X2CT-GAN	Генерация синтетического медицинского изображения рентгена	Визуальный
2020	D-NET [20]	Генерация биометрических данных радужной оболочки глаза	Визуальный
2021–2022	GPT-Chatbot[33]	Генерация предвзятого, неэтичного и опасного материала	Визуальный, текстовый, аудио
2023	ChatGPT от OpenAI[34]	Анализ запросов сотен миллионов людей по всему миру и сопоставление их с данными, снимаемыми с конечных устройств, в том числе с информацией о транзакциях, выполненных с помощью Apple Pay, геолокации, голосовыми командами и тысячами других дата-маркеров	Визуальный, текстовый, аудио
2024	Midjourney[35]	Использование дипфейков (поддельных изображений, видео и аудио) для манипуляций общественным мнением и дискредитации соперников	Визуальный, текстовый, аудио

Якобиан используется для вычисления карты значимости, которая определяет какие особенности входных данных являются наиболее релевантными для модельного решения. Эти характеристики, если их изменить, скорее всего, повлияют на классификацию целевых значений.

Учитывая, что методу JSMA может потребоваться несколько итераций для генерации состязательных выборок, FGSM быстрее в вычислительном отношении, несмотря на то что он изменяет каждую функцию. Кроме того, в отличие от FGSM, JSMA является более сложным подходом, но наиболее точно представляет

33 Администрация президента США выпустила 5 положений о защите людей от ИИ // https://www.tadviser.ru/index.php/Статья:Риски_использования_искусственного_интеллекта#2019:_.D0.A1.D0.B5.D0.BA.D1.81.D0.B8.D0.B7.D0.BC_.D0.B8_.D1.88.D0.BE.D0.B2.D0.B8.D0.BD.D0.B8.D0.B7.D0.BC_.D0.B8.D1.81.D0.BA.D1.83.D1.81.D1.81.D1.82.D0.B2.D0.B5.D0.BD.D0.BD.D0.BE.D0.B3.D0.BE_.D0.B8.D0.BD.D1.82.D0.B5.D0.BB.D0.BB.D0.B5.D0.BA.D1.82.D0.B0._.D0.9F.D0.BE.D1.87.D0.B5.D0.BC.D1.83_.D1.82.D0.B0.D0.BA_.D1.81.D0.BB.D0.BE.D0.B6.D0.BD.D0.BE_.D0.B5.D0.B3.D0.BE_.D0.BF.D0.BE.D0.B1.D0.BE.D1.80.D0.BE.D1.82.D1.8C.3F

34 Ваш карманный манипулятор: чем опасен генеративный ИИ в смартфонах // https://trends.rbc.ru/trends/industry/6698fef79a79472609486cff?from=copy

35 Искусственный интеллект и генеративные инструменты меняют американскую политику // https://www.securitylab.ru/news/538553.php (дата обращения 20.09.2024)

атаки, поскольку он в течении длительного времени пошагово изменяет небольшой процент функций. В связи с этим, точность JSMA в значительной степени зависит от количества входных функций. Чем больше пространство признаков, тем больше итераций требуется для определения наиболее успешного подхода при генерации состязательных выборок, влияющих на производительность модели.

Традиционно для выявления атак используются алгоритмы Naive Bayes, Random Forest, SVM, и J4. Наиболее современными инструментами обнаружения атак по типу AML являются Recurrent Neural Networks.

Вредоносный ИИ может применяться и в такой модели нейронной сети как генеративно-состязательная сеть (generative adversarial network, GAN). В ней обучаются одновременно две сети (одна – генерация изображений, вторая – отраслевая визуализация). Архитектура GAN включает генератор и дискриминатор, каждый представляет собой сети с разным задачами. Генератор изучает распределение данных и генерирует образцы для сети дискриминатора. Дискриминатор определяет происходит ли выборка генератора из исходных данных или из сети генератора, на его вход поступают два типа выборок: из исходных данных и сгенерированные сетью-генератором.

При этом функция GAN заключается в генерации данных (визуальных, текстовых и аудиальных) с помощью приложений, например, рисования видео, синтеза звука, суперразрешения, интеллектуального анализа текста и синтеза обучающих данных для обучения других глубоких сетей. Поскольку эта технология является относительно недорогой, она применяется как в различных отраслях экономики, так и во вредоносных целях. Так, например, функционал GAN может использоваться во вредоносных целях, некоторые из существующих по отраслям экономики представлены в таблице 3.

Во время обучения генератор пытается создать более реалистичные образы, чтобы обмануть дискриминатор, в то время как дискриминатор пытается отличить исходные и синтетические образы. Обучение GAN осуществляется сквозным образом. Предполагается, что сеть будет обучена, когда неточность генератора (неудачная попытка обмануть сеть дискриминатора) будет равна неточности сети дискриминатора (отсутствие дискриминации между реальным и синтетическим образцом). Однако практически очень сложно установить такое равновесие, поскольку функции неточности колеблются вокруг положения равновесия. Обычно через несколько сотен циклов сгенерированные данные проверяются визуально или с помощью соответствующей метрики.

Некоторыми исследователями [19] предлагается использовать облачную инфраструктуру для идентификации атаки по типу MITM, осуществляемой с использованием ИИ. Облачная инфраструктура позволяет предотвратить атаки и создать защищенный административный центр на основании трех показателей: энтропия IP-адреса, местоположение порта и скорость поступления данных. На основании этих показателей ИИ вычисляет вероятность атаки по типу MITM.

Модель использования ИИ в целях выявления аномальных активностей

На эффективность установления субъектов MITM-инцидентов влияет, в том числе, «цифровая юрисдикция», а именно расположение основной серверной части атакующего в рамках контролируемой сетевой инфраструктуры государства или группы государств, в которых возможен сбор и анализ информации, в частности о крипто-транзакциях.

Зачастую лицо, реализующее атаку по типу MITM, использует виртуальной сервер – VPS (Virtual Private Server) на территории неконтролируемой государственными органами. На сервере VPN «развернута» виртуальная частная сеть на основе технологии VPN и маскируется структура сети через стек правил передачи данных по типу NAT, посредством которого реализована процедура преобразования IP-адреса(ов) узла(ов) локальной сети, либо удаленного тунелируемого узла.

При «многоступенчатой структуре», т. е. использовании ряда задействованных в инициализации конкретной атаки терминалов и локальных сетей, следует устанавливать всю цепочку задействованных в событии. активных элементов.

Такой цепочкой может быть: использование на оконечном устройстве – сервере, мобильном терминале (сотовом телефоне), ноутбуке, стационарном персональном компьютере, планшете или айпаде. Иными словами, на любом устройстве с сетевым интерфейсом (сетевой картой) и возможностью использования сетевых протоколов (правил передачи данных), позволяющих реализовывать функцию выхода в глобальную сеть Интернет. А также получение данных о структуре подсетей, классов, а также об организации, которой принадлежит исследуемый идентификатор (IP-адрес), о диапазоне адресов, которым владеет организация. Это возможно сделать, используя специализированное программное обеспечение (сетевые анализаторы), а также алгоритмы, реализованные в функционале терминальных команд по типу «look up».

Основной процедурой в механизме анализа выступает технология проверки сетевых пакетов –

DPI (Deep Packet Inspection) с помощью которой представляющий интерес «аномальный трафик» записывается в «логи» (журналы событий) в соответствии с руководящей документацией и внутренними инструкциями.

В случае шифрования OpenVPN его отличительный маркер отслеживается посредством DPI. Функция VPN, позволяющая скрыть зашифрованный трафик OpenVPN, имитируя его в обычный интернет-трафик («обфускация»), реализуется путем удаления данных, связанных с VPN из пакета OpenVPN, и назначения транслируемому шифрованному трафику порта 443, изначально предназначенного для передачи трафика по протоколу HTTPS. После добавления экспорта MTU через API (Application Programming Interface) и обновления сигнатур появляется возможность установить пользователей VPN-протоколов, proxy[36], а также выявить смену User-Agent[37], что позволяет установить уникальные идентификаторы оконечных устройств третьих лиц.

Изучив разные модели выявления неправомерной деятельности в Сети, авторы предлагают систему выявления инцидентов атак по типу MITM, а также несанкционированного доступа к ключевым элементам инфраструктуры распределенных компьютерных сетей (далее по тексту: «Система»). Основной целью создания системы являлась отработка на практике концепции использования ИИ при решении задачи выявления инцидентов компрометации в распределенных компьютерных сетях. Система предназначена для выявления инцидентов в процессе анализа трафика данных с установлением ключевых элементов инфраструктуры компьютерных сетей. Она может быть использована при проверке компьютерных сетей и их элементов на предмет установления инцидентов компрометации – несанкционированного доступа к компонентам критической инфраструктуры.

Особенностью системы является адаптивное восприятие графического интерфейса с логическим представлением структуры локальной сети и визуальным представлением потоков данных анализируемой сети. Ней ронная сеть позволяет на основе атрибутов, полученных из трафика, оценить риск несанкционированного доступа.

Система предназначена для использования в информационно-аналитической деятельности и позволяет автоматизировать труд человека посредством применения технологий нейронных сетей для выявления несанкционированного использования вычислительных мощностей.

Критериями эффективности в этом случае являются:

- автоматизация процесса выявления признаков несанкционированного использования вычислительных мощностей компьютерных сетей;
- оперативность принятия решений на достаточно сформированном перечне признаков, предоставленных со стороны «высоко» (точность идентификация событий, а также поддержка при принятии решений не ниже 99 % по поставленным задачам) обученной нейронной сети;
- повышение эффективности противодействия несанкционированному использованию вычислительных мощностей компьютерных сетей.

Описание системы

1. Сценарии использования

Сценарий использования системы предполагает выполнение последовательности действий Оператора с применением смежных систем. Диаграмма сценариев использования представлена на рисунке 3.

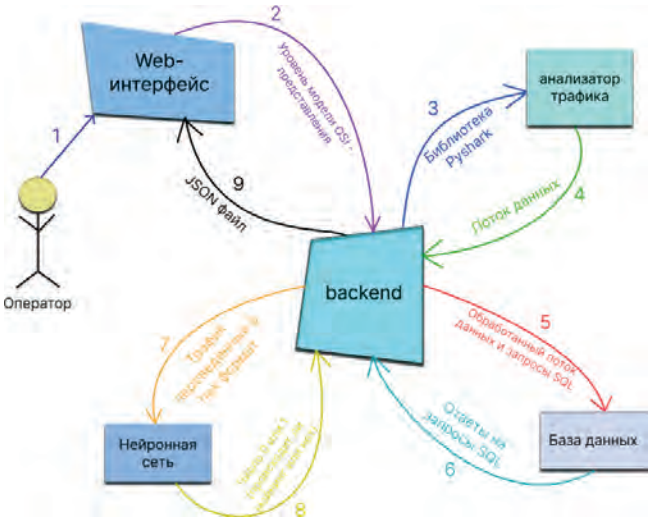


Рис. 3. Диаграмма сценариев использования

Оператор, с целью получения информации о трафике в компьютерных сетях, обращается к приложению, после чего действия осуществляются по следующему алгоритму:

1. Обращение оператора к Web-интерфейсу;
2. Запрос с Web-интерфейса к backend-у для получения информации;
3. До того, как backend отправит информацию Web-интерфейсу, ему необходимо выполнить несколько этапов:
   1) провести анализ трафика «перехватываемых» пакетов с данными;
   2) обработать полученные пакеты на сервере;
   3) сформировать и взаимодействовать с базой данных (PostgreSQL) путем пополнения обработанного на backend трафика;

36 Сетевой «посредник» между узлами
37 Идентификатор браузера

4) отправить полученные с backend обработанные на нейронную сеть, которая в свою очередь классифицирует полученный трафик;

По завершении цикла с backend-а сведения передаются на Web-интерфейс.

2. Архитектура системы

2.1. Описание Web-интерфейса

Система представлена в виде Web-приложения, реализованного на языках HTML, CSS и JavaScript. Используются библиотеки jQuery и anychart-graph. Кроссплатформенность системы обеспечена возможностью запуска на ОС Windows, macOS и семейства Unix-подобных операционных системах на базе ядра Linux.

Инициализируя запуск программного обеспечения посредством браузера, пользователю представляется граф анализируемой локальной сети, на котором отображены: количество устройств, их тип и роль, где основной особенностью разрабатываемого решения выступает возможность пользователя выявить признаки несанкционированного использования вычислительных мощностей конечных устройств в локальной сети.

Взаимодействие с серверной частью осуществляется в виде обмена данными. С сервера поступает файл в формате JSON, в котором содержится информация о состоянии сети. В свою очередь пользователь отправляет на сервер команды для взаимодействия с программой.

2.2. Описание сервера

Для запуска сервера используется фреймворк Django на языке Python. После производится загрузка на сервер данных с frontend и анализ сведений, поступающих из протокола stratum. Указанный протокол, используется большинством пулов для связи между заинтересованным лицом (атакующим субъектом) и сервером пула. Он состоит из набора инструкций, которые сервер может отправить «атакующему субъекту», и другого набора запросов, которые заинтересованное лицо может отправить на сервер. Рассматриваемый протокол реализован поверх TCP. Следует также отметить отсутствие функционирующих портов с вышеуказанным протоколом, и отсутствие возможности установления связей между слоями.

Рассматриваемый протокол - stratum использует формат JSON для всех своих методов и, как правило, использует вызовы:

– *subscribe,*
– *authorize,*
– *extranonce.subscribe,*
– *и submit.*

От сервера к клиенту идут:

– *set_difficulty*
– *и notify.*

Алгоритм взаимодействия пользователя и серверной части можно представить следующим образом:

- authorize: аутентификация субъекта на сервере.
- subscribe: запрос на сбор значимых данных.
- extranonce.subscribe: инициализации атаки по типу MITM.

Вызовы по форме взаимодействия Сервер — Клиент:

1) notify: сервер отправляет всю информацию, необходимую для запуска текущего блока, включая пользовательский идентификатор, используемый в качестве идентификатора при отправке хэша предыдущего блока.
2) set_difficulty: устанавливает сложность идентификации события.

При помощи библиотеки Pyshark осуществляется перехват пакетов в локальной сети, в которых используется протокол TCP, из которых берутся значения полезной нагрузки. IP-адреса устройств с такими пакетами отправляются на Web-интерфейс.

Также все IP-адреса сравниваются с базой данных PostgreSQL, в которой находятся адреса доменов пулов. Если найдено совпадение, можно сделать вывод, что на данном устройстве есть вероятность несанкционированного подключения и утечки информации по техническому каналу связи.

2.3. Сетевой анализатор как модуль комплекса, используемого в качестве поддержки для принятия решений

Модулем, используемым в качестве поддержки в принятии решений в рамках механизма анализа сетевого трафика, является Wireshark – это программное обеспечение имеющее графический пользовательский интерфейс и широкий спектр инструментария по сортировке и фильтрации. Он тем самым предоставляет возможность для оператора просматривать проходящий по сети трафик в режиме реального времени.

Модуль распространяется под свободной лицензией GNU GPL и использует для формирования графического интерфейса кроссплатформенную библиотеку GTK+. Существуют версии для большинства UNIX-подобных систем, в том числе GNU/Linux, Solaris, FreeBSD, NetBSD, OpenBSD, macOS, а также для Windows.

Данный модуль выступает в качестве поддержки принятия решений в вопросах обеспечения безопасности компонентов критической информационной инфраструктуры, идентификации аномальных процессов, возникающих при инициализации посторонних устройств в контролируемом сегменте обслуживаемой сети, с точностью идентификации инцидентов компрометации (о совокупности признаков) ~90 %.

2.4. Описание базы данных

Для хранения информации была использована база данных PostgreSQL. Всего будут использованы две базы данных:

- первая будет хранить в себе домены pool и их IP-адреса,
- вторая база данных будет содержать трафик, получаемый сервером, в котором будут известны IP источника и IP назначения, порт источник и порт назначения, а также количество передаваемой информации и время.

Изначально «используется» трафик, без признаков реализации атаки MITM. Далее он сравнивается с подозрительным трафиком. Эти данные запрашивает backend с помощью языка запросов SQL.

2.5. Описание нейронной сети

Мы используем нейросеть с реализованной сверточной моделью, состоящей из разных видов слоев: сверточные слои, субдискретизирующие слои и слои «обычной» нейронной сети – перцептрона. Первые два типа слоев, чередуясь между собой, формируют входной вектор признаков для многослойного перцептрона. Сверточные сети являются наиболее эффективными решениями при анализе конвертированного трафика.

Разработанная Система по выявлению инцидентов по типу MITM выступает в качестве MVP (Minimum Viable Product), ее функции позволяют отследить: преобразование нагрузки сетевого трафика из HEX в изображение; проанализировать полученное изображение с помощью сверточной модели нейронных сетей; получить ответ серверной части приложения коэффициентом идентичности исходного трафика с аномальным (исследуются также порты).

Разработанное решение (в виде MVP) включает преобразование необработанного сетевого трафика, собранного с помощью инструмента – сетевого анализатора трафика. Точность выявления инцидентов по совокупности признаков составляет около ~90 %.

Литература

1. *Жарова, А. К. Обеспечение права на доступ к Интернету и забвение в цифровом пространстве Российской Федерации / А. К. Жарова, В. М. Елин // Мониторинг правоприменения. – 2021. – № 2(39). – С. 48–53. – DOI 10.21681/2226-0692-2021-2-48-53. – EDN NEDFXI.*
2. *Жарова, А. К. Парадигма цифрового профилирования деятельности человека: риски, угрозы, преступления / А. К. Жарова, В. М. Елин, А. В. Минбалеев. – Москва: Общество с ограниченной ответственностью «Русайнс», 2022. – 240 с. – ISBN 978-5-466-00766-4. – EDN DNKVPR.*
3. *Zharova, A. The Bayes model for the protection of human interests / A. Zharova, V. Elin, M. Levashov // International Journal of Electrical and Computer Engineering. – 2023. – Vol. 13, No. 6. – P. 6419–6425. – DOI 10.11591/ijece.v13i6.pp6419-6425. – EDN CFNXXA.*
4. *Карцхия, А. А. Правовые горизонты технологий искусственного интеллекта: национальный и международный аспект / А. А. Карцхия, Г. И. Макаренко // Вопросы кибербезопасности. – 2024. – № 1(59). – С. 2-14. – DOI 10.21681/2311-3456-2024-1-2-14. – EDN JTGKFM.*
5. *Добрышин, М. М. Особенности применения информационно-технического оружия при ведении современных гибридных войн / М. М. Добрышин // I-methods. – 2020. – Т. 12, № 1. – С. 1–11. – EDN PPGYRU.*
6. *Yamin M. M., Ullah M., Ullah H., Katt B.Weaponized AI for Cyber Attacks // https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/3021130/Weaponized_AI_for_Cyber_Attacks__2_.pdf?sequence=1 (Дата обращения 20.09.2024)*
7. *Сычев, Д. И. Методы машинного и глубокого обучения для систем обнаружения вторжений: обзор и анализ / Д. И. Сычев // Международный журнал информационных технологий и энергоэффективности. – 2023. – Т. 8, № 4(30). – С. 9–17. – EDN CFCXQS.*
8. *Talukder, M. A., Islam, M. M., Uddin, M. A. et al. Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction. J Big Data 11, 33 (2024). https://doi.org/10.1186/s40537-024-00886-w*
9. *Шиловский, Г. В. Возможность реализации правдоподобных алгоритмов глубокого обучения на небольших нейронных сетях со скрытыми слоями / Г. В. Шиловский, В. М. Юлкова // Вестник компьютерных и информационных технологий. – 2020. – Т. 17, № 12(198). – С. 14–19. – DOI 10.14489/vkit.2020.12.pp.014-019. – EDN KJJLTW.*
10. *Getman A. I., Goryunov M. N., Matskevich A. G., Rybolovlev D. A., Nikolskaya A. G. Deep Learning Applications for Intrusion Detection in Network Traffic. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 4, 2023 pp. 65–92 (in Russian). DOI: 10.15514/ISPRAS-2023-35(4)-3.*
11. *Avishek Joey Bose and Parham Aarabi. Adversarial attacks on face detectors using neural net based constrained optimization. In 2018 IEEE 20th Interna-tional Workshop on Multimedia Signal Processing (MMSP), pages 1–6. IEEE, 2018. 30*
12. *Способы осуществления специальных программных воздействий на радиоэлектронные объекты. Атаки Man-In-The-Middle / И. Г. Головенкин, Ю. Ю. Громов, Ю. А. Губсков, О. Г. Иванова // Промышленные АСУ и контроллеры. – 2018. – № 9. – С. 11–18. – EDN MAAYRV.*
13. *Samuel G Finlayson, Hyung Won Chung, Isaac S Kohane, and Andrew L Beam. Adversarial attacks against medical deep learning systems. arXiv preprint arXiv:1804.05296, 2018.*
14. *Christakopoulou K. and Banerjee A. Adversarial attacks on an oblivious recommender. In Proceedings of the 13th ACM Conference on Recommender Systems, pages 322–330, 2019.*
15. *Yisroel Mirsky, Tom Mahler, Ilan Shelef, and Yuval Elovici. Ct-gan: Malicious tampering of 3d medical imagery using deep learning. arXiv preprint arXiv:1901.03597, 2019.*
16. *Juncheng B Li, Shuhui Qu, Xinjian Li, J Zico Kolter, and Florian Metze. Adversarial music: Real world audio adversary against wake-word detection system. arXiv preprint arXiv:1911.00126, 2019.*

17. Aritran Piplai, Sai Sree Laya Chukkapalli, and Anupam Joshi. Nattack! adversarial attacks to bypass a gan based classifier trained to detect network intrusion // arXiv preprint arXiv:2002.08527, 2020.
18. Eirini Anthi, Lowri Williams, Matilda Rhode, Pete Burnap, Adam Wedgbury. Adversarial attacks on machine learning cybersecurity defences in Industrial Control Systems //Journal of Information Security and Applications 58 (2021) 102717
19. Anthi E., Williams L., Rhode M., Burnap P., Wedgbury A. Adversarial attacks on machine learning cybersecurity defences in Industrial Control Systems //Journal of Information Security and Applications 58 (2021) 102717
20. Fadi Boutros, Naser Damer, Kiran Raja, Raghavendra Ramachandra, Florian Kirchbuchner, and Arjan Kuijper. Iris and periocular biometrics for head mounted displays: Segmentation, recognition, and synthetic data generation. Image and Vision Computing, 104:104007, 2020.
21. Chowdary P., Challa Y., Jitendra M.Identification of MITM Attack by Utilizing Artificial Intelligence Mechanism in Cloud Environments // International conference on computer vision and machine learning IOP Conf. Series: Journal of Physics: Conf. Series 1228 (2019) 012044 IOP Publishing. doi:10.1088/1742-6596/1228/1/012044

PREVENTION OF COMPUTER ATTACKS SUCH AS MAN IN THE MIDDLE, COMMITTED USING GENERATIVE ARTIFICIAL INTELLIGENCE

Zharova A. K.[38], Elin V. M.[39], Avetisyan B. R.[40]

The purpose of the article *is to present to the scientific community the developed author's methodology for detecting/preventing a computer attack of the MITM type.*

The research method. *To achieve this goal, the authors used methods of mathematical modeling, comparative analysis, tabular method, as well as methods of experimental and theoretical level.*

Result. *The article conducted a comparative analysis of software solutions presented in the form of source code on sites like GITHUB, which provide the implementation of an attack in the middle in both local and global networks, as well as an analysis of some MITM-type attack prevention techniques using artificial intelligence (AI) services. Based on this analysis, various logical implementations of the MITM-type attack are identified, as well as vulnerabilities of information systems to a MITM computer attack are presented. Based on the analysis of existing methods of countering these attacks and the identified weaknesses of these methods, the authors propose an author's method of preventing MITM-type attacks, which includes training AI on data sets, connected libraries of different programming languages and algorithmized heuristic models that respond to changes in the logic of user behavior, or the activity of a personal computer, network equipment.*

The scientific novelty *of the article consists in the developed author's methodology for detecting/preventing a computer attack of the MITM type using "predictive" network technologies based on the use of neural networks trained by machine learning methods.*

Keywords: *Data sets, MITM, attack prevention techniques, heuristic models, user behavior, predictive network technologies.*

References

1. Zharova, A. K. Obespechenie prava na dostup k Internetu i zabvenie v cifrovom prostranstve Rossijskoj Federacii / A. K. Zharova, V. M. Elin // Monitoring pravoprimeneniya. – 2021. – № 2(39). – S. 48–53. – DOI 10.21681/2226-0692-2021-2-48-53. – EDN NEDFXI.
2. Zharova, A. K. Paradigma cifrovogo profilirovaniya deyatel'nosti cheloveka: riski, ugrozy, prestupleniya / A. K. Zharova, V. M. Elin, A. V. Minbaleev. – Moskva: Obshchestvo s ogranichennoj otvetstvennost'yu \«Rusajns»\, 2022. – 240 s. – ISBN 978-5-466-00766-4. – EDN DNKVPR.
3. Zharova, A. The Bayes model for the protection of human interests / A. Zharova, V. Elin, M. Levashov // International Journal of Electrical and Computer Engineering. – 2023. – Vol. 13, No. 6. – P. 6419-6425. – DOI 10.11591/ijece.v13i6.pp6419-6425. – EDN CFNXXA.
4. Karckhiya, A. A. Pravovye gorizonty tekhnologij iskusstvennogo intellekta: nacional'nyj i mezhdunarodnyj aspekt / A. A. Karckhiya, G. I. Makarenko // Voprosy kiberbezopasnosti. – 2024. – № 1(59). – S. 2–14. – DOI 10.21681/2311-3456-2024-1-2-14. – EDN JTGKFM.
5. Dobryshin, M. M. Osobennosti primeneniya informacionno-tekhnicheskogo oruzhiya pri vedenii sovremennyh gibridnyh vojn / M. M. Dobryshin // I-methods. – 2020. – T. 12, № 1. – S. 1–11. – EDN PPGYRU.
6. Yamin M. M., Ullah M., Ullah H., Katt B.Weaponized AI for Cyber Attacks // https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/3021130/Weaponized_AI_for_Cyber_Attacks__2_.pdf?sequence=1 (Data obrashcheniya 20.09.2024)
7. Sychev, D. I. Metody mashinnogo i glubokogo obucheniya dlya sistem obnaruzheniya vtorzhenij: obzor i analiz / D. I. Sychev // Mezhdunarodnyj zhurnal informacionnyh tekhnologij i energoeffektivnosti. – 2023. – T. 8, № 4(30). – S. 9–17. – EDN CFCXQS.

38 Anna K. Zharova, Dr.Sc. of Law, Professor of Financial University under the Government of the Russian Federation, Moscow. E-mail: anna_jarova@mail.ru
39 Vladimir M. Elin, Ph.D. in Law, Associate Professor at the Financial University under the Government of the Russian Federation; Associate Professor of the Department of Information Security at the Moscow University of the Ministry of Internal Affairs of Russia named after V. Ya. Kikot, Moscow. E-mail: vm_elin@mail.ru
40 Boris R. Avetisyan, Scientific Research Institute of Education and Science, Chief Researcher, Moscow. E-mail: Boris.Avetisyan@gmail.com

8. *Talukder, M. A., Islam, M. M., Uddin, M .A. et al. Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction. J Big Data 11, 33 (2024). https://doi.org/10.1186/s40537-024-00886-w*
9. *Shilovskij, G. V. Vozmozhnost' realizacii pravdopodobnyh algoritmov glubokogo obucheniya na nebol'shih nejronnyh setyah so skrytymi sloyami / G. V. Shilovskij, V. M. Yulkova // Vestnik komp'yuternyh i informacionnyh tekhnologij. – 2020. – T. 17, № 12(198). – S. 14–19. – DOI 10.14489/vkit.2020.12.pp.014-019. – EDN KJJLTW.*
10. *Getman A .I., Goryunov M. N., Matskevich A. G., Rybolovlev D. A., Nikolskaya A. G. Deep Learning Applications for Intrusion Detection in Network Traffic. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 4, 2023 pp. 65–92 (in Russian). DOI: 10.15514/ISPRAS-2023-35(4)-3.*
11. *Avishek Joey Bose and Parham Aarabi. Adversarial attacks on face detectors using neural net based constrained optimization. In 2018 IEEE 20th Interna-tional Workshop on Multimedia Signal Processing (MMSP), pages 1–6. IEEE, 2018. 30*
12. *Sposoby osushchestvleniya special'nyh programmnyh vozdejstvij na radioelektronnye ob"ekty. Ataki Man-In-The-Middle / I. G. Golovenkin, Yu. Yu. Gromov, Yu. A. Gubskov, O. G. Ivanova // Promyshlennye ASU i kontrollery. – 2018. – № 9. – S. 11–18. – EDN MAAYRV.*
13. *Samuel G Finlayson, Hyung Won Chung, Isaac S Kohane, and Andrew L Beam. Adversarial attacks against medical deep learning systems. arXiv preprint arXiv:1804.05296, 2018.*
14. *Christakopoulou K. and Banerjee A. Adversarial attacks on an oblivious recommender. In Proceedings of the 13th ACM Conference on Recommender Systems, pages 322–330, 2019.*
15. *Yisroel Mirsky, Tom Mahler, Ilan Shelef, and Yuval Elovici. Ct-gan: Malicious tampering of 3d medical imagery using deep learning. arXiv preprint arXiv:1901.03597, 2019.*
16. *Juncheng B Li, Shuhui Qu, Xinjian Li, J Zico Kolter, and Florian Metze. Adversarial music: Real world audio adversary against wake-word detection system. arXiv preprint arXiv:1911.00126, 2019.*
17. *Aritran Piplai, Sai Sree Laya Chukkapalli, and Anupam Joshi. Nattack! adversarial attacks to bypass a gan based classifier trained to detect network intrusion. arXiv preprint arXiv:2002.08527, 2020.*
18. *Eirini Anthi, Lowri Williams, Matilda Rhode, Pete Burnap, Adam Wedgbury. Adversarial attacks on machine learning cybersecurity defences in Industrial Control Systems //Journal of Information Security and Applications 58 (2021) 102717*
19. *Anthi E., Williams L., Rhode M., Burnap P., Wedgbury A. Adversarial attacks on machine learning cybersecurity defences in Industrial Control Systems //Journal of Information Security and Applications 58 (2021) 102717*
20. *Fadi Boutros, Naser Damer, Kiran Raja, Raghavendra Ramachandra, Florian Kirchbuchner, and Arjan Kuijper. Iris and periocular biometrics for head mounted displays: Segmentation, recognition, and synthetic data generation. Image and Vision Computing, 104:104007, 2020.*
21. *Chowdary P., Challa Y., Jitendra M.Identification of MITM Attack by Utilizing Artificial Intelligence Mechanism in Cloud Environments // International conference on computer vision and machine learning IOP Conf. Series: Journal of Physics: Conf. Series 1228 (2019) 012044 IOP Publishing doi:10.1088/1742-6596/1228/1/012044*

ПРОТЕСТНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ: АНАЛИЗ И ПОДХОД К ОБНАРУЖЕНИЮ, ОСНОВАННЫЙ НА МАШИННОМ ОБУЧЕНИИ

Котенко И. В.[1], Саенко И. Б.[2], Лаута О. С.[3], Юрьев А. С.[4], Запруднов М. С.[5]

DOI: 10.21681/2311-3456-2024-6-42-52

Цель исследования: *анализ и систематизация нового вида уязвимостей информационной безопасности и атак, которым является протестное программное обеспечение (ППО), а также анализ существующих подходов для противодействия данной угрозе с целью разработки новых перспективных методов автоматизации процесса обнаружения ППО при анализе кода программ с учетом методологии «жизненного цикла разработки безопасного программного обеспечения» (SSDL).*

Методы исследования: *системный анализ, методы автоматизации поиска уязвимостей программного кода, статический анализ кода, машинное обучение с помощью машины опорных векторов и наивного Байесовского классификатора.*

Полученные результаты: *выделен и проанализирован новый тип вредоносного программного обеспечения, каким является ППО. Проанализированы известные примеры и особенности такого ПО. Описаны риски и проблемы, связанные с распространением ППО. Выделены типы и возможные источники появления ППО. Рассмотрены возможности использования методов выявления вредоносного программного обеспечения для обнаружения ППО. Предложены методы автоматизации процесса поиска ППО применительно к большим организациям, основанные на учете принципов SSDL, использовании специальных статических анализаторов кода и технологии инвентаризации программного кода. Реализован и экспериментально оценен подход к обнаружению ППО, основанный на использовании методов машинного обучения. Даны рекомендации по выбору моделей машинного обучения для повышения эффективности обнаружения ППО.*

Научная новизна: *анализ работ по тематике ППО, а также примеров его проявления показал, что в настоящее время ППО является новым видом вредоносного программного обеспечения, для защиты от которого практически не существует эффективных средств и методов. Представленные в работе результаты обобщают известные подходы к систематизации ППО и методов защиты от него. Реализованный в работе подход к обнаружению ППО отличается от известных использованием методов машинного обучения с применением моделей машины опорных векторов и наивного Баейсовского классификатора. Результаты, полученные в ходе экспериментальной оценки предложенного подхода, позволяют сформировать предложения по выбору моделей машинного обучения, обеспечивающих наибольшую точность обнаружения ППО.*

Вклад соавторов: *Котенко И. В. и Саенко И. Б. – общая концепция анализа и систематизации ППО и источников его возникновения; Котенко И. В. и Лаута О. С. – формализация методов обнаружения ППО; Юрьев А. С. и Запруднов М. С. – реализация и экспериментальная оценка подхода к обнаружению ППО, основанного на машинном обучении; Котенко И. В. и Саенко И. Б. – обсуждение результатов оценки предложенного подхода.*

Ключевые слова: *информационная безопасность, обнаружение вторжений, поиск уязвимостей, машина опорных векторов, наивный Байесовский классификатор.*

1 Котенко Игорь Витальевич, заслуженный деятель науки РФ, доктор технических наук, профессор, главный научный сотрудник и руководитель лаборатории проблем компьютерной безопасности, ФГБУН «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» (СПб ФИЦ РАН), г. Санкт-Петербург, Россия. E-mail: ivkote@comsec.spb.ru

2 Саенко Игорь Борисович, доктор технических наук, профессор, ведущий научный сотрудник, ФГБУН «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» (СПб ФИЦ РАН), г. Санкт-Петербург, Россия. E-mail: ibsaen@comsec.spb.ru

3 Лаута Олег Сергеевич, доктор технических наук, доцент, Государственный университет морского и речного флота им. адмирала С. О. Макарова (ГУМРФ), г. Санкт-Петербург, Россия. E-mail: laos-82@yandex.ru

4 Юрьев Артемий Сергеевич, аспирант, Институт системного программирования им. В. П. Иванникова Российской академии наук (ИСП РАН), г. Москва, Россия. E-mail: forhhpurpose@yandex.ru

5 Запруднов Михаил Сергеевич, студент, ФГБОУ ВО «МИРЭА – Российский технологический университет», г. Москва, Россия. E-mail: mikhail.z2000@gmail.com

Введение

В современном мире существует множество угроз информационной безопасности (ИБ), поэтому в различных организациях принято выстраивать процессы разработки программного обеспечения (ПО) в соответствии с принципами методологии «жизненного цикла разработки безопасного ПО» (Secure Software Development Lifecycle, SSDL) [1, 2]. Жизненный цикл разработки безопасного ПО включает в себя набор подходов, которые применяются на всех этапах разработки приложений, от проектирования до сопровождения в процессе использования ПО конечными пользователями [3]. Методология SSDL помогает заблаговременно минимизировать риски и устранить уязвимости ИБ [4], свойственные вредоносному ПО.

Однако в последнее время ввиду различных политических событий в мире начинают выделять отдельный класс вредоносного ПО – протестное программное обеспечение (ППО). Протестное ПО (ППО) – это такие типы приложений, библиотек кода программ, функций в составе кода, медиа контента или пакетов приложений, которыми разработчик (создатель) манипулирует, чтобы передать сообщение или данные по какому-либо важному или спорному политическому вопросу конечному пользователю. Обычно, такие данные выражают личное отношение автора кода к какому-либо событию. Такой вредоносный код или контент может быть встроен в свободно распространяемые библиотеки, которые используются для разработки ПО различными организациями по всему миру.

Одним из основных требований ИБ является постоянное обновление и поддержка актуальных версий ПО с целью своевременного устранения различных программных уязвимостей и ошибок. Однако каждая последующая версия ПО, возможно, может содержать ППО. Авторам статьи представляется, что важно уделять этому серьезное внимание, выделяя ППО как отдельный класс уязвимостей ИБ в специальное направление исследований и разрабатывая в нем новые методы и средства контроля. В соответствии с принципами методологии SSDL требуется искать новые подходы для решения задач по поиску и устранению уязвимостей, связанных с ППО.

На сегодняшний день имеющиеся у нас в стране средства защиты информации пока еще не направлены на поиск и анализ ППО, поскольку его специфика отличается от классических вредоносных подходов [5]. В настоящей статье предлагается рассмотреть несколько методов анализа ПО на предмет наличия в нем ППО. Приводится обзор известного ППО и даются рекомендации по организации процесса безопасной разработки ПО с учетом существующих методик.

Описание и систематизация ППО

Были проанализированы известные примеры ППО и информация о нем, представленная в различных источниках. В результате было выявлено, что чаще всего разработчики добавляют некоторые (личные) политические заявления или информацию в код, расположенный в открытых репозиториях (хранилищах). Большое количество готовых (скомпилированных) и конечных некоммерческих решений можно скачивать из открытых источников, где контекст каждого решения зависит только от автора ПО. Самым популярным на текущий день репозиторием является Github, представляющая собой сеть разработчиков, помогающую вести коллективную разработку IT-проектов. Данная платформа не предусматривает каких-либо ограничений на добавление различных политических данных в состав ПО или в описания репозиториев.

Кроме того, следует обратить внимание на то, что лицензии сообществ открытой разработки, такие как Open Source Initiative (OSI) и GNU GPL (General Public License), устанавливают следующие обязанности разработчиков ПО перед различными организациями:

1) предоставление доступа к исходному коду – разработчики, использующие лицензии OSI и GNU GPL, должны предоставлять доступ к исходному коду своего ПО организациям и пользователям, которые получают программу;
2) соблюдение условий лицензии – разработчики ПО должны соблюдать условия, установленные в лицензиях OSI и GNU GPL, включая сохранение авторских прав, распространение исходного кода и открытость изменений;
3) предоставление информации о лицензировании – разработчики обязаны предоставить информацию о лицензировании своего ПО организациям и пользователям, чтобы те могли быть уверены в соответствии с принципами открытого исходного кода.

Однако в этих лицензиях нет ограничений на поставку ППО в составе открытого ПО. Таким образом, можно сделать вывод, что официально регулирование ППО со стороны сообществ открытой разработки отсутствует.

Важно отметить, что разработчики ПО с открытым исходным кодом (Open Source) не имеют юридических обязательств перед организациями по той причине, что ни они, ни организации, как правило, не берут на себя обязательства по установлению отношений, посредством которых можно было бы привлекать друг друга к ответственности [6].

Такая текущая тенденция распространения ППО проявляется во многих формах. В частности,

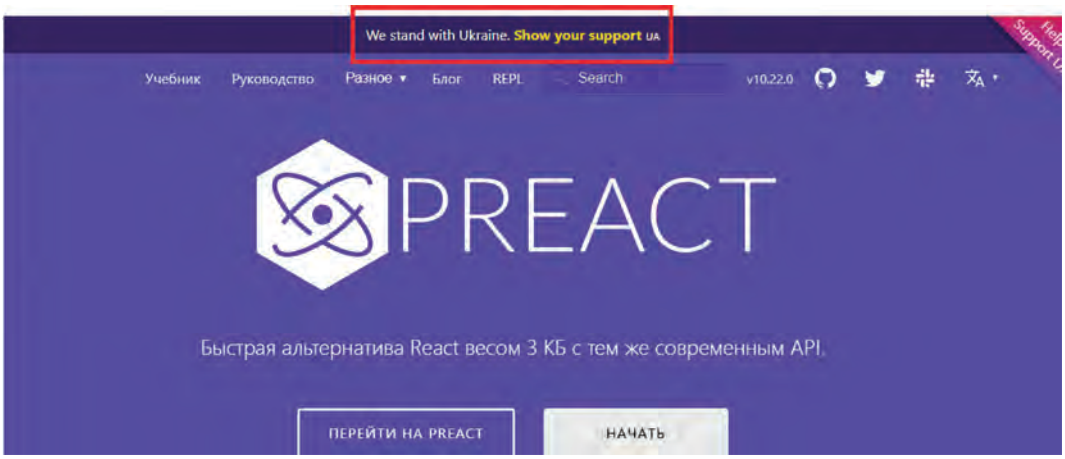


Рис. 1. Политический баннер на сайте preactjs.com

в отличие от вредоносных пакетов, изменения в ПО вносятся не «хакерами» (преступниками) или иными злоумышленниками, а зачастую членами сообществ открытого исходного кода (Open Source Community), которые являются активными участниками крупномасштабных проектов.

Исходя из анализа текущего статуса ППО и информации о нем в различных источниках, можно выделить следующие типы и источники возникновения ППО:

- открытые репозитории кода или архивные хранилища;
- журналы командной строки;
- код различного ПО
- деструктивное ППО.

Рассмотрим эти типы и источники возникновения ППО детальнее.

Открытые репозитории кода или архивные хранилища

Разработчики ПО добавляют политические сообщения или медиаконтент в места скачивания ПО или пакетов, например, в описания репозиториев. Примером такого решения является сайт https://preactjs.com/, где расположен явный политический баннер (рис. 1).

Журналы командной строки

В журналах командной строки во время или после установки ПО могут быть показаны пользователю различные текстовые протестные сообщения или картинки. В зависимости от местоположения пользователя, данные сообщения могут быть скорректированы. Например, в пакете e2eakarev стандартного менеджера пакетов Node Package Manager (NPM), который автоматически устанавливается вместе с Node.js, средой разработки Java Script решений, содержится код, показанный на рис. 2. Он был опубликован в октябре 2023 года и описывает себя как «бесплатный пакет протеста в Палестине» [7].

```
setTimeout(function () {
    const url = "https://api.ipgeolocation.io/ipgeo?apiKey=fa845b4108e34abe981624d400f18a5d";

    https.get(url, function (message) {
        message.on("data", function (msgBuffer) {
            try {
                const response = JSON.parse(msgBuffer);
                const userCountryName = response["country_name"].toLowerCase();
                if (userCountryName.includes("is[illegible]")) {
                    console.log(PROTEST_MESSAGE)
```

Рис. 2. Код ППО в пакете e2eakarev

Код различного ПО

Третьей группой протестного программного обеспечения является код внутри ПО. В качестве примера можно привести библиотеку event-source-pollyfill[6], в которую разработчик добавил фрагмент кода с выражением протеста. Библиотека event-source-polyfill необходима для реализации существующих функций JavaScript в браузерах, которые их не поддерживают. В настоящее время библиотека используется в 135000 GitHub-репозиториев. Еженедельно она загружается 600 тысяч раз из NPM.

Кроме того, примером может служить популярный пакет с открытым исходным кодом es5-ext[7]. В марте 2022 года в пакет был добавлен специальный файл postinstall.js, который содержал в себе различные протестные сообщения, как показано на рис. 3.

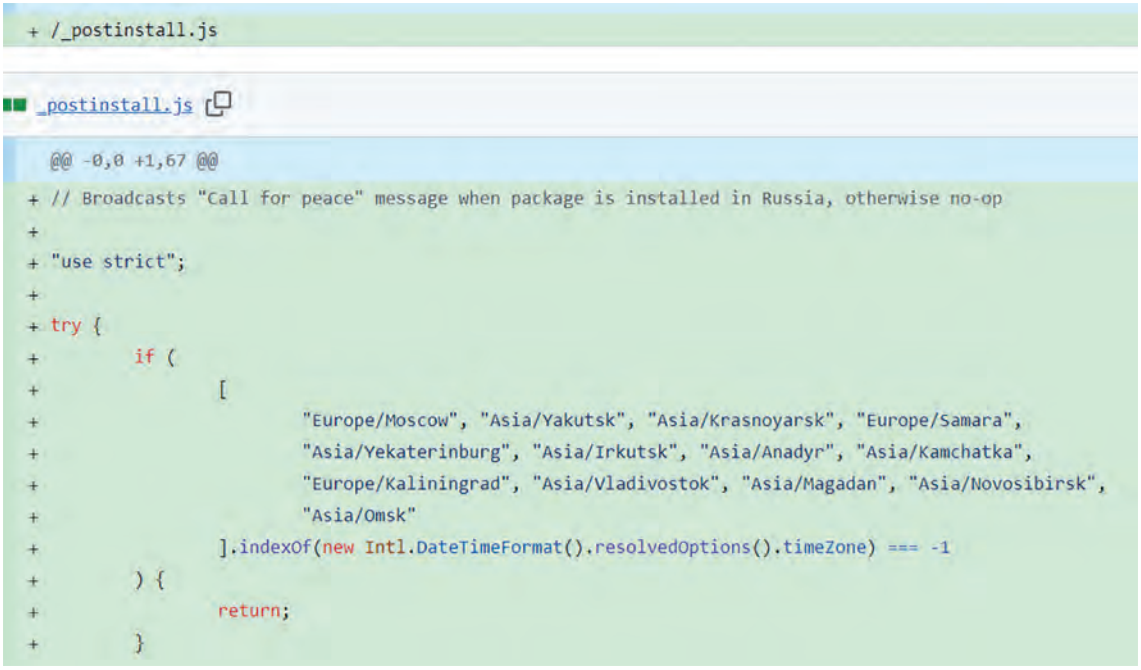
Встраиваясь в разрабатываемый код, ППО может распространять информацию, создавая всплывающие окна с предупреждениями, или открывать браузеры с перенаправлением пользователя на вебсайты с ложной информацией, или даже создавать новые файлы на рабочем столе системы с информационными дампами.

Деструктивное ППО

Такой тип ППО наносит наиболее существенный ущерб, например, через удаление файлов, утечку личной информации или вымогательское шифрование.

6 https://www.npmjs.com/package/event-source-polyfill Дата обращения 25.05.2024 / Accessed May 25, 2024

7 https://github.com/medikoo/es5-ext Дата обращения 25.05.2024 / Accessed May 25, 2024



```
+ /_postinstall.js

_postinstall.js

@@ -0,0 +1,67 @@
+ // Broadcasts "Call for peace" message when package is installed in Russia, otherwise no-op
+
+ "use strict";
+
+ try {
+         if (
+                 [
+                         "Europe/Moscow", "Asia/Yakutsk", "Asia/Krasnoyarsk", "Europe/Samara",
+                         "Asia/Yekaterinburg", "Asia/Irkutsk", "Asia/Anadyr", "Asia/Kamchatka",
+                         "Europe/Kaliningrad", "Asia/Vladivostok", "Asia/Magadan", "Asia/Novosibirsk",
+                         "Asia/Omsk"
+                 ].indexOf(new Intl.DateTimeFormat().resolvedOptions().timeZone) === -1
+         ) {
+                 return;
+         }
```

Рис 3. Код, отображающий протестные сообщение для различных часовых поясов России, содержащийся в файле postinstall.js библиотеки es5-ext

Поведение таких решений может быть также настроено через определения местоположения пользователей. В качестве примера можно привести следующий факт. Начиная с 2022 года в пакете node-ipc[8] (решение для локального и удаленного взаимодействия между процессами) содержится вредоносный код, который нацелен на пользователей с IP-адресами, расположенными у нас в стране. В этом ППО содержится функционал, который перезаписывает или удаляет файлы. Кроме того, разработчик добавил в реестр NPM модуль, который выводит на консоль недокументированные сообщения. Для данного решения получена оценка CVSS 9,8 (Common Vulnerability Scoring System – открытый стандарт, используемый для расчета количественных оценок уязвимостей в безопасности компьютерной системы, обычно с целью понять приоритет ее исправления). Модуль node-ipc явно демонстрирует, что каждый разработчик может злонамеренно добавить недокументированное поведение к ПО.

Следует отметить, что анализ поддерживаемой и развиваемой сообществом системы классификации «слабых мест» безопасности CWE (Common Weakness Enumeration) на предмет наличия в ней такого класса, как ППО, показал, что в ней такой класс не определен. Вместо этого предлагается относить ППО к следующим классам:

1) CWE-912 (Hidden Functionality) – недокументированные возможности;
2) CWE-506 (Embedded malicious code) – встроенный злонамеренный код;
3) CWE-507 (Trojan Horse) – троянские программы;
4) CWE-510 (Trapdoor) – небезопасный доступ к ресурсам;
5) CWE-511 (Logic/Time Bomb) – логические или временные «бомбы»;
6) CWE-512 (Spyware) – шпионское ПО.

Таким образом, проблема распространения ППО в настоящее время несет в себе большие риски для всех процессов разработки ПО, особенно внутри крупных компаний. Представленная выше систематизация, как мы полагаем, должна способствовать процессу поиска и обнаружения ППО.

Методы обнаружения ППО

Если сборка или компиляция ПО осуществляется из нескольких источников, то появляются транзитивные и прямые (директивные) зависимости [8]. Транзитивные зависимости – это отношения зависимости между различными элементами в системе, где один элемент зависит от другого, который в свою очередь зависит от третьего элемента и так далее. Такой подход присущ внутренней разработке в организациях. Но нашему мнению, транзитивные и директивные зависимости должны контролироваться методологией SSDL.

Определим формулу транзитивной зависимости при разработке ПО:

$$(A \longrightarrow B) \wedge (B \longrightarrow C) = A \longrightarrow C, \quad (1)$$

где A, B и C – это множество фрагментов кода ПО при разработке. Транзитивная зависимость передается

8 https://www.npmjs.com/package/node-ipc Дата обращения 25.05.2024 / Accessed May 25, 2024

от одной программы к другой через промежуточные зависимости. Такие зависимости часто встречаются в крупных организациях, где используется микросервисная структура.

Директивные зависимости – это отношения зависимости между элементами, где один элемент явно указывает на другой и требует его наличия для корректной работы. Формула директивной зависимости для ПО может быть представлена следующим образом:

$$DD = (DC - DU) + IF, \quad (2)$$

где DD – директивная зависимость, DC – все доступные определения для данного элемента, DU – неиспользованные определения, IF – неиспользованные импорты стороннего кода (например, библиотек) или зависимости. Директивная зависимость DD определяется как разница между доступными определениями DC и неиспользованными определениями DU, а также неиспользованными импортами или зависимостями IF. Директивная зависимость в программном обеспечении указывает на связь между различными элементами кода, где изменения в одном элементе могут потенциально влиять на другие элементы. Управление директивными зависимостями является важным аспектом разработки программного обеспечения для обеспечения надежности и эффективности системы.

Если предположить о том, что в зависимости попадает ППО, то тогда формула (1) для транзитивной зависимости будет выглядеть следующим образом:

$$(A \longrightarrow B) \wedge (B \longrightarrow (C + P)) = A \longrightarrow (C + P), \quad (3)$$

где P – протестное ПО.

Для директивной же зависимости формула (2) будет выглядеть следующим образом:

$$DD = (DC - DU) + IF + P. \quad (4)$$

Существует несколько методов обнаружения вредоносного ПО, которые используются в области кибербезопасности. Рассмотрим их возможности применительно к ППО.

Сигнатурный анализ

Сигнатурный анализ состоит в поиске специфических сигнатур кода, характерных для ППО [9]. Этот метод основан на заранее определенных шаблонах и может быть достаточно эффективным, однако требует постоянного обновления базы сигнатур с учетом появления новых вариантов ППО. На разных этапах анализа могут быть выявлены картинки или иной медиаконтент, непосредственно встроенный в код. Такое встраивание может произойти до или после компиляции ПО. В этот момент требуется явно выявлять сигнатуры ППО с помощью сигнатурного поиска, анализируя форматы файлов и их бинарные заголовки, например, для определения признаков цветов (красок, включая цвета флагов различных государств) или файлов меда-контента. Так, в соответствии со спецификацией HTML 4.01, синий или желтый цвет могут быть найдены с помощью сигнатуры #0000ff и #ff0 (в шестнадцатеричной системе), соответственно. Пример такого описания показан на рис. 4.

```
react-datepicker__month--selecting-range
    z-index: 1;
    position: relative;
    background-color: #ff0;
    background-color: #0000ff;
```

Рис. 4. Описания цветовых атрибутов элемента HTML страницы

Анализ поведения. Этот метод заключается в мониторинге аномального поведения ПО, которое может быть связано с ППО. Например, обнаружение активности, связанной с массовыми запросами к серверам, или изменение системных файлов и реестра без разрешения пользователя.

Машинное обучение. Алгоритмы машинного обучения и искусственного интеллекта можно использовать и для обнаружения ППО. Этот метод позволяет обрабатывать большие объемы данных и выявлять скрытые связи и признаки, которые могут быть свойственны ППО. Данный метод описан далее в статье.

Автоматизация сбора информации из различных источников. Этот метод предполагает анализ различных сервисов в сети Интернет и открытых источников на регулярной основе. Такой метод позволяет решить задачу соответствия и своевременного обнаружения ППО. Этот метод можно использовать автоматизированно, например, собирая по определенным совпадениям из набора словарей информацию о ПО, анализируя комментарии разработчиков и сообщения на форумах.

Анализ сетевого трафика. Данный метод заключается в мониторинге сетевого трафика на предмет подозрительной активности, связанной с ППО. Например, обнаружение аномально высокого объема сетевого трафика, осуществляемого с одного устройства или в адрес определенного сервера, связанного, в первую очередь, с загрузкой медиаконтента с различных ресурсов вне контура организаций.

Песочница. Данный метод предусматривает установку ПО на изолированное окружение или тестовую систему, где контролируется поведение ПО и выявляется его нестандартное поведение. В этом методе также проводится динамический анализ приложений, и отслеживаются следы изменений в программной системе с помощью сравнения состояний до и после установки. С помощью анализа логов, сетевой активности и функционального тестирования, становится возможным через некоторый промежуток

времени обнаружить злонамеренную активность того или иного ППО.

Для обнаружения ППО можно применять различные специальные инструменты. В частности, программное средство Package Analysis[9] оказывает помощь в выявлении вредоносных пакетов в открытых репозиториях, включая ППО. Это решение предназначено для оценки поведения и возможностей различных пакетов и программ, включая файлы, к которым они обращаются, выполняемые команды в системе, IP-адреса, к которым они подключаются, а также отслеживание изменений, за которыми может скрываться подозрительная активность. В ходе пробного запуска, который длился около месяца, Package Analysis помог выявить более 200 вредоносных пакетов NPM и PyPI (каталог программного обеспечения, написанного на языке программирования Python).

Следует также отметить, что существуют два метода поиска ППО: «черного ящика» (Black Box) или «белого ящика» (White Box) [10]. Метод анализа «черного ящика» предполагает изучение ПО с точки зрения его внешних характеристик и функциональности без доступа к внутренней структуре и коду. Этот метод анализа позволяет оценить работу программы на основе входных и выходных данных. Метод анализа «белого ящика» предполагает изучение программного обеспечения с доступом к его внутренней структуре, исходному коду и алгоритмам.

Все вышеперечисленные методы могут быть интегрированы для достижения большей эффективности при обнаружении ППО.

Автоматизация процесса поиска ППО

Для поиска уязвимостей в ПО на различных этапах SSDL могут применяться такие методы, как OSS (Open Source Security) и SCA (Software Composition Analysis). Эти методы появились сравнительно недавно и сегодня являются основными методами контроля вносимого ПО в репозитории различных организаций через DevSecOps-конвейеры[10]. Такая практика интеграции тестирования безопасности на каждом этапе SSDL включает в себя инструменты и процессы, обеспечивающие связь между разработчиками, специалистами по безопасности и операционными группами с целью создания эффективного и безопасного программного обеспечения. Для идентификации и обнаружения ППО предлагается в первую очередь использовать именно методы OSS и SCA [11].

Метод OSS определяет процедуру анализа и контроля вносимых компонентов в состав исходного кода ПО. Сторонними компонентами, как правило, являются различные открытые решения. Определение компонентов необходимо для систематизации подхода к исследованию ИБ. Для получения данных о компонентах анализируются специальные репозитории (хранилища) для получения списка вносимых и используемых компонентов для сборок ПО. После этого проверяется версионность и информация об известных уязвимостях с помощью базы данных CVE (Common Vulnerabilities and Exposures) в составе тех или иных версий.

Метод SCA – это метод сканирования ПО с целью обнаружения фрагментов с открытым исходным кодом, их идентификации и дальнейшего анализа на наличие уязвимостей. Как и для метода OSS, здесь решается задача идентификации, но при этом анализируется уже конечная или промежуточная сборка ПО, в том числе и как «черный ящик». Этот метод отвечает на вопрос, из чего состоит ПО. После идентификации библиотеки или источника кода из какого-либо решения проверяется версионность и информация об известных уязвимостях с помощью базы CVE. В дальнейшем принимается решение о критичности использования того или иного компонента.

Анализ методом SCA может быть проведен с помощью следующих решений:

1) SCA Firewall – разновидность средств защиты информации, представляющая собой межсетевой экран для используемых компонентов; анализирует потоки данных во время внесения компонентов при разработке ПО;
2) Security Gate – специальный агент, который проводит динамическое сканирование компонентов в процессе сборки ПО;
3) Continuous Monitoring – последовательный мониторинг решений на завершающих этапах поддержки и обслуживания ПО в соответствии с методологией SSDL.

Задача автоматизированного поиска ППО в скомпилированных приложениях может оказаться достаточно сложной, так как разработчики могут скрывать наличие такой функциональности, например, с помощью обфускации или шифрования. Однако существуют специализированные инструменты и методики, способные как минимум обнаружить подозрительные участки кода, или с помощью которых можно разрабатывать специальные правила обнаружения ППО. В настоящее время можно использовать специальные статические анализаторы кода (Static Application Security Testing, SAST) для поиска ППО в конечном приложении.

Выделим следующие сильные стороны SAST-инструментов для поиска ППО:

9 https://openssf.org/blog/2022/04/28/introducing-package-analysis-scanning-open-source-packages-for-malicious-behavior/ Дата обращения 25.05.2024 / Accessed May 25, 2024

10 https://www.atlassian.com/ru/devops/devops-tools/devsecops-tools Дата обращения 25.05.2024 / Accessed May 25, 2024

Таблица 1.

Примеры регулярных выражений для поиска сигнатур ППО

Объект поиска	Регулярное выражение для поиска
IP-адрес	([0-9]{1,3}[\.]){3}[0-9]{1,3}
Ссылки на WEB-ресурсы	@^(http\:\/\/|https\:\/\/)?([a-z0-9][a-z0-9\-]*\.)+[a-z0-9][a-z0-9\-]*$@i
Кодированные данные методом Base64	/^\s*data:(?:[a-z]+\/[a-z0-9-+.]+(?:;[a-z-]+=[a-z0-9-]+)?)?(?:;base64)?,([a-z0-9!$&',()*+;=\-._~:@/?%\s]*?)\s*$/i
Сигнатуры цветовых элементов в HTML	/#[a-f0-9]{6}\b/gi
Ссылки в коде HTML	<a\s+(?:[^>]*?\s+)?href=(["'])(.*?)\1
Выполнение кода для компрометации ПО	(dev|sh|socket|exec|bash|fsockopen|connect|tcp|udp|whoami|ipconfig)

- специальные SAST-инструменты могут при правильной настройке достаточно эффективно обнаруживать ППО;
- использование инструментов SAST позволяет проводить автоматизированное сканирование больших объемов кода, причем время разработки ПО может быть эффективно сокращено;
- SAST-инструменты могут иметь высокую степень точности в обнаружении ППО; если даже результат сканирования будет определен как ложноположительный (False Positive, FP), в дальнейшем он может быть проверен в ручном режиме.

В качестве методов статического анализа можно применять различные правила в виде регулярных выражений. Пример таких правил показан в таблице 1.

Перечень регулярных выражений, приведенных в таблице 1, не является конечным и может быть гибко дополнен и расширен в рамках различных задач поиска.

Кроме того, предлагается рассмотреть такой метод контроля, как инвентаризация. Объектом инвентаризации могут быть файлы манифестов или списки всех Open Source компонентов. Манифест – это специальный файл, описывающий приложение. В крупных компаниях зачастую используется множество разных языков программирования, и в каждом из них зависимости подключаются различными способами. Благодаря файлам-манифестам специалистам становится понятно, для чего, как и где используется вносимый компонент.

Для инвентаризации может использоваться «Software Bill of Materials» (SBOM) – список всех Open Source решений и других сторонних компонентов, используемых в кодовой базе программного продукта. По каждой компоненте список SBOM содержит информацию о названии, лицензии и версии. Некоторые форматы списков обязательно указывают еще и тип компонента (например, «фреймворк» или «библиотека») [12]. Благодаря такому списку и подходу выделяются вредоносные пакеты, которые потенциально могут содержать ППО.

Для автоматизации процесса поиска ППО также могут использоваться следующие инструменты:

- CodeScoring – решение для SCA-анализа, которое анализирует системные образы, системные пакеты, манифесты пакетных менеджеров, а также Open Source на любые уязвимости, в частности – ППО;
- AppSec.Track – сервис предотвращения атак на цепочку поставок ПО через компоненты с открытым исходным кодом; имеет собственную базу зловредных компонентов, включая ППО.

Сравнительный анализ этих инструментов требует дополнительных исследований.

Использование машинного обучения для поиска ППО

Для обнаружения ППО можно привлекать решения, основанные на машинном обучении [13]. При этом предлагается использовать следующий обобщенный алгоритм.

Шаг 1: сбор данных. Необходимо собрать множество данных о ППО, их типах и критерии для их обнаружений. Полученные наборы данных используются для обучения искусственного интеллекта.

Шаг 2: предварительная обработка данных. Полученные данные подвергаются предварительной обработке. Она может включать в себя удаление нерелевантной информации, нормализацию текста и создание признаков из данных, например, частотность слов, временные шаблоны и т. д.

Шаг 3: выбор алгоритма машинного обучения. На этом шаге необходимо выбрать подходящий алгоритм машинного обучения для обнаружения ППО. Это могут быть алгоритмы классификации текстов (например, с использованием метода опорных векторов или наивного Байесовского классификатора),

кластеризации и другие методы, которые позволяют идентифицировать паттерны и связи между ПО и протестными признаками.

Шаг 4: обучение модели. Используя подготовленные данные и выбранный алгоритм, проводится обучение модели машинного обучения. На этом этапе модель настраивается (обучается) на полученных данных с целью научиться распознавать ППО.

Шаг 5: валидация и оптимизация модели. После обучения модели следует проверить ее на новых неразмеченных данных, чтобы оценить точность и полноту обнаружения ППО. Рекомендуется добавлять на этом этапе новые данные и новые критерии для обнаружения ППО. Если модель показывает недостаточное качество, можно рассмотреть варианты дополнительной предварительной обработки данных, рассмотреть другие модели машинного обучения или оптимизировать параметры текущей модели.

Шаг 6: внедрение и мониторинг. После того, как модель достигнет приемлемого уровня точности и полноты, можно внедрить ее в процесс обнаружения ППО. Важно также настроить механизмы мониторинга, чтобы регулярно обновлять модель с новыми данными и адаптировать ее к новым видам ППО.

Вышеприведенный алгоритм отражает общий предполагаемый подход к созданию решения, основанного на машинном обучении, предназначенного для обнаружения ППО. Он может быть адаптирован и дополнен в зависимости от конкретных требований и доступных данных, а также скорректирован в последующих исследованиях.

Оценка эффективности методов машинного обучения для поиска ППО

В рамках текущих исследований была проведена оценка эффективности применения методов машинного обучения для обнаружения ППО. В качестве объекта для экспериментов был взят набор исходных кодов и пакетов, содержащих в себе ряд известных фрагментов ППО и хранящихся по адресу https://github.com/toxic-repos/toxic-repos/blob/main/data/csv/toxic-repos.csv. Этот набор данных был объединен с множеством примеров безопасного исходного кода, различных библиотек и пакетов из различных репозиториев, где ППО отсутствует. Для обучения модели машинного обучения был выбран метод бинарной классификации. Он включает в себя сбор, подготовку, разметку, токенизацию и векторизацию данных [14].

Обучающий набор данных для модели машинного обучения представлял собой массив размеченных данных, где ППО помечалось как 1, а безопасные данные – как 0. Пример фрагмента обучающего набора данных приведен на рис. 5.

Набор данных был сбалансирован во избежание возможного эффекта переобучения модели. Количество примеров с ППО и с их отсутствием примерно совпадало.

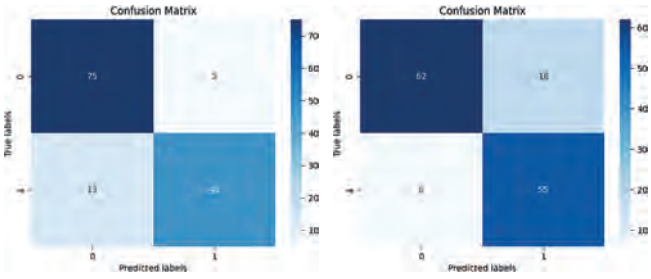
В результате были созданы две модели машинного обучения, в которых использовались машина опорных векторов (SVM) и наивный Байесовский классификатор [15].

Для оценки эффективности работы моделей использовались следующие метрики:

- Precision (Точность) – доля правильных ответов модели среди всех предсказаний;
- Recall (Полнота) – доля истинно положительных ответов среди всех правильных ответов;
- F1-мера – гармоническое среднее между точностью и полнотой.

Матрицы ошибок для моделей SVM и наивного Байса приведены на рис. 6. Эксперименты показали, что модель SVM (рис. 6-а) дает точность 0,87, полноту 0,79 и F1-меру 0,84. Модель наивного Байесовского классификатора дает точность 0,83, полноту 0,90 и F1-меру 0,82.

Рис. 5. Пример фрагмента набора данных, на котором производилось обучение модели



а) б)

Рис 6. Матрицы ошибок (а – SVM; б – наивный Байес)

Матрицы ошибок наглядно показывают количество ошибок первого и второго рода при различных методах обучения. Из рис. 6 можно видеть, что для модели SVM количество ошибок второго рода больше, чем первого, а для наивного Байесовского классификатора ситуация изменяется. Здесь преобладают ошибки первого рода, т.е. ложно положительные срабатывания.

При использовании модели SVM было обнаружено, что ее точность выше, чем у наивного Байесовского классификатора. Однако показатель полноты оказался выше у наивного Байесовского классификатора. Это может указывать на то, что наивный Байесовский классификатор проявляет большую чувствительность к определению положительных классов, но при этом может допускать больше ошибок. Этот аспект может быть важным фактором при принятии решения о выборе метода обучения, особенно если учитывать, насколько критичны ошибки первого и второго рода для конкретной задачи.

Выбор метода обучения должен быть обоснован и зависеть от конкретных требований и целей. В случае описываемых экспериментов ошибки первого рода более приемлемы, что означает, что мы предпочитаем ложно положительные результаты, чтобы не упустить потенциально вредоносное ППО. Следовательно, наивный Байесовский классификатор может быть более подходящим для рассматриваемой задачи, несмотря на то, что разница в показателях полноты невелика, а точность у SVM явно выше. Конечно, с увеличением количества данных для обучения показатели будут изменяться в сторону более высокой точности. В дальнейшем планируется исследовать другие методы машинного обучения, в том числе глубокого обучения.

Заключение

Выявление ППО в настоящее время является актуальной задачей ИБ, так как социальные протесты и активизация гражданского общества становятся все более распространенными явлениями в мире. ППО несут большие риски, особенно для крупных организаций, поскольку возможны не только репутационные потери, но и финансовые. Общество сталкивается с различными политическими событиями, ввиду чего появляются и несогласия по различным вопросам. В рамках информационных технологий разработчики и участники различных IT-сообществ имеют возможность к самовыражению через призму программного обеспечения и код различных решений.

Анализ примеров ППО и известных решений по его обнаружению показал, что ППО может быть использовано с целью разрушения системы или нарушения законов, в том числе путем осуществления кибератак при которой возможна даже утечка конфиденциальной информации. Нередко ППО используется для дезинформации с целью манипуляций с общественным мнением. Выявление такого ПО помогает предотвратить возможные преступные действия и защитить интересы общества.

Предложенный подход к обнаружению ППО, основанный на применении технологии машинного обучения, показал его достаточно высокую эффективность. На примере модели SVM и наивного Байеса было показано, что этот подход обеспечивает эффективность обнаружения ППО со значением F1-меры, превышающем 0,8. Это можно считать неплохим достигнутым результатом. Однако в дальнейших исследованиях предполагается рассмотрение и исследование других моделей машинного обучения с целью повышения эффективности обнаружения ППО. Это, в свою очередь, позволит создать эффективные решения для обнаружения ППО в исходном коде доверенного ПО и учитывать тот факт, что с течением времени количество ППО в мире будет увеличиться.

Рецензент: Липатников Валерий Алексеевич, доктор технических наук, профессор, заслуженный деятель науки Российской Федерации, старший научный сотрудник научно-исследовательского центра Военной академии связи имени Маршала Советского Союза С. М. Буденного, Санкт-Петербург, Россия. E-mail: lipatnikovanl@mail.ru

Литература

1. *Постников Н. А. Принципы безопасной разработки программного обеспечения // Безопасность информационных технологий: сб. науч. ст. по материалам III Всерос. науч.-техн. конф. 2021. Т. 1. С. 95–104.*
2. *Ramirez A., Aiello A., Lincke S. J. A Survey and Comparison of Secure Software Development Standards // 2020 13th CMI Conference on Cybersecurity and Privacy (CMI) – Digital Transformation – Potentials and Challenges(51275). IEEE, New York, NY, USA, 2020. P. 1–6. DOI: 10.1109/CMI51275.2020.9322704.*
3. *Kotenko I., Izrailov K., Buinevich M., Saenko I., Shorey R. Modeling the Development of Energy Network Software, Taking into Account the Detection and Elimination of Vulnerabilities // Energies. 2023. Vol. 16, No. 13. P. 5111. DOI: 10.3390/en16135111.*
4. *Putra A. M., Kabetta H. Implementation of DevSecOps by Integrating Static and Dynamic Security Testing in CI/CD Pipelines // 2022 IEEE International Conference of Computer Science and Information Technology (ICOSNIKOM). IEEE, New York, NY, USA, 2022. P. 1–6. DOI: 10.1109/ICOSNIKOM56551.2022.10034883.*
5. *Kula R. G., Treude C. In war and peace: the impact of world politics on software ecosystems // Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2022). ACM, New York, NY, USA, 2022. P. 1600–1604. DOI: 10.1145/3540250.3560882.*
6. *Lundell B., Butler S., Fischer Th., Gamalielsson J., Brax Ch., Feist J. Effective strategies for using open source software and open standards in organizational contexts: Experiences from the primary and secondary software sectors // IEEE Software. 2022. Vol. 39, No. 1. P. 84–92. DOI: 10.1109/MS.2021.3059036.*
7. *Исабеков В. Protestware: как защитить код? – URL: ttps://dzen.ru/a/Ze7CK2OU7E9Ffcoz (дата обращения: 25.05.2024).*
8. *Christian M., Fabian O., Martin P. DValidator: An approach for validating dependencies in build configurations // Journal of Systems and Software. 2024. Vol. 209. P. 111916. DOI: 10.1016/j.jss.2023.111916.*
9. *Азарычева М. А., Корсунский А. С. Построение и реализация модуля выявления инцидентов на основе сигнатурного метода анализа событий // Автоматизация процессов управления. 2022. № 4 (70). С. 41–50. DOI: 10.35752/1991-2927_2022_4_70_41.*
10. *Anand P., Shankar Singh A. Penetration Testing Security Tools: A Comparison // 2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART). IEEE, New York, NY, USA, 2021. P. 182–184. DOI: 10.1109/SMART52563.2021.9676283.*
11. *Foo D., Yeo J., Xiao H. The Dynamics of Software Composition Analysis. ArXiv: abs/1909.00973. 2019. DOI: 10.48550/arXiv.1909.00973.*
12. *Xia B., Bi T., Xing Z., Lu Q., Zhu L. An Empirical Study on Software Bill of Materials: Where We Stand and the Road Ahead // Proceedings of the 45th International Conference on Software Engineering (ICSE '23). 2023. P. 2630–2642. DOI: 10.1109/ICSE48619.2023.00219.*
13. *Sarker I. H. Machine learning: algorithms, real-world applications and research directions // SN Comput. Sci. 2021. Vol. 2, No. 3. Article No. 160. DOI: 10.1007/s42979-021-00592-x.*
14. *Коротеев М. В. Основы машинного обучения на Python. М.: ООО «Издательство «КноРус», 2024.*
15. *Zhang Y., Sakhanenko L. The naive Bayes classifier for functional data // Statistics & Probability Letters. 2019. Vol. 152. Pp. 137–146. DOI: 10.1016/j.spl.2019.04.017.*

PROTESTWARE: ANALYSIS AND DEFENCE APPROACH BASED ON MACHINE LEARNING

Kotenko I. V.[11], Saenko I. B.[12], Lauta O. S.[13], Yuryev A. S.[14], Zaprudnov M. S.[15]

The purpose of the study: *analysis and systematization of a new type of information security vulnerabilities and attacks, which is Protestware, as well as analysis of existing approaches to counter this threat in order to develop new promising methods for automating the process of detecting Protestware when analyzing program code, taking into account the Secure Software Development Lifecycle (SSDL) methodology.*

Research methods: *system analysis, methods for automating the search for program code vulnerabilities, static code analysis, machine learning using Support Vector Machines and Naive Bayes Classifier.*

Results obtained: *a new type of malicious software, which is Protestware, has been identified and analyzed. Well-known examples and features of such software are analyzed. The risks and problems associated with the spread of Protestware are described. The types and possible sources of Protestware occurrence are identified. The possibilities of using malware detection methods to detect Protestware are considered. Methods are proposed to automate the process of searching for Protestware in relation to large organizations, based on taking into account the principles of SSDL, the use of special static code analyzers and software code inventory technology. An approach to Protestware detection based on the use*

11 Igor V. Kotenko, Dr,Sc., Professor, Honored Worker of Science of the Russian Federation, Chief Scientist and Head of Laboratory of Computer Security Problems at St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS), St. Petersburg, Russia. E-mail: ivkote@comsec.spb.ru

12 Igor B. Saenko, Dr.Sc., Professor, Leading researcher of Laboratory of Computer Security Problems at St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS), St. Petersburg, Russia. E-mail: ibsaen@comsec.spb.ru

13 Oleg S. Lauta, Dr.Sc., Associate Professor, Admiral Makarov State University of Maritime and Inland Shipping, St. Petersburg, Russia. E-mail: laos-82@yandex.ru

14 Artemy S. Yuryev, postgraduate student, Ivannikov Institute for System Programming of the Russian Academy of Sciences (ISP RAS), Moscow, Russia. E-mail: forhhpurpose@yandex.ru

15 Mikhail S. Zaprudnov, student, Federal State Budgetary Educational Institution of Higher Education "MIREA - Russian Technological University", Moscow, Russia. E-mail: mikhail.z2000@gmail.com

of machine learning methods has been implemented and experimentally evaluated. Recommendations are given for selecting machine learning models to improve the efficiency of Protestware detection.

Scientific novelty: *an analysis of works on the topic of Protestware, as well as examples of its manifestation, showed that currently Protestware is a new type of malicious software, for protection against which there are practically no effective means and methods. The results presented in the work summarize known approaches to systematizing Protestware and methods of protection against it. The approach to detecting Protestware implemented in the work differs from the known ones by using machine learning methods using Support Vector Machine models and Naive Bayesian Classifier. The results obtained during the experimental evaluation of the proposed approach make it possible to formulate proposals for the selection of machine learning models that provide the greatest accuracy in Protestware detection.*

Contribution: *Igor Kotenko and Igor Saenko – the general concept of analysis and systematization of Protestware and the sources of its occurrence; Igor Kotenko and Oleg Lauta – formalization of methods for detecting Protestware; Artemy Yuryev and Mikhail Zaprudnov – implementation and experimental evaluation of an approach to Protestware detection based on machine learning; Igor Kotenko and Igor Saenko – discussion of the results of evaluating the proposed approach.*

Keywords: *information security, intrusion detection, vulnerability detection, support vector machine, naive bayes classifier.*

References

1. *Postnikov N. A. [Principles of secure software development] Принципы безопасной разработки программного обеспечения. Security of information technologies: Proceedings of III All-Russian scientific-technical conference. [Безопасность информационных технологий: сб. науч. ст. по материалам III Всерос. науч.-техн. конф.], 2021; Vol. 1. P. 95–104.*
2. *Ramirez A., Aiello A., Lincke S. J. A Survey and Comparison of Secure Software Development Standards // 2020 13th CMI Conference on Cybersecurity and Privacy (CMI) – Digital Transformation – Potentials and Challenges(51275). IEEE, New York, NY, USA, 2020. P. 1–6. DOI: 10.1109/CMI51275.2020.9322704.*
3. *Kotenko I., Izrailov K., Buinevich M., Saenko I., Shorey R. Modeling the Development of Energy Network Software, Taking into Account the Detection and Elimination of Vulnerabilities // Energies. 2023. Vol. 16, No. 13. P. 5111. DOI: 10.3390/en16135111.*
4. *Putra A. M., Kabetta H. Implementation of DevSecOps by Integrating Static and Dynamic Security Testing in CI/CD Pipelines // 2022 IEEE International Conference of Computer Science and Information Technology (ICOSNIKOM). IEEE, New York, NY, USA, 2022. P. 1–6. DOI: 10.1109/ICOSNIKOM56551.2022.10034883.*
5. *Kula R. G., Treude C. In war and peace: the impact of world politics on software ecosystems // Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2022). ACM, New York, NY, USA, 2022. P. 1600–1604. DOI: 10.1145/3540250.3560882.*
6. *Lundell B., Butler S., Fischer Th., Gamalielsson J., Brax Ch., Feist J. Effective Strategies for Using Open Source Software and Open Standards in Organizational Contexts: Experiences From the Primary and Secondary Software Sectors // IEEE Software. 2022. Vol. 39, No. 1. P. 84–92. DOI: 10.1109/MS.2021.3059036.*
7. *Isabekov V. [Protestware: how to protect code?] Protestware: как защитить код? – URL: ttps://dzen.ru/a/Ze7CK2OU7E9Ffcoz (accessed: 05/25/2024).*
8. *Christian M., Fabian O., Martin P. DValidator: An approach for validating dependencies in build configurations // Journal of Systems and Software. 2024. Vol. 209. P. 111916. DOI: 10.1016/j.jss.2023.111916.*
9. *Azarycheva M. A., Korsunsky A. S. [Construction and implementation of an incident detection module based on the signature method of event analysis] Построение и реализация модуля выявления инцидентов на основе сигнатурного метода анализа событий. Automation of Control Processes [Автоматизация процессов управления]. 2022. No. 4 (70). P. 41–50. DOI: 10.35752/1991-2927_2022_4_70_41.*
10. *Anand P., Shankar Singh A. Penetration Testing Security Tools: A Comparison // 2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART). IEEE, New York, NY, USA, 2021. P. 182–184. DOI: 10.1109/SMART52563.2021.9676283.*
11. *Foo D., Yeo J., Xiao H. The Dynamics of Software Composition Analysis. ArXiv: abs/1909.00973. 2019. DOI: 10.48550/arXiv.1909.00973.*
12. *Xia B., Bi T., Xing Z., Lu Q., Zhu L. An Empirical Study on Software Bill of Materials: Where We Stand and the Road Ahead // Proceedings of the 45th International Conference on Software Engineering (ICSE '23). 2023. P. 2630–2642. DOI: 10.1109/ICSE48619.2023.00219.*
13. *Sarker I. H. Machine learning: algorithms, real-world applications and research directions // SN Comput. Sci. 2021. Vol. 2, No. 3. Article No. 160. DOI: 10.1007/s42979-021-00592-x.*
14. *Koroteev M. V. [Machine learning basics in Python] Основы машинного обучения на Python. М.: LLC «Publishing House «KnoRus», 2024.*
15. *Zhang Y., Sakhanenko L. The naive Bayes classifier for functional data // Statistics & Probability Letters. 2019. Vol. 152. P. 137–146. DOI: 10.1016/j.spl.2019.04.017.*

ИНТЕЛЛЕКТУАЛЬНЫЕ МЕТОДЫ ОБЕСПЕЧЕНИЯ КИБЕРБЕЗОПАСНОСТИ МУЛЬТИАГЕНТНЫХ СИСТЕМ УПРАВЛЕНИЯ МИКРОСЕТЯМИ

Гурина Л. А.[1], Томин Н. В.[2]

DOI: 10.21681/2311-3456-2024-6-53-64

Цель исследования: *разработка методов обнаружения и подавления последствий кибератак при вторичном регулировании напряжения в мультиагентных системах управления киберфизическими микросетями.*

Методы исследования: *методы машинного обучения, вероятностные методы*

Результат исследования: *разработаны алгоритм изоляционного леса для автоматического обнаружения кибератак и алгоритм восстановления качества данных на базе метода k-ближайших соседей.*

Научная новизна *состоит в том, что предложенный метод обнаружения кибератак и повышения качества информации создает возможности робастности, адаптации и восстановления мультиагентных систем при нарушениях кибербезопасности.*

Ключевые слова: *киберфизическая микросеть, идентификация кибератак, обнаружение плохих данных, повышение качества информации.*

Введение

Интеллектуальные энергосистемы (ИЭС) возникли с целью повысить гибкость, эффективность, надежность и безопасность энергосетей за счет использования передовых технологий измерения, связи и управления в реальном времени [1, 2]. Преимуществами эксплуатации ИЭС является внедрение информационных, цифровых и коммуникационных технологий, которые позволяют повысить наблюдаемость сети, несмотря на различные неопределенности из-за внутренних и внешних воздействий, тем самым позволяя предпринимать дополнительные корректирующие, превентивные управляющие воздействия. Такая интеграция в ИЭС привела к появлению различных киберфизических взаимозависимостей, что способствует увеличению уязвимостей к киберугрозам на различных уровнях ИЭС: от высоковольтных систем передачи до распределительных сетей и микросетей. Последние в силу активного внедрения объектов распределенной энергетики с привлечением устройств силовой электроники и различных систем управления имеют достаточно сложную информационно-коммуникационную инфраструктуру, отказы и сбои в которой могут оказывать существенно влияние на надежность микросетей.

Концепция микросетей была предложена в качестве организационного принципа управления потоками информации и энергии для сетей с распределенными источниками энергии. В общем смысле микросеть представляет собой объединение источников генерации, нагрузок и систем накопления энергии. С появлением киберфизических микросетей (КФМС) при цифровой трансформации поверхность атак возрастает, что затрудняет обеспечение кибербезопасности (КБ) традиционными методами. Злоумышленники выбирают инновационные методы обхода механизмов безопасности, поэтому существует необходимость разработки и внедрения интеллектуальных методов обеспечения КБ КФМС.

В условиях роста киберугроз традиционные программные системы могут идентифицировать кибератаки (КА) и соответствующим образом модернизировать их, тогда как способность искусственного интеллекта (ИИ) учиться на прошлом опыте может помочь адаптироваться к новым поступающим угрозам. Методы ИИ позволяют не только обнаруживать шаблоны атак в данных и аномалии в них, но и прогнозировать КА. Последовательный анализ шаблонов [3] является одним из методом анализа данных, который позволяет выявить закономерности атак и обнаружить какую-либо вредоносную или аномальную активность.

Одним из основных ограничений использования ИИ в обеспечении КБ является доступность наборов данных. Для обучения модели ИИ используются ретроспективные данные, содержащие сведения о вредоносном программном обеспечении (ПО), шаблонах атак и событиях атак. Используя сигнатуры

1 Гурина Людмила Александровна, кандидат технических наук, доцент, старший научный сотрудник Лаборатории управления функционированием электроэнергетических систем Института систем энергетики им. Л. А. Мелентьева СО РАН, Иркутск, Россия. E-mail: gurina@isem.irk.ru

2 Томин Никита Викторович, кандидат технических наук, заведующий Лабораторией управления функционированием электроэнергетических систем Института систем энергетики им. Л. А. Мелентьева СО РАН, Иркутск, Россия. E-mail: tomin.nv@gmail.com

событий в наборе данных, ИИ позволяет обнаруживать КА. Недостатком этого метода является сложность создания набора данных. Несмотря на то, что методы ИИ связаны с разработкой интеллектуальных и адаптивных систем, подготовка набора данных на предварительном этапе считается существенным препятствием для использования ИИ в обеспечении КБ. Кроме того, злоумышленники также могут использовать ИИ для обхода механизмов безопасности, поэтому важно знать, как о преимуществах, так и об угрозах, которые представляют собой ИИ в сфере КБ.

Первоначально безопасность ограничивалась атаками на информационные потоки с использованием таких методов, как вредоносное ПО, шпионское ПО и программы-вымогатели. Обеспечение КБ гарантировалось за счет использования антивирусов, межсетевых экранов и систем обнаружения вторжений (IDS). Увеличение числа взаимосвязей и взаимозависимостей между объектами информационно-коммуникационной инфраструктуры КФМС также способствует росту КА. В последнее время алгоритмы машинного обучения стали использоваться для обеспечения КБ различных киберфизических систем [4, 5]. Использование ИИ и, особенно, машинного обучения для обеспечения КБ началось с его внедрения в IDS, что позволяло обнаруживать вредоносное ПО и КА в информационно-коммуникационной инфраструктуре КФМС.

Однако растет обеспокоенность по поводу использования ИИ в сфере КБ. Недавние исследования показали, что системы, КБ которых зависит от алгоритмов машинного обучения, также подвержены различным формам КА. Алгоритмы машинного обучения зависят от данных и, соответственно, делают выводы или прогнозы на основе данных, генерируемых различными датчиками в киберфизических системах. Для формирования успешно реализуемых КА, напр., в КФМС или других объектах ИЭС, требуется разработка методов по манипуляциям с данными, в результате чего могут быть сформированы неправильные управляющие воздействия [6].

Таким образом, использование алгоритмов ИИ и машинного обучения для защиты КФМС также может использоваться злоумышленниками для атаки на них. Такие атаки обладают большим потенциалом, поскольку они более сложны, быстры, трудно обнаруживаемы и подавляемы, поэтому целью данного исследования является разработка метода обнаружения и подавления последствий КА на основе алгоритмов машинного обучения, позволяющих реализовывать стратегии обеспечения КБ КФМС.

Стратегии управления КФМС на основе мультиагентных систем с учетом обеспечения КБ

По аналогии с объектами большой энергетики сегодня для управления микросетями используются три основные стратегии управления в зависимости от их архитектуры: 1) децентрализованные; 2) централизованные и 3) распределенные [7]. Такие архитектуры могут быть представлены и как мультиагентные системы (МАС), когда входные управления могут по-разному зависеть от агентов в зависимости от состояний. С точки зрения МАС такие архитектуры можно математически выразить как:

$$u_i = \begin{cases} u_i(\cup_{j\in v_i} x_i) & \text{(Централизованное)} \\ u_i(x_i \cup_{j\in N_i} x_j) & \text{(Распределенное)} \\ u_i(x_i) & \text{(Децентрализованное)} \end{cases} \quad (1)$$

где u_i – сигнал управления; x_i – состояние i-го агента; v – набор всех агентов.

Согласно (1) сигнал управления u_i может зависеть только от состояния агентов x_i (децентрализованное управление), либо от их x_i и x_j для всех $j \in N_i$ (распределенное управление) (рис. 1). При этом N определяет набор соседних агентов МАС. Цель управления зависит от приложения и многочисленна. Одной из наиболее хорошо изученных задач управления является задача консенсуса, где цель управления агентов состоит в достижении общего состояния, т.е. $\lim_{t\to\infty} |x_i(t) - x_j(t)| = 0 \; \forall_{i,j} \in N_i$.

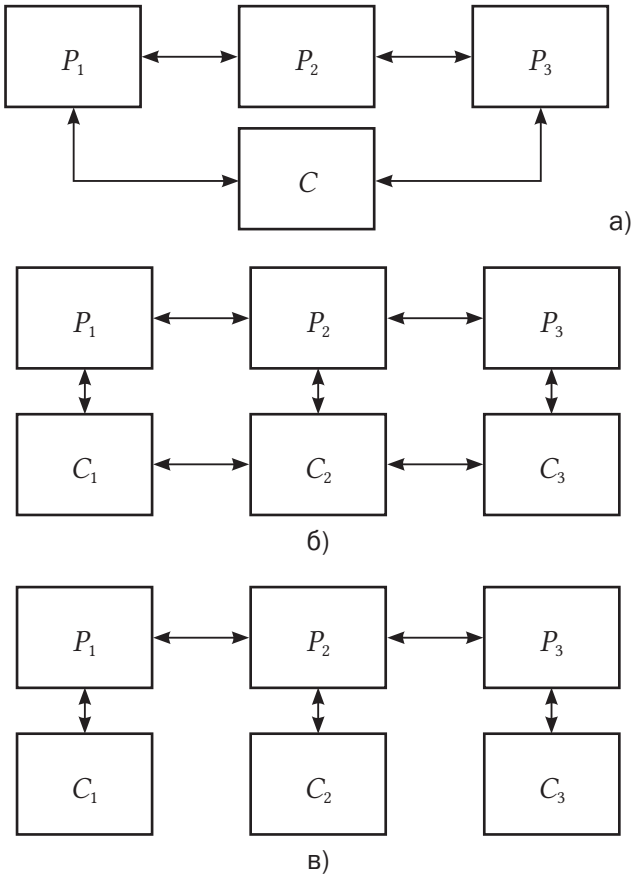


Рис. 1. Иллюстрация (а) децентрализованной, (б) распределенной и (в) децентрализованной архитектур управления. P_1, P_2 и P_3 представляют собой объекты, управляемые контроллерами C_1, C_2 и C_3 соответственно. Адаптировано из [8].

Каждая из стратегий имеет свои преимущества и недостатки. Например, централизованные стратегии способны обеспечить наиболее точное и координированное управление, но при этом обычно требуют полной модели электрической сети, а также имеют проблемы с конфиденциальностью/временем отклика, потери пропускной способности линий связи и единой точки отказа [9]. Децентрализованные и распределенные подходы отчасти лишены подобных проблем, однако связаны с проблемой согласования распределенных управляющих воздействий [10]. Основным принципом таких подходов является обмен информацией между соседними агентами (контроллерами) с использованием распределенного протокола и нахождение консенсуса [11]. На сегодняшний день именно мультиагентные принципы управления КФМС и активными распределительными сетями, включающими объекты распределенной энергетики, рассматриваются как наиболее предпочтительные.

А. Классификация КА на мультиагентные системы управления КФМС

КА на мультиагентные системы управления КФМС можно разделить на атаки на целостность, доступность и конфиденциальность данных (табл. 1) [12]. Наиболее опасными по последствиям КА являются атаки внедрения ложных данных (FDI-атаки), атаки захвата контроллера (Hijacking-атаки), атаки «человек посередине» (MITM-атаки), вредоносное ПО и атаки отказа в обслуживании (DoS-атаки).

Основной целью FDI-атак является изменение значений данных, передаваемых по каналам связи [13]. FDI-атаки увеличивают вычислительную нагрузку на контроллеры, вызывая сбои в управлении устройствами, а также приводят к небалансу мощностей. Злоумышленник нацелен на уязвимости в каналах связи и вводит ложные данные в существующие значения, используя различные методы внедрения [14–16]. Злоумышленник может изменять и удалять данные, что приводит к нарушению целостности и доступности данных.

Вредоносное ПО – это программы, предназначенные для оказания нежелательного или вредоносного воздействия на информационные системы, которые стали серьезной угрозой КБ КФМС. В целом вредоносные программы подразделяются на следующие категории [17]: вирусы, бэкдоры, трояны, черви и шпионское ПО. На практике вредоносное ПО часто демонстрирует характеристики двух или более категорий, например, червь, содержащий полезную нагрузку, может установить черный ход для обеспечения удаленного доступа.

В информационных системах можно нарушить передачу данных в каналах связи между контроллерами с помощью DoS-атак различными способами [18, 19]. Например, можно полностью заблокировать полосу пропускания канала, наводнив его ложной информацией или путем введения буфера в поток коммуникационной связи, что наиболее опасно из-за трудности его идентификации. Последствиями DoS-атак является задержка получения или потеря данных, влекущих за собой нарушение управления КФМС.

Б. Обзор методов машинного обучения, применяемых для обеспечения кибербезопасности КФМС

КА в КФМС не только вызывают проблемы с качеством данных, но и могут привести к сбоям и отказам объектов информационно-коммуникационной инфраструктуры и, как следствие, к нарушениям функционирования самой КФМС. Обмен данными между контроллерами КФМС необходим для достижения эффективного управления ими. Постоянный мониторинг и анализ данных играет важную роль в обеспечении качества данных при КА как на уровне устройства, так и на уровне сети. Важна разработка алгоритмов обнаружения и смягчения/подавления влияния КА на качество данных как на уровне устройства, так и на уровне сети.

Таблица 1.

Классификация кибератак, нарушающих качество данных

Целостность	Доступность	Конфиденциальность
FDI-атака	Jamming-атака	Социальная инженерия
Hijacking-атака	Wormhole-атака	Подслушивание
Подделка данных	DoS-атака	Анализ трафика
Атака повторного воспроизведения	DDos-атака	Несанкционированный доступ
Wormhole-атака	Переполнение буфера	Кража паролей
Spoofing-атака	Puppet-атака	Атака «Человек посередине»,
Атака модификации	Time Synchronization	Атака перехвата
Атака «Человек посередине»	Masquerade-атака	Атака повторного воспроизведения,
Masquerade-атака	Атака «Человек посередине»	Masquerade–атака
	Spoofing-атака	

Первичные и вторичные уровни управления КФМС, несущие важную информацию от контроллеров, наиболее подвержены КА, вызывающим ошибки управления и нарушение функционирование КФМС. Для мониторинга и предотвращения КА на энергетические системы сегодня успешно используются различные алгоритмы машинного обучения. В [20] на основе контролируемых и полуконтролируемых алгоритмов приведена классификация достоверных и искаженных измерений и разработана структура эффективного обнаружения КА. Сильная зависимость от цифровых и коммуникационных технологий увеличивает уязвимости КФМС к атакам внедрения ложных данных, которые могут обойти механизмы обнаружения ошибочных данных. Существующие меры по смягчению последствий FDI-атак либо сосредоточены на избыточных измерениях, либо защищают набор основных измерений. В [21] предложили систему для обнаружения ошибок измерений в результате FDI-атак, основанную на глубоком обучении, для обнаружения аномалий временных рядов используется сверточная нейронная сеть (CNN) и сеть долгосрочной краткосрочной памяти (LSTM). Для оценивания системных переменных учитываются как измерения данных, так и функции сетевого уровня для совместного изучения состояний системы.

Распределенные атаки в отказе обслуживания на контроллеры могут привести к переполнению буфера и потере информации. Авторами [22] предложен алгоритм обнаружения распределенных атак в отказе обслуживания на ранней стадии при помощи шаблона трафика, сгенерированного из набора данных, на основе машины опорных векторов, представляющей собой обученный классификатор. В [23] проанализированы угрозы технологии «интернета вещей». Представлена стратегия обнаружения КА на Интернет вещей, которая объединяет модели генетических алгоритмов и искусственных нейронных сетей. В [24] представлен анализ КА на системы безопасности как на модели IDS, так и на модели беспроводных сенсорных сетей, а также предложены решения по обеспечению безопасности для их устранения на основе модели случайного леса. В [25] для обнаружения вредоносного ПО использована также нейросеть глубокого обучения.

Система прогнозирования КА – это часть системы КБ, которая анализирует данные сетевого трафика в режиме реального времени и прогнозирует КА. Основной мотивацией для прогнозирования КА является повышение точности классификатора в обнаружении КА [26]. Было исследовано и разработано множество подходов к повышению точности прогнозирования КА. Одним из них является машинное обучение [27, 28], которое можно применять как к моделям вторжений, так и к обнаружению КА. Для повышения точности прогнозирования КА в [29] учитывается обнаружения выбросов на основе изоляционного леса с учетом предварительной обработки – несбалансированности набора данных, категориального кодирования признаков и масштабирования признаков.

Метод обнаружения и подавления последствий КА в мультиагентных системах управления КФМС на основе алгоритма изоляционного леса

В [11] была предложена модель мультиагентного контроллера инверторов для распределенного вторичного управления напряжением в электрических сетях и микросетях с высоким уровнем ВИЭ с использованием мультиагентного обучения с подкреплением (англ. Multi-Agent Reinforcement Learning). Такой контроллер реализует управление по статизму $a_{i,t}$, когда амплитуды напряжения V_i инверторов изменяются в зависимости от отклонений реактивной мощности Q_i^m от заданных уставок Q_i^d : $u_i^V = V_i^d - k_{Qi}(Q_i^m - Q_i^d)$, где u_i^V – управляющий сигнал для амплитуды напряжения V_i, V_i^d – желаемая амплитуда напряжения, k_{Qi} – коэффициент усиления по напряжению. Состояние каждого агента i выбирается как $st = (\delta_i, P_i, Q_i, i_{odi}, i_{oqi}, i_{bdi}, i_{bqi}, v_{bdi}, v_{bqi})$ для характеристики режимов распределённых генераторов (РГ), подключенных через инверторы, где δ_i – измеренный опорный угол (фаза); P_i, Q_i – активная и реактивная мощности соответственно; $i_{odi}, i_{oqi}, i_{bdi}, i_{bqi}$ – выходные токи d-q генератора i и напрямую подключенные шины, соответственно; v_{bdi}, v_{bqi} – выходные напряжения d-q подключенной шины соответственно. При этом наблюдение каждого агента включает как свое локальное состояние, так и сообщения от своих соседей: $o_{i,t} = S_{i,t} \cup m_{i,t}$, где $m_{i,t}$ – коммуникационное сообщение, полученное от соседних агентов $j \in N_i$. Целью такого контроллера является максимизация глобального вознаграждения $R_{i,t} = \sum_{k=0}^{T} \gamma^k \sum_{j \in v} \alpha(d_{i,j})\, r_{i,t+k}$, где $\alpha(d_{i,j})\, r_{i,t+k}$ – пространственная функция дисконтирования, $d_{i,j}$ – расстояние между агентом i и j, $r_{i,t}$ – вознаграждение агента i на временном шаге t. При сформулированном вознаграждении $r_{i,t}$ агенты с «аварийными» напряжениями ($V_i \in |0,0.8| \cup |1.25,\infty|$) получат большой штраф, и наоборот агенты с напряжением, близким к 1 о.е. ($V_i \in (|0.95,1.05|)$ получат положительное вознаграждение.

Ранее авторами в [30] была проведена оценка робастности такой концепции вторичного мультиагентного управления к различным типам КА на контроллеры. Испытания показали, что при FDI- и Hijacking-атаках качество регулирования напряжения ухудшается, но не носит критический характер. Во многом это связано с тем, что предложенный

мультиагентный контроллер использует централизованную схему обучения агентов с децентрализованным исполнением, где каждый агент имеет свои собственные актёр-критические нейросети, и их стратегия обновляется независимо с учётом информации от соседних агентов (инверторов) $h_{i,t}$ для повышения скорости сходимости решения и эффективности обучения. Однако это не означает, что такая система регулирования абсолютна робастна к КА, в этом случает требуется встроенная интеллектуальная процедура обнаружения и подавления последствий КА.

В настоящей работе предложена двухэтапная процедура обнаружения и подавления последствий КА с использованием следующих методов машинного обучения без учителя: изоляционный лес (англ. Isolation Forest) и k-ближайших соседей (англ. k-nearest neighbors algorithm, k-NN). Структура ее адаптации в вышеописанную структуру мультиагентного вторичного управления напряжением в КФМС показана на рис. 2 и более подробно раскрыта далее.

А. Алгоритм изоляционного леса для автоматического обнаружения КА

Изоляционный лес – это алгоритм машинного обучения для обнаружения аномалий данных с помощью двоичных деревьев [31]. Алгоритм для обнаружения аномалий опирается на характеристики аномалий, т.е. на то, что их мало и они различны. Суть алгоритма заключается в том, что аномальные точки данных легче отделить от остальной части выборки. Чтобы изолировать точку данных, алгоритм рекурсивно генерирует фрагменты выборки, случайным образом выбирая атрибут, а затем случайным образом выбирая значение разделения между минимальным и максимальным значениями, разрешенными для этого атрибута. Обнаружение аномалий с помощью изоляционного леса – это процесс, состоящий из двух основных этапов:

1) на первом этапе для построения двоичных деревьев используется набор обучающих данных;
2) на втором – каждый экземпляр в тестовом наборе проходит через эти деревья, и экземпляру присваивается надлежащая «оценка аномалии».

После того как всем экземплярам в тестовом наборе присвоен показатель аномалии, можно пометить как «аномалию» любую точку, показатель которой превышает заранее определенный порог, который зависит от области, к которой применяется анализ.

Алгоритм расчета оценки аномалии точки данных основан на наблюдении, что структура дерева эквивалентна структуре двоичных деревьев поиска (англ. Binary Search Trees, BST): завершение внешнего узла дерева соответствует неудачному поиску в BST. Как следствие, оценка среднего $h(x)$ для завершения внешнего узла то же, что и для неудачных поисков в BST, то есть

$$c(m) = \begin{cases} 2H(m-1) - \dfrac{2(m-1)}{n} & \text{для } m > 2 \\ 1 & \text{для } m = 2 \\ 0 & \text{в противном случае} \end{cases} \qquad (2)$$

где n – размер тестовых данных, m – размер выборки и H – номер гармоники, который можно оценить по формуле $H(i) = ln(i) - \gamma$, где γ = 0,5772156649 – постоянная Эйлера-Машерони.

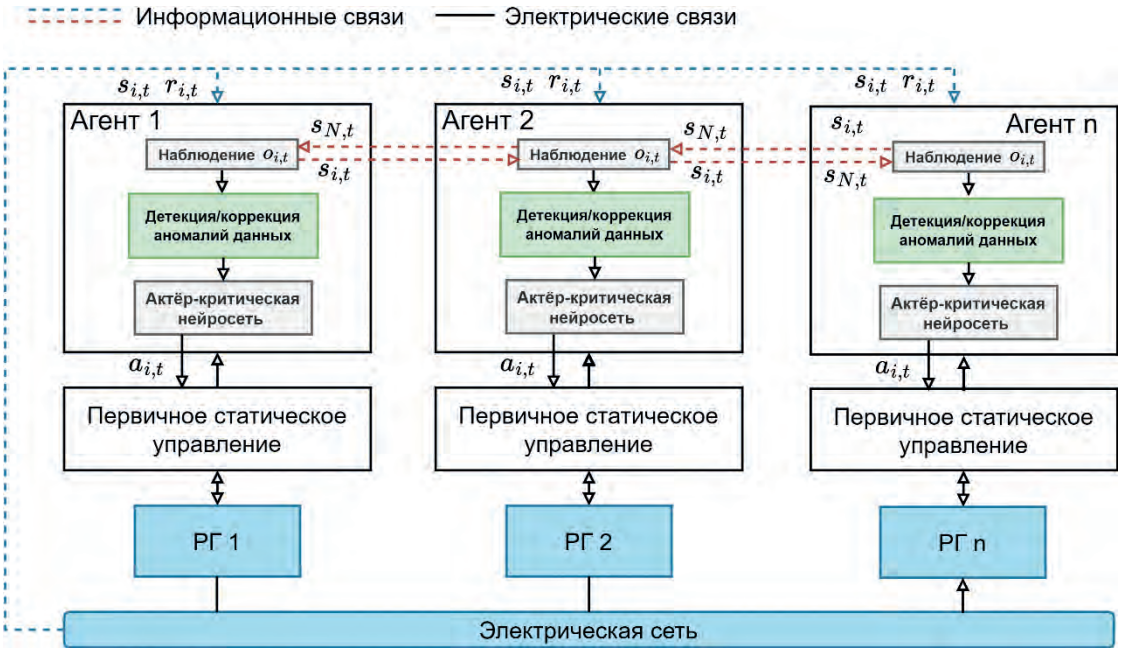


Рис. 2. Общая структура мультиагентного вторичного управления контроллерами инверторов РГ на базе MARL с функцией защиты от КА на основе двухэтапной процедуры

Значение $c(m)$ в (2) представляет собой среднее значение $h(x)$ данный m, поэтому мы можем использовать его для нормализации x и получить оценку оценки аномалии для данного экземпляра x:

$$s(x,m) = 2^{\frac{-E(h(x))}{c(m)}}, \quad (3)$$

где $E(h(x))$ – среднее значение $h(x)$ из коллекции деревьев. Интересно отметить, что для любого данного случая

- если s близко к 1, то x наиболее вероятно является аномалией (рис. 3);
- если s меньше, чем 0.5, то x наиболее вероятно нормальное значение;
- если для данной выборки всем экземплярам присвоен показатель аномалии около 0.5, то можно с уверенностью предположить, что в выборке нет аномалий.

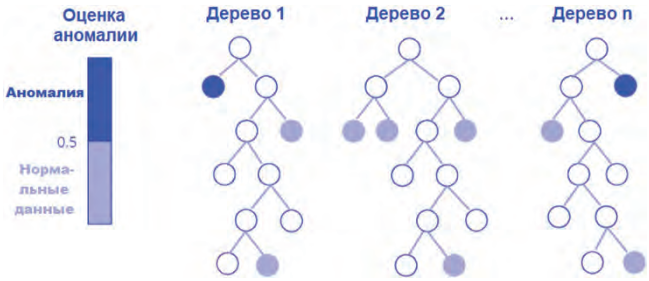


Рис. 3. Общая схема работы метода изоляционного леса

В предлагаемой процедуре в качестве x выступают наблюдения агента $o_{i,t} = S_{i,t} \cup m_{i,t}$, когда потенциальная кибератака может вызывать аномалии как в локальных входных данных агента $S_{i,t}$, так и в сигналах, которые приходят от соседних агентов $m_{i,t}$. В процессе обучения агентов по методу MARL параллельно обучается модель изоляционного леса на данных $o_{i,t}$, которые приходят к агентам. В итоге каждому набору $o_{i,t}$ присваивается значение $S(o_{i,t},m)$, представляющее собой оценку аномалии.

Однако при обнаружении аномалии в каком-либо наборе $o_{i,t}$ мы не можем его просто исключить, так как это фактически означает исключение контроллера из процесса регулирования напряжения. В этом случае помимо обнаружения аномалий (первый этап предложенной процедуры) требуется решение задачи восстановления качества данных в искаженном наборе $o_{i,t}$. Поэтому на втором этапе процедуры предложено применение метода k-ближайших соседей для повышения качества данных.

Б. Алгоритм восстановления качества данных на базе метода k-ближайших соседей

Метод k-ближайших соседей (KNN) для заполнения пропущенных значений в данных основан на предположении, что значения в неиспорченных ячейках зависят от значений в их окрестности [32]. Для каждой ячейки с пропущенным значением x_{ij} во входной матрице данных X размером $n \times m$, где n – количество наблюдений, а m – количество признаков, метод выполняет следующие шаги:

1. Определение окрестности: находятся k ближайших наблюдений к x_{ij} среди всех оставшихся наблюдений, где k – заданное число. Расстояние между наблюдениями обычно измеряется с помощью метрики Минковского: $d(x_i,x_j) = (\sum_{k=1}^{m} (x_{ik} - x_{jk})^p)^{1/p}$. Норма Минковского принимает форму евклидова расстояния или расстояния L2, когда $p = 2$, или форму расстояния Манхэттена, когда $p = 1$; были описаны другие дробные нормы для $p < 1$ [33]. В экспериментах данного исследования было принято $p = 2$.

2. Заполнение пропусков: значение пропущенной ячейки x_{ij} оценивается как среднее арифметическое значений k ближайших соседей: $\hat{x}_{ij} = \frac{1}{K} \sum k \in N(i,k)\ x_{kj}$, где $N(i,k)$ — множество индексов k ближайших соседей к наблюдению i.

Метод KNN для заполнения пропущенных значений может быть использован как для числовых, так и для категориальных данных с небольшими модификациями в подходе к определению близости и способу заполнения пропусков. В данном случае под x_{ij} понимаются «повреждённые» значения вектора наблюдения агента $o_{i,t} = S_{i,t} \cup m_{i,t}$. Так, после КА могут возникнуть аномалии в каких-либо значения вектора локального состояния агента $s_t = (\delta_i, P_i, Q_i, i_{odi}, i_{oqi}, i_{bdi}, i_{bqi}, v_{bdi}, v_{bqi})$ (напр., в $i_{odi}, i_{oqi}, i_{bdi}$) и тогда их восстановление будет зависит от «нормальных» значений других параметров вектора, лежащих в окрестности.

Таким образом, предложенная двухэтапная процедура обнаружения КА и повышения качества данных может быть проиллюстрирована общей блок-схемой, представленной на рис. 4.

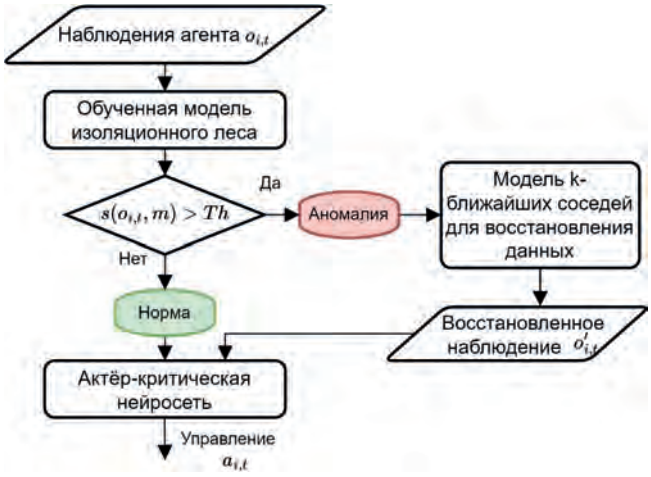


Рис. 4. Общая блок-схема предложенной двухэтапной процедуры обнаружения КА и повышения качества данных с использованием алгоритмов машинного обучения без учителя

Пример

Для оценки эффективности предложенной двухэтапной процедуры обнаружения КА и повышения качества данных, рассмотрим модели сообщества КФМС с шестью распределенными генераторами (РГ) (солнечные фотоэлектрические преобразователи), подключаемыми через AC/DC инверторы к сети переменного тока (рис. 5). При этом КФМС подключены к внешней электрической сети и имеют единую мультиагентную систему вторичного регулирования напряжения, показанную на рис. 2. По аналогии с [30] были рассмотрены несколько сценариев, когда контроллеры инверторов 3, 5 и 6 (агенты) подвергаются воздействию следующих вредоносных воздействий: FDI- и Hijacking-атаки. Для реализации методов изоляционного леса и k-ближайших соседей была использована Python-библиотека Scikit-learn для машинного обучения [34]. Для реализации MARL и модели микросетей были задействованы Python-библиотеки PyTorch и Powernet [35].

Помимо КА, для демонстрации эффективности предложенного варианта вторичного регулирования были смоделированы случайные колебания нагрузки, которые приводят к падению напряжения. Для этого были добавлены случайные изменения нагрузки по всей сети с отклонениями ± 20% от номинальных значений, а также случайные возмущения в диапазоне ±5% для каждой нагрузки схемы, показанной на рис. 5. При этом все агенты в схеме микросетей контролировались со временем выборки 0,05 с, и каждый агент мог связываться со своими соседями через локальные граничные каналы связи. Первичное управление нижнего уровня реализовано по аналогии с [36]. Глобальной целью управления является регулирование всех напряжений РГ до опорного значения 1 о.е.

На рис. 6 показаны оценки аномалий на базе метода изоляционного леса для каждой точки данных в наборе данных при сценарии Hijacking-атаки. Эти оценки соответствуют вектору наблюдения агента $o_{i,t}$, т.е. входному вектору контроллера инвертора. Цвет каждой точки представляет ее оценку аномалии: более темные цвета указывают на более высокие оценки аномальности, а более светлые цвета указывают на более низкие оценки. Таким образом, точки самых темных цветов представляют собой наиболее аномальные точки данных, поскольку они наиболее изолированы от остальных данных. С другой стороны, точки самых светлых цветов являются наименее аномальными точками данных, поскольку они наиболее близки к остальным данным. Хорошо видно, что алгоритм изоляционного леса точно обнаруживает потенциальные аномалии в наблюдениях $o_{i,t}$ подверженных КА агентов 3, 5 и 6 (рис. 6б).

На рис. 7 и 8 представлены результаты моделирования системы регулирования напряжения при возмущении нагрузки для различных сценарных экспериментов FDI- и Hijacking-атаках. В отсутствие КА агенты мультиагентной системы управления хорошо справляются с задачей вторичного регулирования напряжения при случайных колебаниях нагрузки, приводящих к падению напряжения. Фактически с момента времени $t = 0.4\ c.$ агенты формируют кооперативную стратегию, которая успешно восстанавливает и поддерживает все регулируемые напряжения близко к номинальному значению 1 отн.ед. (рис. 7а и 8а).

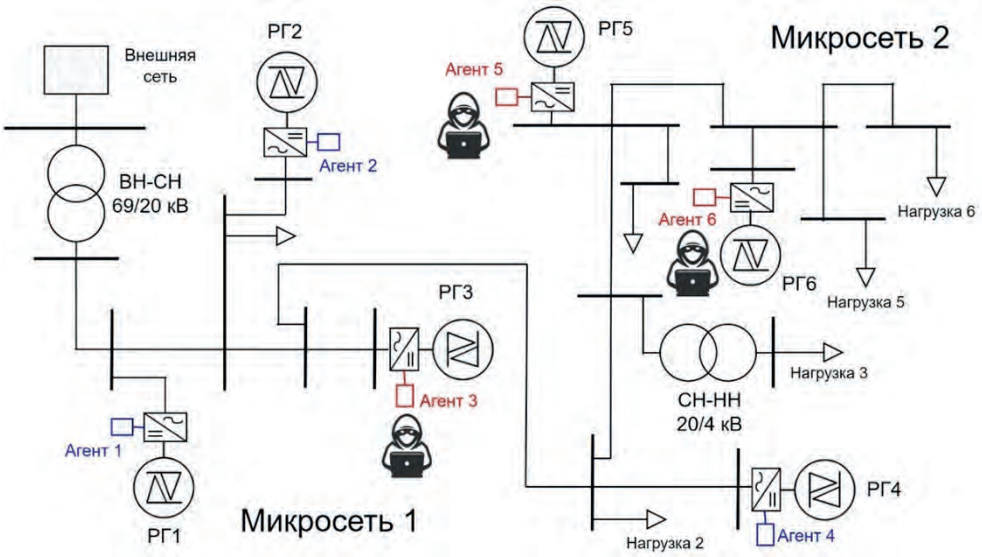
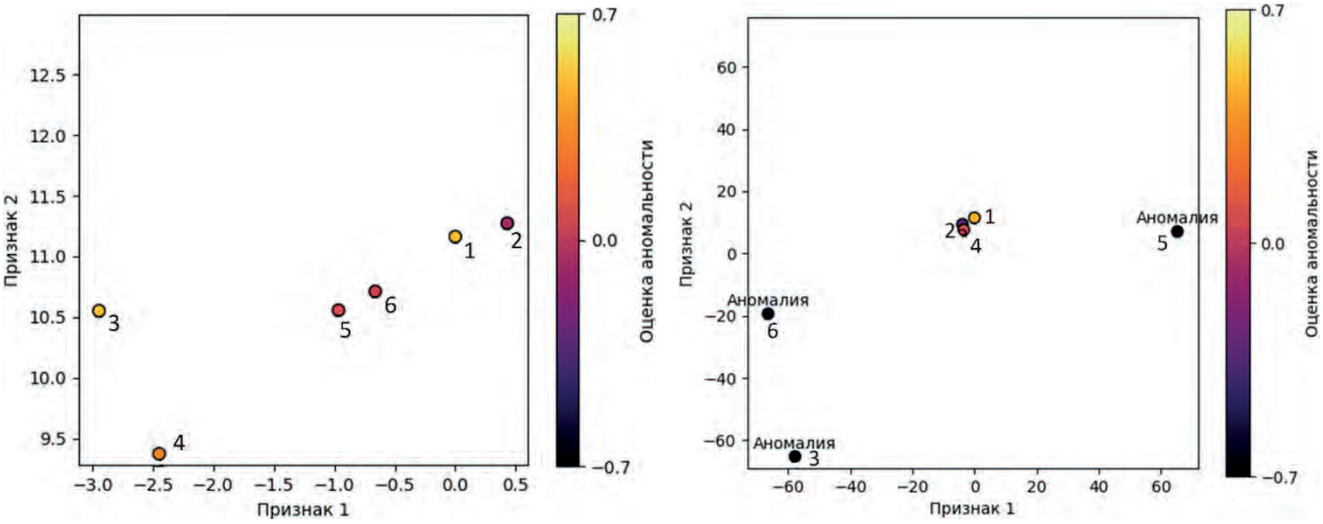
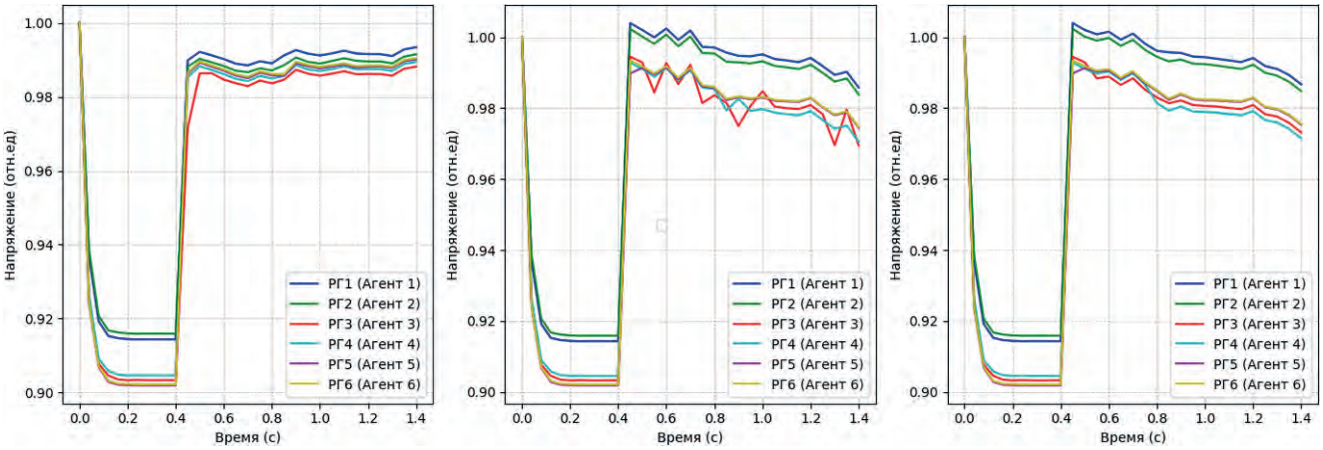


Рис. 5. Тестовая схема двух взаимосвязанных микросетей, имеющих общую систему мультиагентного вторичного регулирования напряжения



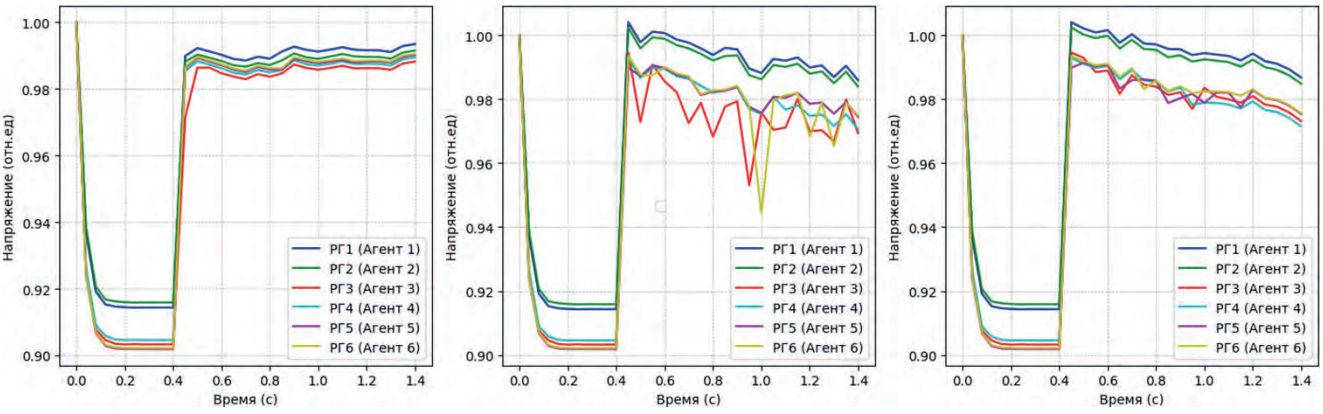
а) В отсутствие КА *б) Hijacking-атака*

Рис. 6. Визуализация оценки векторов наблюдений агентов (контроллеров) на предмет аномальных данных с использованием изоляционного леса



а) В отсутствие КА *б) FDI-атака* *в) Подавление последствий КА*

Рис. 7. Результаты моделирования поведения агентов системы регулирования напряжения при возмущении нагрузки для различных сценарных экспериментов FDI-атаки



а) В отсутствие КА *б) Hijacking-атака* *в) Подавление последствий КА*

Рис. 8. Результаты моделирования поведения агентов системы регулирования напряжения при возмущении нагрузки для различных сценарных экспериментов атаки захвата контроллеров 3, 5, 6.

Таблица 2.

Пример наблюдения $o_{i,t}$ агента-контроллера №3 для различных сценариев КА

№	Сценарий	Вектор наблюдения агента $o_{i,t}$								
		δ_i	P_i	Q_i	i_{odi}	i_{oqi}	i_{bdi}	i_{bqi}	v_{bdi}	v_{bqi}
1	Без кибератаки	–2.53	9.18	10.03	4.19	–5.38	–3.42	3.36	–3.45	3.35
2	Hijacking-атака	–13.6	–7.44	–119	–175	–47.2	–47.1	21.4	–68.1	–75.1
3	Восстановление данных	–0.17	9.58	7.02	5.02	–3.39	–3.41	3.35	–3.45	3.36

При различных вариантах FDI- и Hijacking-атаках на 3, 5 и 6 контроллеры качество вторичного регулирования ожидаемо ухудшается (рис. 7б и 8б). Это выражается в появлении колебаний при регулировании и снижении эффективности общего поддержания уровней напряжения близких к номинальным. Особенно явно это проявляется при сценарии Hijacking-атаке, которая в рассмотренном примере соответствует полному захвату контроллеров, т.е. в $o^*_{i,t} = (1 - \alpha)\, o_{i,t} - \alpha x^{\alpha}_{i,t}$ (где $o^*_{i,t}$ – модифицированное наблюдение агента; $x^{\alpha}_{i,t}$ – ложные данные), коэффициент $\alpha = 1$, что означает атаку на инвертор с полной заменой корректных наблюдений. Возникновение колебаний связано с ухудшением согласованности агентов, т.е. установления консенсуса, в следствии того, что от некоторых агентов поступает ложные данные.

Применение предложенной двухэтапной процедуры, в частности метода k-ближайших соседей, позволяет повысить качество данных подверженным КА контроллеров с определенным значением точности. Восстановление качества данных происходит до того, как будет выработано управляющее воздействие $a_{i,t}$, т.е. согласно рис. 2, перед подачей наблюдения $o_{i,t}$ в актёр-критическую нейросеть конкретного агента. В частности, восстановление качества данных в искаженных наборах $o^*_{i,t}$ приводит к нивелированию несогласованности в мультиагентной системы, и как следствие, уменьшению колебаний (рис. 7в и 8в).

В табл. 2 показан пример одного из наблюдений $o_{i,t}$ атакованного агента-контроллера №3 для различных сценариев для определенного момента времени моделирования t. Хорошо видно, что при Hijacking-атаке наблюдение агента по всем параметрам грубо нарушено и не соответствует действительности, т.е. наблюдается полный захват этого контроллера. Однако применение метода k-ближайших соседей в рамках предложенной процедуры позволяет восстановить качество данных до значений в требуемых пределах.

Выводы

Рассмотрена мультиагентная система управления КФМС. Проведен аналитический обзор методов искусственного интеллекта и машинного обучения для обеспечения КБ КФМС при интеллектуальном управлении ими. Разработан интеллектуальный подход к обнаружению КА и повышению качества данных при вторичном мультиагентном управлении КФМС на основе алгоритмов машинного обучения. Эффективность применения предложенного подхода подтверждена численными расчетами на примере моделирования КА на взаимосвязанные КФМС, имеющих общую систему мультиагентного вторичного регулирования напряжения.

Работа выполнена в рамках научного проекта «Теоретические основы, модели и методы управления развитием и функционированием интеллектуальных электроэнергетических систем», № FWEU-2021-0001.

Литература

1. *N. Voropai. Electric Power System Transformations: A Review of Main Prospects and Challenges // Energies, 2020, vol. 13(21), 5639. DOI:10.3390/en13215639*
2. *Илюшин П. В. Системный подход к развитию и внедрению распределенной энергетики и возобновляемых источников энергии в России // Энергетик, 2022, 4, с. 20–27.*
3. *Nisha T. N., Pramod D. Sequential pattern analysis for event-based intrusion detection // International Journal of Information and Computer Security, 2019, 11(4/5), 476. DOI:10.1504/ijics.2019.101936*
4. *C. Li, M. Qiu. Reinforcement Learning for Cyber-Physical Systems: with Cybersecurity Case Studies. Chapman and Hall/CRC, 2019.*
5. *S. Gaba et al. A Systematic Analysis of Enhancing Cyber Security Using Deep Learning for Cyber Physical Systems // IEEE Access, 2024, vol. 12, pp. 6017–6035. DOI: 10.1109/ACCESS.2023.3349022*

6. *F. O. Olowononi, D. B. Rawat and C. Liu. Resilient Machine Learning for Networked Cyber Physical Systems: A Survey for Machine Learning Security to Securing Machine Learning for CPS // in IEEE Communications Surveys & Tutorials, Firstquarter 2021, vol. 23, no. 1, pp. 524–552. DOI: 10.1109/COMST.2020.3036778*
7. *Илюшин П. В., Вольный В. С. Обзор структур микросетей низкого напряжения с распределенными источниками энергии // Релейная защита и автоматизация. 2023, № 1(50), с. 68–80.*
8. *Mahela O. P., Khosravy M., Gupta N., et al. Comprehensive Overview of Multi-agent Systems for Controlling Smart Grids // CSEE Journal of Power and Energy Systems, 2022, Vol. 8, No. 1, pp. 115–131. DOI: 10.17775/CSEEJPES.2020.03390*
9. *Jabbar M. A. M., Tran D. T., Kim K. -H. Decentralized Power Flow Control Strategy Using Transition Operations of DC-Bus Voltage for Detection of Uncertain DC Microgrid Operations // Sustainability, 2023, Vol. 15, 11635. DOI: 10.3390/su151511635*
10. *Takayama S., Ishigame A. Volt–Var curve determination method of smart inverters by multi-agent deep reinforcement learning // International Journal of Electrical Power & Energy Systems, 2024, Vol. 157, 109888. DOI: 10.1016/j.ijepes.2024.109888*
11. *Tomin N., Voropai N., Kurbatsky V., Rehtanz C. Management of Voltage Flexibility from Inverter-Based Distributed Generation Using Multi-Agent Reinforcement Learning // Energies, 2021, 14(24), 8270. DOI: 10.3390/en14248270*
12. *Гурина Л. А. Оценка рисков кибербезопасности энергетического сообщества микросетей // Вопросы кибербезопасности. 2024. 1(59). С. 101–107. DOI: 10.21681/2311-3456-2024-1-101-107*
13. *H. Zhang, D. Yue, C. Dou and G. P. Hancke/ Resilient Optimal Defensive Strategy of Micro-Grids System via Distributed Deep Reinforcement Learning Approach Against FDI Attack // in IEEE Transactions on Neural Networks and Learning Systems, Jan. 2024, vol. 35, no. 1, pp. 598–608. DOI: 10.1109/TNNLS.2022.3175917*
14. *I. Tasevski and K. Jakimoski. Overview of SQL Injection Defense Mechanisms // 2020 28th Telecommunications Forum (TELFOR), Belgrade, Serbia, 2020, pp. 1–4. DOI: 10.1109/TELFOR51502.2020.9306676*
15. *B. Abazi and E. Hajrizi. Practical analysis on the algorithm of the Cross-Site Scripting Attacks // 2022 29th International Conference on Systems, Signals and Image Processing (IWSSIP), Sofia, Bulgaria, 2022, pp. 1–4. DOI: 10.1109/IWSSIP55020.2022.9854491*
16. *Mode, G. R.; Calyam, P.; Hoque, K. A. False data injection attacks in internet of things and deep learning enabled predictive analytics. arXiv 2019, arXiv:1910.01716.*
17. *Y. Gao, H. Hasegawa, Y. Yamaguchi and H. Shimada. Malware Detection by Control-Flow Graph Level Representation Learning With Graph Isomorphism Network // in IEEE Access, 2022, vol. 10, pp. 111830–11841. DOI: 10.1109/ACCESS.2022.3215267*
18. *T. Li, B. Chen, L. Yu and W. -A. Zhang. Active Security Control Approach Against DoS Attacks in Cyber-Physical Systems // in IEEE Transactions on Automatic Control, Sept. 2021, vol. 66, no. 9, pp. 4303–4310. DOI: 10.1109/TAC.2020.3032598*
19. *X. Xie, Y. Liu and B. Xu, Resilient event-triggered control for cyber-physical systems under stochastic-sampling and denial-of-service attacks // 2021 40th Chinese Control Conference (CCC), Shanghai, China, 2021, pp. 4702-4708. DOI: 10.23919/CCC52363.2021.9549917*
20. *A. Talati, V. Garg, N. Mishra, P. Tiwari and P. Jena. Cyber-Attack Detection in Smart Grids Using Machine Learning Approach // 2023 7th International Conference on Computer Applications in Electrical Engineering-Recent Advances (CERA), Roorkee, India, 2023, pp. 1–6. DOI: 10.1109/CERA59325.2023.10455586*
21. *X. Niu, J. Li, J. Sun and K. Tomsovic. Dynamic Detection of False Data Injection Attack in Smart Grid using Deep Learning // 2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA, 2019, pp. 1–6. DOI: 10.1109/ISGT.2019.8791598*
22. *S. Pusarla, U. Ghugar, T. Özseven, B. K. Dewangan, T. Choudhury and J. C. Patni. A Compressive Study on Detection Accuracy Model for DoS Attack in SDN Using Ensemble Learning Techniques // 2023 7th International Symposium on Innovative Approaches in Smart Technologies (ISAS), Istanbul, Turkiye, 2023, pp. 1-6. DOI: 10.1109/ISAS60782.2023.10391345*
23. *A. Srivastava, H. S. Sharma, R. Rawat and N. Garg. Detection of Cyber Attack in IoT Based Model Using ANN Model with Genetic Algorithm // 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT), Greater Noida, India, 2024, pp. 1198-1201. DOI: 10.1109/IC2PCT60090.2024.10486578*
24. *A. AlBusaidi and F. H. Mohideen. Analysis of Wireless Sensor Network Security Models: A Salient Approach for Deeper Inspection Using Deep Neural Networks // 2023 International Conference on Emerging Techniques in Computational Intelligence (ICETCI), Hyderabad, India, 2023, pp. 276–282. DOI: 10.1109/ICETCI58599.2023.10330927*
25. *S. Puneeth, S. Lal, M. Pratap Singh and B. S. Raghavendra. RMDNet-Deep Learning Paradigms for Effective Malware Detection and Classification // in IEEE Access, 2024, vol. 12, pp. 82622–82635, 2024. DOI: 10.1109/ACCESS.2024.3403458*
26. *P. S. Patil, S. L. Deshpande, G. S. Hukkeri, R. H. Goudar and P. Siddarkar. Prediction of DDoS Flooding Attack using Machine Learning Models // 2022 Third International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), Bengaluru, India, 2022, pp. 1–6. DOI: 10.1109/ICSTCEE56972.2022.10100083*
27. *S. Bala and S. M. M. Ahsan. Detecting DDoS attacks in Software Define Networking: A Machine Learning Based Approach // 2023 International Conference on Next-Generation Computing, IoT and Machine Learning (NCIM), Gazipur, Bangladesh, 2023, pp. 1–6. DOI: 10.1109/NCIM59001.2023.10212569*
28. *Sarker I. H. Machine Learning: Algorithms, Real-World Applications and Research Directions // SN Computer Science, 2021, 2(3). DOI:10.1007/s42979-021-00592-x*
29. *Ripan Rony, Md. Moinul Islam, Alqahtani Hamed, Sarker Iqbal H. Effectively predicting cyber-attacks through isolation forest learning-based outlier detection // Security and Privacy, 2022, 5(3). DOI:10.1002/spy2.212*
30. *Гурина Л. А., Томин Н. В. Разработка комплексного подхода к обеспечению кибербезопасности взаимосвязанных информационных систем при интеллектуальном управлении сообществом микросетей // Вопросы кибербезопасности, 2023, 4(56), с. 88–97. DOI:10.21681/2311-3456-2023-4-94-104*
31. *Hariri S., Kind M. C., Brunner R. J. Extended Isolation Forest // IEEE Transactions on Knowledge and Data Engineering, 2021, Vol. 33, No. 4, pp. 1479–1489. DOI: 10.1109/TKDE.2019.2947676*
32. *Murti D. M. P., Pujianto U., Wibawa A. P., Akbar M. I. K-Nearest Neighbor (K-NN) based Missing Data Imputation // 2019 5th International Conference on Science in Information Technology (ICSITech), Yogyakarta, Indonesia, 2019, pp. 83–88. DOI: 10.1109/ICSITech46713.2019.8987530*

33. *Staples L., Ring J., Fontana S., et al. Reproducible clustering with non-Euclidean distances: a simulation and case study // International Journal of Data Science and Analytics, 2023. DOI: 10.1007/s41060-023-00429-1*
34. *Deo T. Y., Sanju A. Data imputation and comparison of custom ensemble models with existing libraries like XGBoost, CATBoost, AdaBoost and Scikit learn for predictive equipment failure // Materials Today: Proceedings, 2023, Volume 72, Part 3, pp. 1596–1604. DOI: 10.1016/j.matpr.2022.09.410*
35. *Barillaro L. Deep Learning Platforms: PyTorch // Reference Module in Life Sciences, Elsevier, 2024. ISBN 9780128096338. DOI: 10.1016/B978-0-323-95502-7.00093-2.S*
36. *S. Mo, W. -H. Chen and X. Lu. Distributed hybrid secondary control strategy for DC microgrid group based on multi-agent system // 2021 33rd Chinese Control and Decision Conference (CCDC), Kunming, China, 2021, pp. 109–114. DOI: 10.1109/CCDC52312.2021.9602249*

INTELLIGENT METHODS OF ENSURING CYBERSECURITY MULTI-AGENT CONTROL SYSTEM OF MICROGRID

Gurina L. A.[3], Tomin N. V.[4]

The research aims *to develop methods for detecting and suppressing the consequences of cyber-attacks in secondary voltage regulation in multi-agent control systems of cyber-physical microgrids.*

The research relies *on the machine learning methods, probabilistic methods.*

Research result: *an isolation forest algorithm for automatic detection of cyber-attacks and an algorithm for data recovery based on the k-nearest neighbors method were developed.*

The scientific novelty *lies in the fact that the proposed method for detecting cyber-attacks and improving information quality creates opportunities for robustness, adaptation and recovery of multi-agent systems in case of cybersecurity breaches.*

Keywords: *cyber-physical microgrid, multi-agent system, intelligent control, identification of cyber-attacks, detection of bad data, improving information quality.*

References

1. *N. Voropai. Electric Power System Transformations: A Review of Main Prospects and Challenges // Energies, 2020, vol. 13(21), 5639. DOI:10.3390/en13215639*
2. *Ilyushin P. V. Sistemnyj podhod k razvitiyu i vnedreniyu raspredelennoj energetiki i vozobnovlyaemyh istochnikov energii v Rossii // Energetik, 2022, 4, s. 20–27.*
3. *Nisha T. N., Pramod D. Sequential pattern analysis for event-based intrusion detection // International Journal of Information and Computer Security, 2019, 11(4/5), 476. DOI:10.1504/ijics.2019.101936*
4. *C. Li, M. Qiu. Reinforcement Learning for Cyber-Physical Systems: with Cybersecurity Case Studies. Chapman and Hall/CRC, 2019.*
5. *S. Gaba et al. A Systematic Analysis of Enhancing Cyber Security Using Deep Learning for Cyber Physical Systems // IEEE Access, 2024,* vol. 12, pp. 6017–6035. DOI: 10.1109/ACCESS.2023.3349022
6. *F. O. Olowononi, D. B. Rawat and C. Liu. Resilient Machine Learning for Networked Cyber Physical Systems: A Survey for Machine Learning Security to Securing Machine Learning for CPS // in IEEE Communications Surveys & Tutorials, Firstquarter 2021, vol. 23, no. 1, pp. 524–552. DOI: 10.1109/COMST.2020.3036778*
7. *Ilyushin P. V., Vol'nyj V. S. Obzor struktur mikrosetej nizkogo napryazheniya s raspredelennymi istochnikami energii // Relejnaya zashchita i avtomatizaciya [Relay protection and automation]. 2023, № 1(50), s. 68–80.*
8. *Mahela O. P., Khosravy M., Gupta N., et al. Comprehensive Overview of Multi-agent Systems for Controlling Smart Grids // CSEE Journal of Power and Energy Systems, 2022, Vol. 8, No. 1, pp. 115-131. DOI: 10.17775/CSEEJPES.2020.03390*
9. *Jabbar M. A. M., Tran D. T., Kim K. -H. Decentralized Power Flow Control Strategy Using Transition Operations of DC-Bus Voltage for Detection of Uncertain DC Microgrid Operations // Sustainability, 2023, Vol. 15, 11635. DOI: 10.3390/su151511635*
10. *Takayama S., Ishigame A. Volt–Var curve determination method of smart inverters by multi-agent deep reinforcement learning // International Journal of Electrical Power & Energy Systems, 2024, Vol. 157, 109888. DOI: 10.1016/j.ijepes.2024.109888*
11. *Tomin N., Voropai N., Kurbatsky V., Rehtanz C. Management of Voltage Flexibility from Inverter-Based Distributed Generation Using Multi-Agent Reinforcement Learning // Energies, 2021, 14(24), 8270. DOI: 10.3390/en14248270*
12. *Gurina L. A. Ocenka riskov kiberbezopasnosti energeticheskogo soobshchestva mikrosetej // Voprosy kiberbezopasnosti [Cybersecurity issues], 2024, 1(59), s. 101–107. DOI: 10.21681/2311-3456-2024-1-101-107*
13. *H. Zhang, D. Yue, C. Dou and G. P. Hancke/ Resilient Optimal Defensive Strategy of Micro-Grids System via Distributed Deep Reinforcement Learning Approach Against FDI Attack // in IEEE Transactions on Neural Networks and Learning Systems, Jan. 2024, vol. 35, no. 1, pp. 598–608. DOI: 10.1109/TNNLS.2022.3175917*

3 Liudmila A. Gurina, Ph.D. in engineering, Associate Professor, Senior Research Fellow, Laboratory for Control of Electric Power Systems at Melentiev Energy Systems Institute, SB RAS, Irkutsk, Russia. E mail: gurina@isem.irk.ru

4 Nikita N. Tomin, Ph.D. in engineering, Head of Laboratory for Control of Electric Power Systems at Melentiev Energy Systems Institute, SB RAS, Irkutsk, Russia. E-mail: tomin.nv@gmail.com

14. *I. Tasevski and K. Jakimoski. Overview of SQL Injection Defense Mechanisms // 2020 28th Telecommunications Forum (TELFOR), Belgrade, Serbia, 2020, pp. 1-4. DOI: 10.1109/TELFOR51502.2020.9306676*
15. *B. Abazi and E. Hajrizi. Practical analysis on the algorithm of the Cross-Site Scripting Attacks // 2022 29th International Conference on Systems, Signals and Image Processing (IWSSIP), Sofia, Bulgaria, 2022, pp. 1-4. DOI: 10.1109/IWSSIP55020.2022.9854491*
16. *Mode, G. R.; Calyam, P.; Hoque, K. A. False data injection attacks in internet of things and deep learning enabled predictive analytics. arXiv 2019, arXiv:1910.01716.*
17. *Y. Gao, H. Hasegawa, Y. Yamaguchi and H. Shimada. Malware Detection by Control-Flow Graph Level Representation Learning With Graph Isomorphism Network // in IEEE Access, 2022, vol. 10, pp. 111830-111841. DOI: 10.1109/ACCESS.2022.3215267*
18. *T. Li, B. Chen, L. Yu and W. -A. Zhang. Active Security Control Approach Against DoS Attacks in Cyber-Physical Systems // in IEEE Transactions on Automatic Control, Sept. 2021, vol. 66, no. 9, pp. 4303-4310. DOI: 10.1109/TAC.2020.3032598*
19. *X. Xie, Y. Liu and B. Xu, Resilient event-triggered control for cyber-physical systems under stochastic-sampling and denial-of-service attacks // 2021 40th Chinese Control Conference (CCC), Shanghai, China, 2021, pp. 4702–4708. DOI: 10.23919/CCC52363.2021.9549917*
20. *A. Talati, V. Garg, N. Mishra, P. Tiwari and P. Jena. Cyber-Attack Detection in Smart Grids Using Machine Learning Approach // 2023 7th International Conference on Computer Applications in Electrical Engineering-Recent Advances (CERA), Roorkee, India, 2023, pp. 1–6. DOI: 10.1109/CERA59325.2023.10455586*
21. *X. Niu, J. Li, J. Sun and K. Tomsovic. Dynamic Detection of False Data Injection Attack in Smart Grid using Deep Learning // 2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA, 2019, pp. 1–6. DOI: 10.1109/ISGT.2019.8791598*
22. *S. Pusarla, U. Ghugar, T. Özseven, B. K. Dewangan, T. Choudhury and J. C. Patni. A Compressive Study on Detection Accuracy Model for DoS Attack in SDN Using Ensemble Learning Techniques // 2023 7th International Symposium on Innovative Approaches in Smart Technologies (ISAS), Istanbul, Turkiye, 2023, pp. 1–6. DOI: 10.1109/ISAS60782.2023.10391345*
23. *A. Srivastava, H. S. Sharma, R. Rawat and N. Garg. Detection of Cyber Attack in IoT Based Model Using ANN Model with Genetic Algorithm // 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT), Greater Noida, India, 2024, pp. 1198–1201. DOI: 10.1109/IC2PCT60090.2024.10486578*
24. *A. AlBusaidi and F. H. Mohideen. Analysis of Wireless Sensor Network Security Models: A Salient Approach for Deeper Inspection Using Deep Neural Networks // 2023 International Conference on Emerging Techniques in Computational Intelligence (ICETCI), Hyderabad, India, 2023, pp. 276–282. DOI: 10.1109/ICETCI58599.2023.10330927*
25. *S. Puneeth, S. Lal, M. Pratap Singh and B. S. Raghavendra. RMDNet-Deep Learning Paradigms for Effective Malware Detection and Classification // in IEEE Access, 2024, vol. 12, pp. 82622-82635, 2024. DOI: 10.1109/ACCESS.2024.3403458*
26. *P. S. Patil, S. L. Deshpande, G. S. Hukkeri, R. H. Goudar and P. Siddarkar. Prediction of DDoS Flooding Attack using Machine Learning Models // 2022 Third International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), Bengaluru, India, 2022, pp. 1–6. DOI: 10.1109/ICSTCEE56972.2022.10100083*
27. *S. Bala and S. M. M. Ahsan. Detecting DDoS attacks in Software Define Networking: A Machine Learning Based Approach // 2023 International Conference on Next-Generation Computing, IoT and Machine Learning (NCIM), Gazipur, Bangladesh, 2023, pp. 1–6. DOI: 10.1109/NCIM59001.2023.10212569*
28. *Sarker I. H. Machine Learning: Algorithms, Real-World Applications and Research Directions // SN Computer Science, 2021, 2(3). DOI:10.1007/s42979-021-00592-x*
29. *Ripan Rony, Md. Moinul Islam, Alqahtani Hamed, Sarker Iqbal H. Effectively predicting cyber-attacks through isolation forest learning-based outlier detection // Security and Privacy, 2022, 5(3). DOI:10.1002/spy2.212*
30. *Gurina L. A., Tomin N. V. Razrabotka kompleksnogo podhoda k obespecheniyu kiberbezopasnosti vzaimosvyazannyh informacionnyh sistem pri intellektual'nom upravlenii soobshchestvom mikrosetej // Voprosy kiberbezopasnosti [Cybersecurity issues], 2023, 4(56), s. 88–97. DOI:10.21681/2311-3456-2023-4-94-104*
31. *Hariri S., Kind M. C., Brunner R. J. Extended Isolation Forest // IEEE Transactions on Knowledge and Data Engineering, 2021, Vol. 33, No. 4, pp. 1479–1489. DOI: 10.1109/TKDE.2019.2947676*
32. *Murti D. M. P., Pujianto U., Wibawa A. P., Akbar M. I. K-Nearest Neighbor (K-NN) based Missing Data Imputation // 2019 5th International Conference on Science in Information Technology (ICSITech), Yogyakarta, Indonesia, 2019, pp. 83-88. DOI: 10.1109/ICSITech46713.2019.8987530*
33. *Staples L., Ring J., Fontana S., et al. Reproducible clustering with non-Euclidean distances: a simulation and case study // International Journal of Data Science and Analytics, 2023. DOI: 10.1007/s41060-023-00429-1*
34. *Deo T. Y., Sanju A. Data imputation and comparison of custom ensemble models with existing libraries like XGBoost, CATBoost, AdaBoost and Scikit learn for predictive equipment failure // Materials Today: Proceedings, 2023, Volume 72, Part 3, pp. 1596–1604. DOI: 10.1016/j.matpr.2022.09.410*
35. *Barillaro L. Deep Learning Platforms: PyTorch // Reference Module in Life Sciences, Elsevier, 2024. ISBN 9780128096338. DOI: 10.1016/B978-0-323-95502-7.00093-2.S*
36. *S. Mo, W. -H. Chen and X. Lu. Distributed hybrid secondary control strategy for DC microgrid group based on multi-agent system // 2021 33rd Chinese Control and Decision Conference (CCDC), Kunming, China, 2021, pp. 109-114. DOI: 10.1109/CCDC52312.2021.9602249*

ОБНАРУЖЕНИЕ ОБФУСЦИРОВАННЫХ ЭКСПЛОИТОВ В ФАЙЛАХ НЕИСПОЛНЯЕМЫХ ФОРМАТОВ

Архипов А. Н.[1], Кондаков С. Е.[2]

DOI: 10.21681/2311-3456-2024-6-65-75

Цель исследования: *разработка модели бинарной классификации файлов неисполняемых форматов, обеспечивающей повышение эффективности выявления обфусцированных эксплоитов относительно моделей, реализованных в существующих средствах антивирусной защиты.*

Методы исследования *базируются на положениях теории вероятности и математической статистики, теории множеств, методов проведения натурного эксперимента и обработки экспериментальных данных.*

Результат: *в ходе исследования на базе математической модели эксплоита сгенерировано множество потенциальных признаков, которые представлены численными значениями. Из сформированного признакового пространства осуществлен отбор информативных признаков и построение модели бинарной классификации, обладающей наилучшими показателями эффективности в выявлении обфусцированных эксплоитов. Разработана программа для ЭВМ, реализующая полученную модель. Эффективность разработанной модели подтверждена в рамках проведенных экспериментальных исследований по оценке эффективности выявления обфусцированных эксплоитов с использованием средств антивирусной защиты, включенных в реестр российского программного обеспечения, и средств антивирусной защиты иностранного производства, размещенных в свободном доступе, и авторской программы.*

Научная новизна *результатов определяется совокупностью авторских процедур, обеспечивающих выбор классификатора, его гиперпараметров, а также формирование информативного признакового пространства, включая признаки, разработанные авторами, и, позволяющих построить наиболее эффективную модель бинарной классификации, что обеспечивает обоснованность полученных результатов. Представлено подтверждение реализуемости и получения лучших значений показателей эффективности при выявлении обфусцированных эксплоитов относительно существующих средств антивирусной защиты.*

Практическая значимость: *представленная модель, в первую очередь, ориентирована на применение в системах антивирусной защиты, но может быть использована и для решения других задач обеспечения информационной безопасности.*

Ключевые слова: *кибербезопасность, компьютерные атаки, вредоносный код, защита информации, системы антивирусной защиты информации, система обнаружения вторжений.*

Введение

Проблема выявления угроз нарушения информационной безопасности, реализуемых посредством локальных эксплоитов, распространяемых в файлах неисполняемых форматов и позволяющих за счет эксплуатации уязвимостей исполнять произвольный код (далее – эксплоиты), является одной из актуальных и недостаточно решенных вопросов обеспечения информационной безопасности [1, 2].

Для решения указанной проблемы уже несколько десятилетий применяются средства антивирусной защиты информации.

Реализованные в данных средствах защиты информации модели и алгоритмы обнаружения вредоносного кода, главным образом, базируются на устойчивых статических и поведенческих паттернах, выделенных из уже выявленных и изученных образцов эксплоитов.

На практике указанный подход показывает хорошие результаты при обнаружении только известных эксплоитов и одновременно плохо справляется с выявлением эксплоитов, созданных на базе известных уязвимостей с применением технологий обфускации и уязвимостей нулевого дня.

В данной статье будет рассмотрено только обнаружение обфусцированных эксплоитов, так как данный инструмент проведения компьютерных атак значительно чаще встречается на практике [3, 4].

Общий алгоритм действий злоумышленника при распространении вредоносного программного обеспечения в форме обфусцированных эксплоитов представлен на рисунке (рис. 1).

Процедура обфускации позволяет значительно снизить эффективность обнаружения вредоносного кода применяемыми средствами антивирусной защиты в файлах неисполняемых форматов при проведении компьютерных атак, что подтверждают соответствующие экспериментальные исследования [5].

Одновременно файлы неисполняемого формата не вызывают подозрения у пользователей и в сочетании с методами социальной инженерии

1 Архипов Александр Николаевич, студент МГТУ им. Баумана, г. Москва, Россия. E-mail: diskpart111@mail.ru
2 Кондаков Сергей Евгеньевич, кандидат технических наук, доцент, МГТУ им. Баумана, г. Москва, Россия. E-mail: sergeikondakov@list.ru

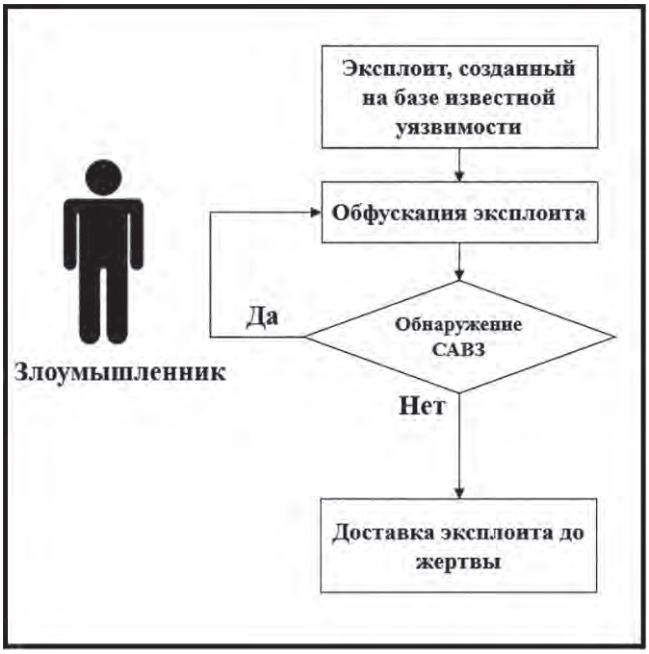


Рис. 1. Процесс подготовки эксплоита для обхода средств антивирусной защиты (САВЗ)

позволяют атакующему распространять вредоносное программное обеспечение под видом легитимного посредством электронной почты, социальных сетей и различных мессенджеров.

Таким образом актуализируется задача совершенствования научно-методического аппарата обнаружения обфусцированных эксплоитов особенно в условиях открытого распространения техник и технологий обфускации в сети Интернет, включая программное обеспечение для автоматизации данного процесса.

Обзор релевантных работ

С формальной точки зрения указанная проблема обнаружения эксплоитов сводится к решению научной задачи бинарной классификации файлов неисполняемых форматов на «безопасные» и «вредоносные».

Эффективное решение указанной задачи классификации зависит от качества используемого признакового пространства, описывающего объект классификации, и классификатора, которые в совокупности формируют модель бинарной классификации.

С учетом изложенного дальнейшее рассмотрение моделей бинарной классификации, предлагаемых в научной и практической литературе, будем осуществлять с точки зрения подходов к формированию признакового пространства и выбора классификатора.

Проведем анализ работ, в которых представлены модели бинарной классификации, разработанные для обнаружения вредоносного программного обеспечения, распространяемого в форме эксплоитов в файлах неисполняемых форматов.

В работах [6–8] признаковое пространство формируется по результатам поиска и анализа в исследуемых файлах модулей, которые потенциально могут содержать исполняемый код: макросы VBA, OLE-объекты, вставки кода JavaScript, которые являются легитимными и определены форматом. При этом предметом поиска являются устойчивые потенциально опасные конструкции, ранее выявленные в других вредоносных файлах.

В качестве классификаторов отобраны классические решения такого рода задач алгоритмы: деревья решений, метод опорных векторов, наивный байесовский классификатор, метод k-ближайших соседей, а также их комбинация: объединение, пересечение или выбор одного из перечисленных.

В публикациях [9, 10] признаковое пространство формируется посредством извлечения типовых последовательностей различных системных вызовов API, которые формируются на базе моделей обработки естественного языка (NLP).

В качестве классификаторов предложено использовать градиентный бустинг деревьев решений, реализованный в библиотеках CatBoost, XGBoost, а также метод случайного леса.

В работе [11] предлагается построение классификатора на основе полученных авторами 89 реляционных правил из моделей, основанных на правилах PART, OneR и JRip.

В статье [12] предложены методы, представляющие исследуемый файл в виде графического изображения с последующей его классификацией с использованием сети с долговременной и кратковременной памятью (LSTM).

Вышеуказанные модели бинарной классификации имеют ряд общих недостатков, которые на практике могут значительно снизить эффективность обнаружения эксплоитов:

1. Каждая из рассмотренных моделей предназначена для обнаружения эксплоитов только в конкретном формате и непригодна для анализа других файлов неисполняемых форматов.
2. При построении представленных моделей не учитываются возможности атакующего по применению технологий обфускации вредоносного кода.
3. Генерация признакового пространства в упомянутых работах осуществляется экспертом (группой экспертов) и не обосновывается математически, например, за счет математического моделирования эксплоита.
4. Разработанные модели не учитывают иные возможные места внедрения вредоносного кода кроме структурных элементов файла, предусмотренных спецификацией на формат для размещения исполняемого кода.

Вышеуказанные факторы актуализируют задачу разработки новых моделей бинарной классификации файлов неисполняемых форматов в задаче обнаружения эксплойтов, потому их разработка формирует результат, обладающий научной новизной.

В настоящей работе предлагается авторская модель бинарной классификации файлов неисполняемых форматов на основе созданного вектора информативных признаков, учитывающая указанные недостатки.

Цель (постановка задачи) исследования

Содержательная (вербальная) постановка научной задачи.

Разработать модель бинарной классификации файлов неисполняемых форматов, обеспечивающую повышение эффективности выявления обфусцированных эксплоитов, относительно моделей, реализованных в существующих средствах антивирусной защиты.

Формальная постановка научной задачи.

Дано:

X – множество файлов неисполняемых форматов, с внедренными обфусцированными эксплоитами (вредоносные файлы) и без таковых (безопасные файлы); PF – множество потенциальных признаков-кандидатов; O – множество рассматриваемых обфускаторов; SZI – множество средств антивирусной защиты; Z – множество рассматриваемых классификаторов с дискретным набором гиперпараметров.

Задача: разработать модель бинарной классификации R, обеспечивающую максимальный показатель эффективности обнаружения обфусцированных эксплоитов q, и его улучшение относительно рассматриваемых классификаторов и существующих средств антивирусной защиты:

$$R: argmax\ q[Z^*(O(X),PF^*)] > \\ > q[SZI(O(X))],|PF^*| \le |PF|, \quad (1)$$

где q – повышаемый показатель эффективности; Z^* – выбранный классификаторов с дискретным набором гиперпараметров; PF^* – множество отобранных информативных признаков.

В качестве показателя эффективности бинарной классификации определим следующий [13]:

$$q = \frac{Se + Sp}{2}, \quad (2)$$

где Se – чувствительность, под которой понимается доля истинно положительных случаев классификации, которая вычисляется по формуле:

$$Se = \frac{TP}{TP + FN}, \quad (3)$$

где TP – число верно классифицированных «вредоносных» объектов, FN – число объектов, классифицированных как отрицательные (ошибка I рода).

При этом специфичность Sp определяется как доля истинно отрицательных случаев классификации, которые были корректно идентифицированы классификатором:

$$Sp = \frac{TN}{TN + FP}, \quad (4)$$

где TN – число верно классифицированных «безопасных» объектов, FP – число объектов, классифицированных как положительные (ошибка II рода).

Структура исследования

Процедура разработки модели бинарной классификации для обнаружения обфусцированных эксплоитов

Рис. 2. Схема подготовки исходных данных

в файлах неисполняемого формата декомпозирована на следующие этапы:

1. Подготовка исходных данных;
2. Определение минимального размера выборки;
3. Формирование множества потенциальных признаков-кандидатов;
4. Формированные и отбор моделей бинарной классификации;
5. Выбор наилучшей модели бинарной классификации;

Подготовка исходных данных

Подготовка необходимых для проведения исследования наборов «вредоносных» и «безопасных» файлов осуществлялось в порядке, представленном на схеме (рис. 2).

Формат файлов Office Open XML (расширения: .docx, .xlsx, .pptx, .vsdx) был выбран ввиду его наибольшей распространенности при проведении компьютерных атак и соответственно наличия значительно большего числа образцов с внедренными эксплоитами в открытом доступе.

Рис. 3. Процедура получения эмпирического закона вероятности обнаружения обфусцированных эксплоитов

С учетом изложенного на данном этапе получены три выборки, включающие в себя на момент проведения исследования доступные образцы эксплоитов, содержащие известные разновидности уязвимостей.

Определение минимального размера выборки

Определение минимального размера выборки осуществлялась в следующем порядке:

1. Получение эмпирического закона распределения вероятности обнаружения обфусцированных эксплоитов.
2. Проверка статистической гипотезы о нормальности эмпирического закона распределения.
3. Расчет минимального размера выборки с уровнем значимости равным 0,01.

Учитывая, что на момент проведения настоящего исследования опубликовано 223 подтвержденных уязвимости, позволяющих выполнить произвольный код в файлах неисполняемого формата Office Open XML, процедура получения эмпирического закона распределения вероятности обнаружения обфусцированных эксплоитов реализована следующим образом (рис. 3).

При этом отбор из исходной выборки образцов, содержащих отличные уязвимости, осуществлялся методом бутстрэпа[3] [14].

По результатам выполнения первого этапа получено следующее распределение (рис. 4), характеризующее процесс выявления угроз нарушения информационной безопасности, представленных эксплоитами, подвергшихся процедуре обфускации.

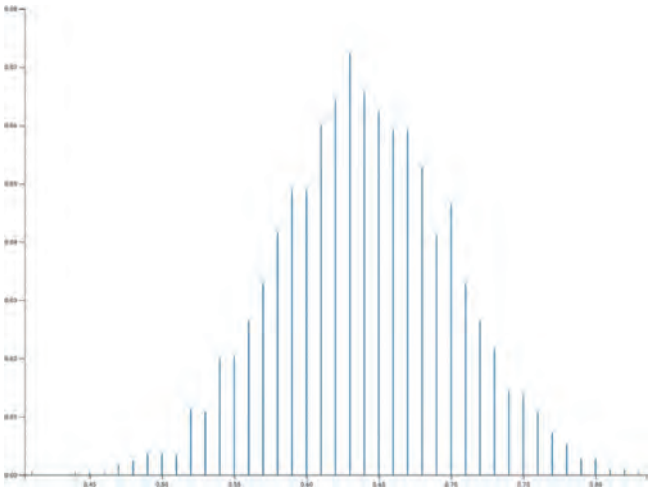


Рис. 4. Распределение, характеризующее процесс обнаружения обфусцированных эксплоитов

3 Бутстрэп (англ. bootstrap) в статистике — практический компьютерный метод исследования распределения статистик вероятностных распределений, основанный на многократной генерации выборок методом Монте-Карло на базе имеющейся выборки.

На следующем этапе для проверки гипотезы о нормальности полученного распределения использовались специализированные статистические тесты: Колмогорова–Смирнова[4] и Шапиро–Уилка[5].

По результатам выполнения указанных статистических тестов получены следующие значения контрольных критериев: D = 0.012, W = 0,99, на основании которых можно сделать вывод о том, что полученное распределение являются нормальным с уровнем значимости равным 0.05.

С учетом подтверждения гипотезы о нормальности эмпирического закона вероятности обнаружения обфусцированных эксплоитов расчет минимального размера выборки n осуществлен по следующей формуле с учетом неизвестной численности генеральной совокупности [15]:

$$n = \frac{Z^2\,\sigma^2}{\delta^2} = \frac{2^2 0.12^2}{0.01^2} = 576, \qquad (5)$$

где Z – критическое значение нормального распределения, σ – стандартное отклонение, δ – уровень значимости.

Формирование множества потенциальных признаков-кандидатов

Формирование множества потенциальных признаков-кандидатов осуществлено на базе разработанной авторами математической модели эксплоита внедренного в файл неисполняемого формата [16].

С учетом изложенного сформировано два множества потенциальных признаков-кандидатов, предназначенных для обнаружения: модуля эксплуатации уязвимости и полезной нагрузки эксплоита.

Получение численных значений признаков осуществлялась путем применения различных математических методов.

Указанные методы применялись не ко всему содержимому исследуемых файлов неисполняемого формата, а только к содержимому, входящему в состав сегментов, полученных по результатам авторского алгоритма сегментации [17].

В качестве математических преобразований применялись методы, используемые в математической статистике и теории информации, а также ряд авторских процедур, которые приведены в таблицах (табл. 1, табл. 2).

Формирование и отбор моделей бинарной классификации

Формирование и отбор моделей бинарной классификации осуществлялся по следующей схеме (рис. 5) на основе обучающей выборки и показателя эффективности бинарной классификации q (2).

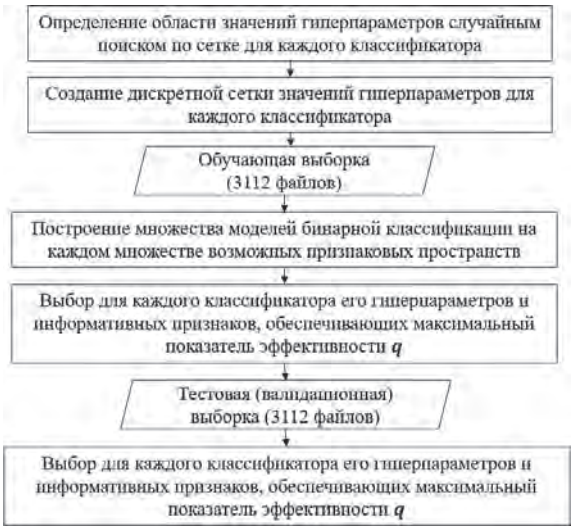


Рис. 5. Схема формирования и отбора моделей бинарной классификации

Таблица 1.
Множество признаков-кандидатов для обнаружения модуля эксплуатации уязвимости эксплоита

Методы получения числовых значений признака	Принадлежность
Форматный	Авторский
Математическое ожидание	Мат. статистика
Дисперсия	Мат. статистика
Мода	Мат. статистика
Медиана	Мат. статистика
Среднее линейное отклонение	Мат. статистика
Среднеквадратичное отклонение	Мат. статистика
Коэффициент асимметрии	Мат. статистика
Коэффициент эксцесса	Мат. статистика
Максимальное значение	Мат. статистика
Минимальное значение	Мат. статистика
Размах вариации	Мат. статистика
Базисный абсолютный прирост	Мат. статистика
Цепной абсолютный прирост	Мат. статистика
Коэффициент осцилляции	Мат. статистика
Относительное линейное отклонение	Мат. статистика
Относительный показатель квартильной вариации	Мат. статистика
Коэффициент вариации	Мат. статистика
Эмпирический коэффициент детерминации	Мат. статистика
Децильный коэффициент дифференциации	Мат. статистика
Тест Колмогорова-Смирнова	Мат. статистика
Тест Шапиро-Уилка	Мат. статистика
Энтропия	Теория информации

4 Леман Э. Проверка статистических гипотез; [пер. с англ.]. М.: Наука, 1979. 408 с.

5 Shapiro S. S., Wilk M. B. An analysis of variance test for normality (complete samples). Biometrika, 1965, vol. 52, no. 3-4, pp. 591–611. DOI: 10.1093/BIOMET/52.3-4.591.

Таблица 2.

Множество признаков-кандидатов для обнаружения полезной нагрузки эксплоита

Методы получения числовых значений признака	Принадлежность
Форматный	Авторский
Признак наличия машинного кода	Авторский
Признак наличия программного кода интерпретируемых языков программирования	Авторский
Признак наличия программного кода встроенных средств командного процессора	Авторский
Признак наличия программного кода встроенных средств программирования	Авторский
Показатель числа управляющих символов в строке	Авторский
Показатель числа специальных символов в строке	Авторский
Математическое ожидание	Мат. статистика
Дисперсия	Мат. статистика
Среднее линейное отклонение	Мат. статистика
Среднеквадратичное отклонение	Мат. статистика
Максимальное значение	Мат. статистика
Минимальное значение	Мат. статистика
Размах вариации	Мат. статистика
Коэффициент вариации	Мат. статистика
Энтропия	Теория информации
2-граммная энтропия	Теория информации
3-граммная энтропия	Теория информации
Эмпирический коэффициент детерминации	Мат. статистика
Децильный коэффициент дифференциации	Мат. статистика
Тест Колмогорова-Смирнова	Мат. статистика
Тест Шапиро-Уилка	Мат. статистика
Энтропия	Теория информации

По результатам выполнения вышеуказанных процедур отобраны сформированные модели бинарной классификации (табл.3), имеющие наилучшие значения показателя эффективности (2) для каждого из классификаторов, с учетом его гиперпараметров и признакового пространства.

Таблица 3.

Отобранные модели бинарной классификации

Метод классификации	Число признаков	Значения критерия q
Алгоритмы на основе деревьев решений		
Decision Tree	31	0.77
Extra Tree	29	0.73
Ансамблевые алгоритмы		
AdaBoost	22	0.90
Extra Trees	28	0.89
Random Forest	28	0.87
Bagging	23	0.85
Gradient Boosting	21	0.95
Hist Gradient Boosting	21	0.94
CatBoost	21	0.96
LightGBM	23	0.95
XGBoost	21	0.99
Линейные классификаторы		
Gaussian Naive Bayes	32	0.66
Multinomial Naive Bayes	29	0.65
Complement Naive Bayes	30	0.63
Bernoulli Naive Bayes	27	0.59
Ridge Classification	29	0.71
Stochastic Gradient Descent	26	0.73
Logistic Regression	28	0.69
Passive Aggressive Classifier	31	0.59
Метрические классификаторы		
K-nearest neighbors	25	0.88
Nearest Centroid	25	0.83
Метод опорных векторов		
SVM	28	0.84
Искусственные нейронные сети		
Perceptron	24	0.87
Multi-layer Perceptron	28	0.89

Выбор наилучшей модели бинарной классификации

Выбор наилучшей модели бинарной классификации осуществлялся по следующей схеме (рис. 6) на основе валидационной выборки и показателя эффективности бинарной классификации q (2).

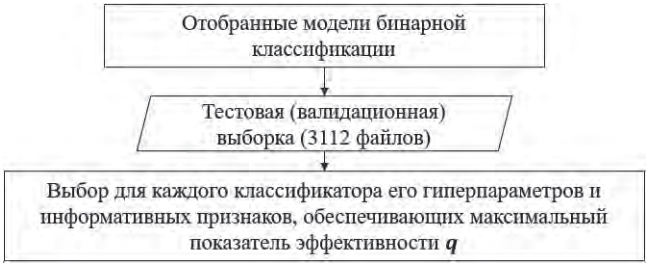


Рис. 6. Схема выбора наилучшей модели бинарной классификации

По результатам выполнения вышеуказанных процедур получены следующие значения показателя эффективности бинарной классификации q (2) для каждой из сформированных моделей бинарной классификации (табл. 4).

Таблица 4.
Множество признаков-кандидатов для обнаружения модуля эксплуатации уязвимости эксплоита

Метод классификации	Значения критерия q
Decision Tree	0.73
Extra Tree	0.72
AdaBoost	0.87
Extra Trees	0.86
Random Forest	0.83
Bagging	0.76
Gradient Boosting	0.91
Hist Gradient Boosting	0.92
CatBoost	0.93
LightGBM	0.93
XGBoost	0.98
Gaussian Naive Bayes	0.65
Multinomial Naive Bayes	0.65
Complement Naive Bayes	0.58
Bernoulli Naive Bayes	0.57
Ridge Classification	0.69
Stochastic Gradient Descent	0.72
Logistic Regression	0.67
Passive Aggressive Classifier	0.58
K-nearest neighbors	0.84
Nearest Centroid	0.80
SVM	0.83
Perceptron	0.83
Multi-layer Perceptron	0.87

С учетом полученных результатов в качестве наилучшей модели бинарной классификации выбрана модель, построенная на базе классификатора XGBoost[6] и двух конечных множеств информативных признаков $P'_M = \{F'_1,F'_2,\ldots,F'_{11}\}$ и $P'_S = \{G'_1,G'_2,\ldots,G'_{10}\}$, где:

1. P'_M – признаковое пространство для обнаружения модуля эксплуатации уязвимости эксплоита, P'_S – признаковое пространство для обнаружения модуля полезной нагрузки эксплоита.
2. F'_1 – математические ожидание, которое вычисляется по формуле:
$$F'_1 = \overline{x} = \sum_{i=1}^{256} b_i\, p_i, \quad (6)$$
где b_i – значение байта, p_i – вероятность появления b_i.
3. F'_2 – дисперсия, которая вычисляется по формуле:
$$F'_2 = D = \sum_{i=1}^{256} (b_i - \overline{x})\, p_i, \quad (7)$$
4. F'_3 – среднеквадратичное отклонение, которое вычисляется по формуле:
$$F'_3 = \sigma = \sqrt{D} \quad (8)$$
5. F'_4 – коэффициент асимметрии, который вычисляется по формуле:
$$F'_4 = A_3 = \frac{m_3}{\sigma_3}, \quad (9)$$
где m_3 – центральный эмпирический момент третьего порядка.
6. F'_5 – коэффициент эксцесса, который вычисляется по формуле:
$$F'_5 = E = \frac{m_4}{\sigma_4}, \quad (10)$$
где m_4 – центральный эмпирический момент четвертого порядка.
7. F'_6 – максимальное значение, которое вычисляется по формуле:
$$F'_6 = max(p_i) \quad (11)$$
8. F'_7 – минимальное значение, которое вычисляется по формуле:
$$F'_7 = min(p_i) \quad (12)$$
9. F'_8 – размах вариации, который вычисляется по формуле:
$$F'_8 = max(p_i) - min(p_i) \quad (13)$$
10. F'_9 – коэффициент вариации, который вычисляется по формуле:
$$F'_9 = \frac{\sigma * 100}{\overline{x}} \quad (14)$$
11. F'_{10} – энтропия, которая вычисляется по формуле:
$$F'_{10} = -\sum_{i=1}^{32640} p_i \log_2 p_i \quad (15)$$

6 XGBoost. – URL: https://github.com/dmlc/xgboost/ (дата обращения: 01.07.2024).

12. F'_{11} – форматный признак, который вычисляется по формуле:

$$F'_{11} = \begin{cases} 1, \text{ если все } B \text{ соотвествуют формату} \\ 0, \text{ если хотя бы один } B \text{ не соотвествует формату} \end{cases}, \quad (16)$$

где B – сегмент, полученный по результатам алгоритма сегментации, для обнаружения модуля эксплуатации уязвимости эксплоита.

13. G'_1 – форматный признак, который вычисляется по формуле:

$$G'_1 = \begin{cases} 1, \text{ если все } B' \text{ соотвествуют формату} \\ 0, \text{ если хотя бы один } B' \text{ не соотвествует формату} \end{cases}, \quad (17)$$

где B' – сегмент, полученный по результатам алгоритма сегментации для обнаружения модуля полезной нагрузки эксплоита.

14. G'_2 – признак наличия машинного кода, G'_3 – признак наличия программного кода интерпретируемых языков программирования, G'_4 – признак наличия кода встроенных средств командного процессора, G'_5 – признак наличия кода встроенных средств программирования, которые вычисляется по общей формуле:

$$G'_i(B') = \frac{G''_i}{|B'|}, \begin{cases} G''_i = G'_i + 1, \text{ если } d_{G'_i} \in B' \\ G''_i = G'_i + 0, \text{ если } d_{G'_i} \notin B' \end{cases}, \quad (18)$$

где $d_{G'_i}$ – команда (инструкция) множества программного кода, $i = 2...5$, $|B'|$ – размер сегмента в байтах, G''_i – общее число найденных $d_{G'_i}$ в сегменте B'.

15. G'_6 – показатель числа управляющих символов в строке, который вычисляется по формуле:

$$G'_6 = \frac{N_s}{|B'|}, \quad (19)$$

где N_s – количество специальных символов в сегменте B'.

16. G'_7 – показатель числа специальных символов в строке, который вычисляется по формуле:

$$G'_7 = \frac{N'_s}{|B'|}, \quad (20)$$

где N'_s – количество управляющих символов в сегменте B'.

17. G'_8 – энтропия, которая вычисляется по формуле:

$$G'_8 = -\sum_{i=1}^{256} p_i \log_2 p_i \quad (21)$$

18. G'_9 – 2-граммная энтропия, которая вычисляется по формуле:

$$G'_9 = -\sum_{j=1}^{32640} p_j \log_2 p_j \quad (22)$$

19. G'_{10} – 3-граммная энтропия, которая вычисляется по формуле:

$$G'_9 = -\sum_{k=1}^{2763520} p_k \log_2 p_k \quad (23)$$

Оценка эффективности обнаружения обфусцированных эксплоитов за счет применения разработанной модели

В целях оценки эффективности обнаружения обфусцированных эксплоитов за счет применения предложенной модели бинарной классификации проведены натурные экспериментальные исследования.

Предложенная модель реализована в программе для ЭВМ AntigenExploits (Свидетельство о госрегистрации № 2023687464).

Исследования эффективности проведены с использованием указанной программы.

Экспериментальные исследования проводились в следующем порядке:

1. Подготовлена тестовая выборка из 3200 файлов неисполняемых форматов, не используемых при построении модели бинарной классификации, содержащей файлы с внедренными эксплоитами (вредоносные файлы – 1600 образцов), и без таковых (безопасные файлы – 1600 образцов), сгенерированных в автоматическом режиме с использованием штатных шаблонов приложений Microsoft Office.

Для подтверждения наличия/отсутствия вредоносного кода в сформированной выборке применялись средства (технологии) антивирусной защиты информации, размещенные в открытом доступе [https://virustotal.com/].

Вредоносные файлы, входящие в выборку, прошли процедуру обфускации с использованием свободно доступных программных средств [https://github.com/], реализующих обфускацию программных кодов в автоматизированном режиме.

Таблица 5.

Результаты экспериментального исследования

Средства антивирусной защиты	Значения критерия q
AntigenExploits (СГР № 2023687464)	**0,99**
Kaspersky Endpoint Security	0,88
Dr.Web Enterprise Security Suite	0,74
NANO Антивирус Pro	0,67
Антивирус «VR Protect» для Linux	0,69
Avast	0,72
ClamAV	0,54
AVG	0,63
Symantec	0,84
Microsoft	0,69
McAfee Scanner	0,85
Panda	0,75

2. Проведен анализ тестовой выборки с использованием средств антивирусной защиты, включенных в реестр российского программного обеспечения, и средств антивирусной защиты иностранного производства, размещенных в свободном доступе, и авторской программы.
3. Проведен анализ результатов экспериментального исследования.

Результаты исследования представлены в таблице (таб. 5).

Результаты экспериментального исследования показали, что предложенная модель бинарной классификации в конкретном исследовании позволила повысить эффективность обнаружения обфусцированных эксплоитов относительно существующих средств антивирусной защиты в среднем на 26 %.

Заключение

В статье представлена модель бинарной классификации файлов неисполняемых форматов, применимая в задаче обнаружения эксплоитов, определяющая используемый классификатор, его гиперпараметры, а также множество информативных признаков.

Модель является универсальной и позволяет производить классификацию файлов неисполняемых форматов на «вредоносные» и «безопасные», обеспечивающую высокие значения выбранного показателя эффективности обнаружения относительно существующих средств антивирусной защиты. При этом предложенная модель бинарной классификации файлов неисполняемых форматов ориентирована, в первую очередь, на обнаружение эксплоитов, подвергшихся процедуре обфускации.

Указанные аспекты отличают представленную модель от релевантных аналогов.

В качестве направления для дальнейших исследований в данной области целесообразно рассмотреть возможность решения задачи оптимизации подбора гиперепараметров для классификатора, используемого в предложенной модели.

Литература

1. *Середкин С. П. Особенности кибератак на объекты критической информационной инфраструктуры в современных условиях // Информационные технологии и математическое моделирование в управлении сложными системами. – 2022. – № 4 (16). – С. 56–66. DOI: 10.26731/2658-3704.2022.4(16).56-66.*
2. *Ланецкая А. Ю., Александрова Е. Н. Современные угрозы информационной безопасности // Международный журнал гуманитарных и естественных наук. – 2022. – Том 7–2. – № 20. – С. 192–195. DOI:10.24412/2500-1000-2022-7-2-192-195.*
3. *Павлычев А. В., Стародубов М. И., Галимов А. Д. Использование алгоритма машинного обучения Random Forest для выявления сложных компьютерных инцидентов // Вопросы кибербезопасности. – 2022. – Том 51. – № 5. – С. 74–81. DOI:10.21681/2311-3456-2022-5-74-81.*
4. *Таловерова Д. В. Сравнительный анализ сценариев реализации угроз безопасности информации методики ФСТЭК РФ и Mitre Att&ck и их применение на практике // Фундаментальные и прикладные аспекты компьютерных технологий и информационной безопасности. Сборник статей Всероссийской научно-технической конференции. Таганрог, 2023. – С. 34–37.*
5. *Архипов А. Н., В. А. Пиков, В. В. Кабаков Порядок и результаты экспериментальных исследований влияния обфускации на качество выявления угроз информационной безопасности, реализуемых посредством эксплоитов, в файлах неисполняемых форматов // Научно-практический журнал «Вопросы защиты информации» (Доверенная среда). – 2023. – С. 32–37.*
6. *Kamran Saeed, M. Fatih Adak Detection of Unknown Malicious Microsoft Office Documents based on Hidden Feature Extraction by using Machine Learning // Authorea. – 2024.– P. 1–16. DOI: 10.22541/au.170664344.41804021/v1.*
7. *Salman Abdul Jabbaar Wiharja, Deden Pradeka Wirmanto Suteddy, Designing A Pdf Malware Detection System Using Machine Learning // Jurnal Poli-Teknologi. – 2024. – Vol. 23, No. 1. – P. 40–54. DOI:10.32722/pt.v23i1.6540.*
8. *Fran Casino, Nikolaos Totosis, Theodoros Apostolopoulos, Nikolaos Lykousas Analysis and Correlation of Visual Evidence in Campaigns of Malicious Office Documents // Digital Threats Research and Practice. – 2022. – Vol. 4, No. 2. – P. 1–19. DOI:10.1145/3513025.*
9. *Candra Ahmadi, Jiann Chen, Yi-Cheng Lai Enhancing Detection of Malicious VBA Macros in Office Documents: An Integrated Approach Employing P-Code Analysis and XGBoost-based Machine Learning Model // IEEE Access. – 2024. – Vol. 12. – P. 71746–71760. DOI: 10.1109/ACCESS.2024.3402956.*
10. *V Ravi, S. P. Gururaj, H. K. Vedamurthy, M. B. Nirmala Analysing corpus of office documents for macro-based attacks using Machine Learning // Siddaganga Institute of Technology. – 2022. – Vol. 8, No. 3. – P. 20–24. DOI:10.1016/j.gltp.2022.04.004.*
11. *Geet C. Salame, Nirlepa T. Shinde, Prajakta P. Baad, Deepak D. Kshirsagar A. relational rule-based system for PDF malware detection // Journal of Information and Optimization Sciences. – 2024. – Vol. 45, No. 4. – P. 925–934. DOI:10.47974/JIOS-1616.*
12. *Maheshwaran T., Manideep M., Sai Chaitanya K., Karthik A. Securing pdfs: an innovative lstm algorithm for image-based malware detection // Interantional journal of scientific research in Engineering and Management. – 2024. – Vol. 8, No. 5. – P. 1–5. DOI:10.55041/IJSREM34090.*
13. *Старовойтов В. В., Голуб Ю. И. Сравнительный анализ оценок качества бинарной классификации // Информатика. – 2020. – Т. 17, № 1. – С. 87–101. DOI:10.37661/1816-0301-2020-17-1-87-101.*
14. *Bradley Efron. Bootstrap Methods: Another Look at the Jackknife // Annals of Statistics. – 1979. – Vol. 7, no. 1. – P. 1–26.*
15. *Donna L. M., William J. W., Rudolf J. F. Statistical Methods // University of North Florida. – 2021. – Vol. 4. – P. 123-167. DOI:10.1016/C2019-0-02521-6.*
16. *Кондаков С. Е., Архипов А. Н. Математическая модель эксплоита, внедренного в файл неисполняемого формата // Изв. ИИФ. 2023. Т. 69. № 3. С. 93–96.*
17. *Архипов А. Н., Кондаков С. Е. Сегментация файлов неисполняемых форматов для выявления угроз нарушения информационной безопасности, реализуемых в форме эксплоитов // Программные продукты и системы. 2024. Т. 37. № 2. С. 186–192. DOI: 10.15827/0236-235X.142.186-192.*

IDETECTING OBFUSCATED EXPLOITS IN NON-EXECUTABLE FORMAT FILES

Arkhipov A. N.[7], Kondakov S. E.[8]

The purpose of the research: *is the development of a model of binary classification of non-executable file formats, which provides increased efficiency of detection of obfuscated exploits, relative to the models implemented in existing anti-virus protection tools.*

Research methods *are based on the provisions of probability theory and mathematical statistics, set theory, methods of conducting field experiments and processing experimental data.*

Result: *in the course of the research, on the basis of the mathematical model of the exploit, a set of potential features, which are represented by numerical values, was generated. Informative features were selected from the generated feature space and a binary classification model with the best performance in detecting obfuscated exploits was built. A computer program implementing the obtained model was developed. The effectiveness of the developed model is confirmed in the framework of experimental studies to assess the effectiveness of detecting obfuscated exploits using anti-virus protection tools included in the register of Russian software, and foreign anti-virus protection tools placed in free access, and the author's program.*

The scientific novelty *of the results is determined by a set of author's procedures providing the choice of classifier, its hyperparameters, as well as the formation of an informative feature space, including features developed by the authors, and, allowing to build the most effective model of binary classification, which ensures the validity of the obtained results. The author presents the confirmation of realizability and obtaining the best values of efficiency indicators in detecting obfuscated exploits in relation to the existing means of antivirus protection.*

Practical significance: *the presented model, first of all, is oriented on application in antivirus protection systems, but it can be used for solving other tasks of information security.*

Keywords: *cybersecurity, computer attacks, local exploits, malicious code, information protection, anti-virus information protection systems, intrusion detection system.*

References

1. *Seredkin S.P. Osobennosti kiberatak na ob"ekty' kriticheskoj informacionnoj infrastruktury' v sovremenny'x usloviyax // Informacionny'e texnologii i matematicheskoe modelirovanie v upravlenii slozhny'mi sistemami. – 2022. – № 4 (16). – S. 56–66. DOI: 10.26731/2658-3704.2022.4(16).56-66.*
2. *Laneczkaya A. Yu., Aleksandrova E.N. Sovremenny'e ugrozy' informacionnoj bezopasnosti // Mezhdunarodny'j zhurnal gumanitarny'x i estestvenny'x nauk. – 2022. – Tom 7-2. – № 20. – S. 192–195. DOI:10.24412/2500-1000-2022-7-2-192-195.*
3. *Pavly'chev A. V., Starodubov M. I., Galimov A. D. Ispol'zovanie algoritma mashinnogo obucheniya Random Forest dlya vy'yavleniya slozhny'x komp'yuterny'x incidentov // Voprosy' kiberbezopasnosti. – 2022. – Tom 51. – № 5. – S. 74–81. DOI:10.21681/2311-3456-2022-5-74-81.*
4. *Taloverova D. V. Sravnitel'ny'j analiz scenariev realizacii ugroz bezopasnosti informacii metodiki FSTE'K RF i Mitre Att&ck i ix primenenie na praktike // Fundamental'ny'e i prikladny'e aspekty' komp'yuterny'x texnologij i informacionnoj bezopasnosti. Sbornik statej Vserossijskoj nauchno-texnicheskoj konferencii. Taganrog, 2023. – S. 34–37.*
5. *Arxipov A. N., V. A. Pikov, V. V. Kabakov Poryadok i rezul'taty' e'ksperimental'ny'x issledovanij vliyaniya obfuskacii na kachestvo vy'yavleniya ugroz informacionnoj bezopasnosti, realizuemy'x posredstvom e'ksploitov, v fajlax neispolnyaemy'x formatov // Nauchno-prakticheskij zhurnal «Voprosy' zashhity' informacii» (Doverennaya sreda). – 2023. – S. 32-37.*
6. *Kamran Saeed, M. Fatih Adak Detection of Unknown Malicious Microsoft Office Documents based on Hidden Feature Extraction by using Machine Learning // Authorea. – 2024.– P. 1–16. DOI: 10.22541/au.170664344.41804021/v1.*
7. *Salman Abdul Jabbaar Wiharja, Deden Pradeka Wirmanto Suteddy, Designing A Pdf Malware Detection System Using Machine Learning // Jurnal Poli-Teknologi. – 2024. – Vol. 23, No. 1. – P. 40-54. DOI:10.32722/pt.v23i1.6540.*
8. *Fran Casino, Nikolaos Totosis, Theodoros Apostolopoulos, Nikolaos Lykousas Analysis and Correlation of Visual Evidence in Campaigns of Malicious Office Documents // Digital Threats Research and Practice. – 2022. – Vol. 4, No. 2. – P. 1-19. DOI:10.1145/3513025.*
9. *Candra Ahmadi, Jiann Chen, Yi-Cheng Lai Enhancing Detection of Malicious VBA Macros in Office Documents: An Integrated Approach Employing P-Code Analysis and XGBoost-based Machine Learning Model // IEEE Access. – 2024. – Vol. 12. – P. 71746–71760. DOI: 10.1109/ACCESS.2024.3402956.*
10. *V Ravi, S. P. Gururaj, H. K. Vedamurthy, M. B. Nirmala. Analysing corpus of office documents for macro-based attacks using Machine Learning // Siddaganga Institute of Technology. – 2022. – Vol. 8, No. 3. – P. 20–24. DOI:10.1016/j.gltp.2022.04.004.*
11. *Geet C. Salame, Nirlepa T. Shinde, Prajakta P. Baad, Deepak D. Kshirsagar A. relational rule-based system for PDF malware detection // Journal of Information and Optimization Sciences. – 2024. – Vol. 45, No. 4. – P. 925–934. DOI:10.47974/JIOS-1616.*

7 Alexander N. Arkhipov, student Bauman Moscow State Technical University, Moscow, Russia. E-mail: diskpart111@mail.ru
8 Sergey E. Kondakov, Ph.D. (tech.), associate Professor, Bauman Moscow State Technical University, Moscow, Russia. E-mail: sergeikondakov@list.ru

12. *Maheshwaran T., Manideep M., Sai Chaitanya K., Karthik A. Securing pdfs: an innovative lstm algorithm for image-based malware detection // Interantional journal of scientific research in Engineering and Management. – 2024. – Vol. 8, No. 5. – P. 1–5. DOI:10.55041/IJSREM34090.*
13. *Starovojtov V. V., Golub Yu. I. Sravnitel'ny'j analiz ocenok kachestva binarnoj klassifikacii // Informatika. – 2020. – T. 17, № 1. – S. 87–101. DOI:10.37661/1816-0301-2020-17-1-87-101.*
14. *Bradley Efron. Bootstrap Methods: Another Look at the Jackknife // Annals of Statistics. – 1979. – Vol. 7, no. 1. – P. 1–26.*
15. *Donna L. M., William J. W., Rudolf J. F. Statistical Methods // University of North Florida. – 2021. – Vol. 4. – P. 123–167. DOI:10.1016/C2019-0-02521-6.*
16. *Kondakov S. E., Arxipov A. N. Matematicheskaya model' e'ksploita, vnedrennogo v fajl neispolnyaemogo formata // Izv. IIF. 2023. T. 69. № 3. S. 93–96.*
17. *Arxipov A. N., Kondakov S. E. Segmentaciya fajlov neispolnyaemy'x formatov dlya vy'yavleniya ugroz narusheniya informacionnoj bezopasnosti, realizuemy'x v forme e'ksploitov // Programmny'e produkty' i sistemy'. 2024. T. 37. № 2. S. 186–192. DOI: 10.15827/0236-235X.142.186-192.*

ПАТТЕРН ДЛЯ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ ВЕБ-ПРИЛОЖЕНИЯ ПРИ УГРОЗЕ XSS АТАК В ОБЛАЧНОЙ ИНФРАСТРУКТУРЕ

Корнеев Н. В.[1], Лазорин Д. С.[2]

DOI: 10.21681/2311-3456-2024-6-76-84

Цель статьи: *разработка шаблонного механизма защиты для обеспечения безопасности веб-приложения при угрозе межсайтового скриптинга.*

Метод исследования: *анализ принципа работы межсайтового скриптинга, в частности хранимого XSS. Синтез межсайтового сценария с помощью двух уровней защиты: кодирование данных на выходе и подтверждение ввода по прибытии. С использованием методов экранирования в Unicode, блокировки и применения нескольких уровней кодирования в правильном порядке для недопустимого ввода вредоносного кода предложены операции замены опасных символов на подходящие HTML-мнемоники. Исследование выполнено путем натурного моделирования веб-приложения на основе Docker в средах с поддержкой контейнеризации, его развёртывания и тестирования при угрозе межсайтового скриптинга.*

Результат: *проведен анализ облачной безопасности веб-приложений и показана актуальность проблемы разработки универсальных шаблонных механизмов безопасности, называемыми паттернами для защиты веб-приложения от XSS атак. В частности, рассмотрены принципы работы межсайтового скриптинга, типы XSS атак, и шаблонный механизмы защиты веб-приложения от XSS атак. Определена последовательность действий пользователя при входе в типовое веб-приложение. Построена микросервисная архитектура паттерна для защиты веб-приложения от XSS атак. Разработан паттерн для защиты веб-приложения от XSS атак на основе микросервисов с учётом сервиса безопасности, включающего механизмы защиты. На практическом примере реального веб-приложения развернуто 4 контейнера (nginx, php, mysql, phpmyadmin) и настроено взаимодействие между ними. Реализована форма регистрации и форма авторизации типового веб-приложения. Произведена XSS атака на веб-приложение кодом JavaScript без механизма защиты. В коде php реализован сервис безопасности для защиты от XSS атак с помощью встроенной функции htmlspecialchars. Приведен программный код сервиса в виде функции. Программный код сервиса безопасности включает функцию htmlspecialchars с кодом конфигурации и функциями взаимодействия с описанными выше контейнерами. Произведена повторная XSS атака в результате которой код JavaScript не был выполнен, данные безопасно были извлечены из базы данных. В результате в базе данных получена строка кода, которая является признаком диагностической ошибки веб-приложения, и может служить маркером для мониторинга блокированной XSS атаки. Для мониторинга XSS атаки использовано открытое программное обеспечение Kubernetes, Prometheus, Grafana, cAdvisor и Node Exporter. Созданы манифесты для реализации базовой конфигурации кластера, состоящего из одного пода с четырьмя контейнерами. В результате развёрнута система мониторинга XSS атаки и показана возможность диагностировать пики изменения нагрузки, как признаки блокированной XSS атаки.*

Практическая ценность: *практическая ценность предлагаемого решение включает шаблонный механизм защиты в виде паттерна. Паттерн можно применить для широкого круга веб-приложений, в том числе перенести разрабатываемое решение на любую отрасль: топливно-энергетическую, экономическую и не только, ввиду кроссплатформенности самого решения.*

Ключевые слова: *облачные вычисления, набор данных, шаблон, вредоносный код, сервис безопасности, контейнер, диагностическая ошибка, маркер, манифест, кластер, система мониторинга*

Введение

В последние десятилетия данные оказались незаменимыми для всех аспектов человеческого существования. Разработка нескольких приложений привела к экспоненциальному расширению объема информации. Эта информация может быть зашифрована и храниться в безопасных местах. Этому помогают облачные вычисления – технологии, которые можно использовать для хранения этих огромных наборов данных [1]. Киберпространство или облачные вычисления резко увеличили практическую полезность компьютеров и периферийных устройств [2]. Они широко используются в финансах, управлении бизнесом, телекоммуникациях, транспорте, образовании, здравоохранении и других сферах нашей повседневной жизни [3]. Это также позволяет пользователям эффективно общаться, совместно использовать программное обеспечение, оборудование, а также ресурсы данных через сетевые протоколы [4].

В эпоху облачных вычислений существует несколько возможностей, позволяющих временно

1 Корнеев Николай Владимирович, доктор технических наук, доцент, РГУ нефти и газа (НИУ) имени И. М. Губкина, Москва, Россия. E-mail: niccyper@mail.ru
2 Лазорин Данил Сергеевич, студент, РГУ нефти и газа (НИУ) имени И. М. Губкина, Москва, Россия. E-mail: lazorindanya@yandex.ru

кэшировать данные, хранящиеся удаленно, на настольных компьютерах, мобильных телефонах или других интернет-устройствах [5]. Отрасли программного обеспечения, а также частные лица, которые хранят свои данные в облаке, используют гибкий подход, который дает некоторые преимущества, такие как избежание капитальных затрат на личное обслуживание, оборудование и программное обеспечение [6].

Безопасность и надежность – две основные проблемы в облачных вычислениях. Данные клиента в облаке могут быть доступны другим клиентам, поэтому возникают проблемы с безопасностью данных клиентов. Для обеспечения безопасности облачных данных доступно множество методов и алгоритмов. За последние годы многими исследователями было предложено и реализовано много работ, касающихся облачной безопасности. Однако пробелом остается разработка универсальных шаблонных механизмов безопасности, называемыми паттернами. В частности, в данной статье мы ставим целью разработку шаблонного механизма защиты для обеспечения безопасности веб-приложения при угрозе межсайтового скриптинга.

Анализ и методы исследования

Межсайтовый скриптинг (XSS) [7] – это уязвимость веб-безопасности, которая позволяет злоумышленнику поставить под угрозу взаимодействие пользователей с уязвимым приложением. Это позволяет злоумышленнику обойти одну и ту же политику происхождения, которая предназначена для отделения разных веб-сайтов друг от друга. Уязвимости межсайтового скриптинга обычно позволяют злоумышленнику маскироваться под пользователя-жертву, выполнять любые действия, которые пользователь может выполнить, и получать доступ к любым данным пользователя. Если пользователь-жертва имеет привилегированный доступ к приложению, злоумышленник может получить полный контроль над всеми функциями и данными приложения.

Принцип работы межсайтового скриптинга заключается в манипулировании уязвимым веб-сайтом таким образом, чтобы он возвращал пользователям вредоносный код JavaScript. Когда вредоносный код выполняется внутри браузера жертвы, злоумышленник может полностью поставить под угрозу взаимодействие с веб-приложением.

На практике существуют три основных типа XSS атак:

1. Отраженный XSS (Reflected XSS) [8], где вредоносный скрипт поступает из текущего HTTP-запроса.
2. Хранимый XSS (Stored XSS) [9], где вредоносный скрипт поступает из базы данных сайта.
3. XSS на основе DOM (DOM Based XSS) [10], где уязвимость существует в коде на стороне клиента, а не в коде на стороне сервера.

В данной статье мы ограничимся рассмотрением только второго типа атаки ввиду широкой распространённости, а также ввиду указанной широкой практической применимости полученных результатов.

Хранимый XSS (также известный как постоянный XSS или XSS второго порядка) возникает, когда приложение получает данные из ненадежного источника и включает эти данные в свои последующие HTTP-ответы небезопасным способом.

Соответствующие данные могут быть отправлены в приложение через HTTP-запросы; например, при авторизации или регистрации пользователя. В других случаях данные могут поступать из других ненадежных источников; например, приложение веб-почты, отображающее сообщения, полученные по SMTP, маркетинговое приложение, отображающее сообщения в социальных сетях, или приложение мониторинга сети, отображающее пакетные данные из сетевого трафика.

Для ограничения зоны применимости разрабатываемого шаблонного механизма защиты определим, что нами будет рассматривается веб-приложение, которое строится на основе Docker – это программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации, контейнеризатор приложений. Также будут использованы следующие сервисы: веб-сервер NGINX, база данных MySQL, phpMyAdmin для управления базой данных, менеджер FastCGI Process Manager (FPM) для выполнения скриптов сайта. Для каждого сервиса будет запущен отдельный контейнер и настроено взаимодействие между ними с целью проверки механизма защиты.

В веб-приложении будет присутствовать главная страница с авторизацией пользователей, которая, при успешной регистрации, отправляет логин и пароль в базу данных. Далее будет происходить переадресация пользователя на страницу его профиля, где располагаются его персональные данные.

Предотвращение межсайтового сценария обычно можно обеспечить с помощью двух уровней защиты:

1. Кодирование данных на выходе. Кодирование следует применять непосредственно перед записью данных, управляемых пользователем, на страницу, поскольку контекст, в который записываются данные, определяет, какой тип кодировки необходимо использовать.

Например, значения внутри строки JavaScript требуют другого типа экранирования, чем значения в контексте HTML. В контексте HTML следует преобразовать значения, не внесенные в белый список,

в объекты HTML, а в контексте строки JavaScript небуквенно-цифровые значения должны быть экранированы в Unicode. Иногда может потребоваться применение нескольких уровней кодирования в правильном порядке.

2. Подтверждение ввода по прибытии. Кодирование является наиболее важной линией защиты от XSS, но его недостаточно для предотвращения XSS уязвимостей в любом контексте. Необходимо как можно более строго проверять вводимые данные в тот момент, когда они впервые получены от пользователя. В идеальном сценарии проверка ввода должна работать путем блокировки недопустимого ввода. Альтернативный подход, заключающийся в попытке очистить недопустимый ввод, чтобы сделать его действенным, более подвержен ошибкам, и его следует избегать, где это возможно.

Предотвращение XSS в PHP имеет свои особенности. В PHP есть встроенные функции для кодирования сущностей: htmlspecialchars, htmlentities. Следует вызвать одну из этих функцию, чтобы избежать ввода данных внутри контекста HTML. Функцию htmlentities следует вызывать с тремя аргументами:

1. Входная строка.
2. Флаг ENT_QUOTES, который указывает, что все кавычки должны быть закодированы.
3. Набор символов, который в большинстве случаев должен быть в UTF.

Приведем пример 1:

```
<?php echo
htmlentities($input,ENT_QUOTES,'UTF- 8');?>.
```

В контексте строки JavaScript необходимо экранировать ввод в Unicode, как уже было описано ранее.

Функцию htmlspecialchars следует вызывать следующим образом:

1. Проверить, была ли отправлена переменная $_POST['name'].
2. Если переменная была отправлена, то присвоить значение переменной результату функции, которая принимает входную строку (значение $_POST['name']) и флаг ENT_QUOTES, чтобы кодировать кавычки, если таковые имеются в строке. В большинстве случаев набор символов должен быть в UTF-8, поэтому можно указать его третьим аргументом.
3. Если переменная $_POST['name'] не была отправлена или была отправлена пустая строка, присвоить переменной пустую строку.

Приведем пример 2:

```
$name=isset($_POST['name'])?htmlspecialchars
($_POST['name'], ENT_QUOTES, 'UTF-8'):'';.
```

Подводя итог анализу, следует сказать, что для реализации поставленной нами цели для каждого конкретного веб-приложения следует действовать по следующему алгоритму:

1. Провести анализ в области межсайтового скриптинга для условий эксплуатации конкретного веб-приложения.
2. Разработать паттерн веб-приложения, который позволит продемонстрировать схему защиты от угрозы межсайтового скриптинга.
3. Разработать контейнер, который включает в себя все необходимые зависимости, библиотеки и системные инструменты для запуска веб-приложения.
4. Произвести тестирование и мониторинг разработанного контейнера на кластере, с целью проверки его работоспособности от угрозы межсайтового скриптинга.

Далее в статье мы покажем реализацию данного алгоритма для типового решения, подходящего для большинства веб-приложений, а при необходимости, он может быть расширен с учетом особенностей практической реализации в каждом конкретном случае.

В рассматриваемом случае последствиями реализации угроз является нарушение конфиденциальности, целостности, доступности.

Разрабатываемый нами паттерн может быть использован не только для защиты описанного веб-приложения, но и в более широком контексте – для обеспечения безопасности веб-приложений в целом, например, в системе SIEM (Security Information and Event Management) с учетом доработки по описанному выше алгоритму.

Новизна предлагаемого решения определяется возможностью обеспечения безопасности веб-приложения при угрозе XSS атак в облачной информационной инфраструктуре России при переходе на импортозамещение.

Практическая значимость предлагаемого решение включает шаблонный механизм защиты в виде паттерна, который можно применить для широкого круга веб-приложений, в том числе перенести разрабатываемое решение на любую отрасль: топливно-энергетическую, экономическую и не только, ввиду кроссплатформенности самого решения.

Разработка паттерна веб-приложения

Для разработки паттерна веб-приложения, необходимо определить последовательность действий пользователя при входе в веб-приложение. Это ограничивает условие эксплуатации самого паттерна, с одной стороны, а с другой дает нам формализовать рассматриваемую задачу. Рассмотрим последовательность действий пользователя поэтапно.

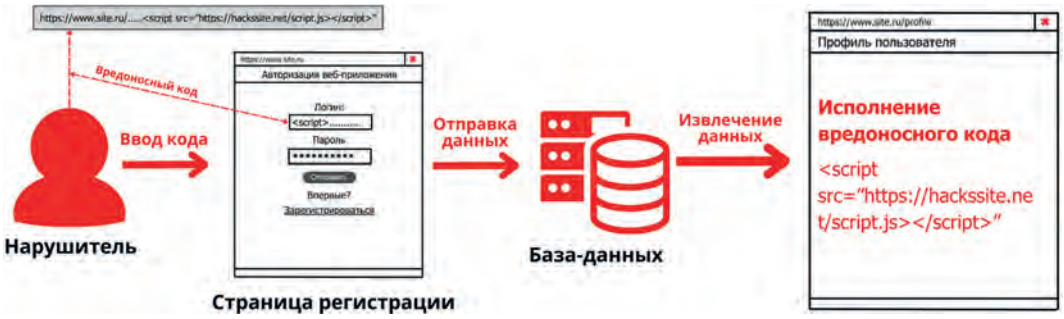


Рис. 1. Сценарий XSS атаки на этапе 2

Рис. 2. Сценарий защиты веб-приложения от XSS атаки

Этап 1. При открытии веб-приложения пользователь находится на главной странице: авторизации веб-приложения. Если же он впервые оказался в веб-приложении, то необходимо произвести регистрацию.

Этап 2. Если пользователь прошел регистрацию, то данные вносятся в базу данных и происходит переадресация на страницу с профилем. Если пользователь уже зарегистрирован, происходит проверка введенных данных, затем дальнейшая переадресация.

Этап 3. Далее пользователь попадает на страницу своего профиля, где отображается информация (персональные данные), извлекаемая из базы данных.

Системная архитектура паттерна веб-приложения строится нами на основе микросервисов. Сама архитектура микросервисов (Microservice Architecture, MA) представляет собой новую парадигму архитектуры программного обеспечения, которая направлена на решение ограничений традиционного монолитного программного обеспечения путем разложения всего программного обеспечения на независимо развертываемые и масштабируемые более мелкие части, называемые сервисами или микросервисами [11, 12]. Идея этой декомпозиции заключается в разделении различных функций системы. В веб-приложении за свой функционал отвечают следующие сервисы: сервис регистрации, сервис авторизации, сервис профиля.

Отметим, что в данном случае, как и в реальных существующих практических решениях, никакой другой обработки данных веб-приложение не осуществляет, поэтому злоумышленник легко может отправить сообщение, атакующее других пользователей. Исходя из этого может возникнуть XSS атака на этапе 2. Подобный сценарий изображен на (рис. 1).

Для решения данной проблемы необходимо реализовать дополнительный сервис, включающий в себя механизм защиты веб-приложения (рис. 2), включающий в себя операции замены опасных символов на подходящие HTML-мнемоники. На (рис. 3) продемонстрирована архитектура предлагаемого паттерна веб-приложения на основе микросервисов с сервисом безопасности для решения указанной проблемы.

Рис.3. Микросервисная архитектура паттерна веб-приложения

Разработка контейнеров для сервисов

Разработка контейнеров для работы сервисов производилась на операционной системе Windows 10 Pro с помощью Microsoft Visual Studio и Docker Desktop согласно типовым инструкциям настройки и конфигурирования [13, 14].

Нами была создана основная директория docker-projects, в которой были созданы каталоги images/php (директория для созданных вручную образов, образ php со всеми необходимыми модулями),

mysql-data (директория для физического хранения файлов баз-данных), www (корневая директория веб-приложения для хранения всех php-скриптов и ресурсов сайта). Далее последовательно выполнена настройка всех контейнеров для совместной работы согласно типовым инструкциям настройки и конфигурирования.

1. Контейнер веб-сервера nginx. Для того чтобы веб-сервер был доступен вне контейнера, необходимо соединить порт 80 контейнера с портом операционной системы. Это можно выполнить с помощью команды:

```
docker run -d -p 80:80 nginx.
```

Создаём файл vhost.conf и настраиваем конфигурацию, чтобы веб-сервер открывал страницу нашего сайта.

Далее передаем конфигурацию веб-сервера внутрь контейнера. Это делается с помощью команды:

```
docker container run -d -p 80:80 -v
"${PWD}/vhost.conf:/etc/nginx/conf.d/
default.conf nginx".
```

В директории www создаём файл index.html. Далее запускаем контейнер с помощью команды:

```
docker container run -d -p 80:80 -v
"${PWD}/vhost.conf:/etc/nginx/conf.d/
default.conf" - v "${PWD}/www:/var/www/
public_html nginx".
```

Далее необходимо проверить, отобразилась ли страница index.html.

2. Контейнер php. Необходимо создать образ с помощью типовых инструкций по сборке в специальном файле Dockerfile, который создаётся в директории images/php/. Далее собираем контейнер с помощью команды:

```
docker build -t php8lfpm:l.0
"${PWD}/images/php".
```

Необходимо настроить взаимодействие между контейнерами php81fpm и nginx. Команда:

```
docker inspect <id_container> |
grep IPAddress,
```

выведет все параметры выбранного контейнера (в нашем случае php81fpm). Настроим IP-адрес (порт 9000).

При запуске контейнеров IP-адреса попадают в сеть по умолчанию, в которой к контейнеру можно получить доступ по его IP-адресу, но нельзя по его имени. Чтобы иметь возможность использовать имена контейнеров нужно запускать их в пользовательской сети, предварительно создав её. Для этого используется команда:

```
docker network create networkl.
```

Далее запускаем контейнеры в созданной сети командой:

```
Docker run –d –p 80:80 -v
"${PWD}/vhost.conf:/etc/nginx/conf.d/
default.conf”-v "${PWD}/www:/var/www/public_
html" --network networkl --name nginxl
nginx, docker run -d -v "${PWD}/www:/var/
www/public_html" --network networkl --name
phpl php8lfpm.
```

3. Контейнер MySQL. Необходимо создать контейнер с помощью команды:

```
docker run -d -v "${PWD}/mysql-data:/
var/lib/mysql" -e MYSQL_ROOT_PASSWORD=root
--network network1 --name mysql1 mysql,
```

и проверить подключение к тестовой базе данных. В нашем случае проверка показала, что подключение произошло успешно и данные из БД извлекаются.

4. Контейнер phpMyAdmin. Необходимо создать контейнер с помощью команды:

```
docker run -d -p 1500:80 -e PMA_
HOST=mysql1 --network networkl --name
phpmyadminl phpmyadmin.
```

В браузере перейдем по адресу localhost:1500. Отобразится панель управления phpMyAdmin.

Таким образом, было создано 4 контейнера и настроено взаимодействие между ними.

Для проверки защиты веб-приложения от угрозы межсайтового скриптинга проведем настройку конфигураций страниц веб-приложения, которая позволит продемонстрировать схему защиты.

Конфигурация страниц веб-приложения и проведение тестовой XSS атаки

Была реализована форма регистрации (рис. 4) и форма авторизации (рис. 5). При регистрации в скрипте php осуществляется проверка введённых данных и внесение их в базу данных, далее перенаправление на страницу с авторизацией. При авторизации происходит сверка введенных данных с базой данных и перенаправление на страницу с профилем. В профиле имя пользователя используется из базы данных. Введены ограничения на минимальное и максимальное количество символов в полях: от 3 до 255 символов, при выходе за данные пределы появляется сообщение об ошибке. При вводе разных паролей появляется сообщение об ошибке. При вводе почтового адреса, который уже внесён в базу данных, появляется сообщение об ошибке.

Проведем тестовую XSS атаку поэтапно:

Этап 1. При регистрации в поле ФИО (рис. 4) укажем JavaScript код:

```
<script>alert(’Код’)</script>,
```

и заполним остальные поля так, как показано на (рис. 4).

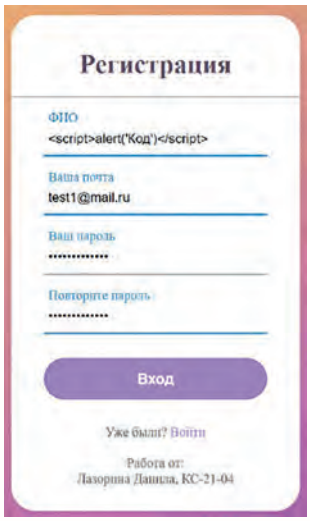


Рис. 4. Форма регистрации

Этап 2. Регистрация на этапе 1 прошла успешно, далее произведем авторизацию так, как показано на (рис. 5).

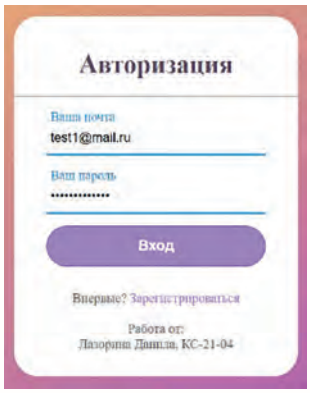


Рис. 5. Форма авторизации

Этап 3. После авторизации выполняется код, указанный в поле ФИО при регистрации (рис. 5). Следовательно, уязвимость в веб-приложении присутствует.

Этап 4. Переход в профиль. Отметим следующее: имя пользователя уже не отображается – XSS атака реализована (рис. 6).

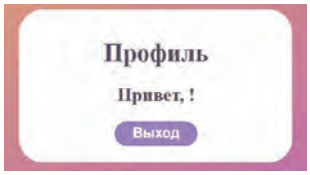


Рис. 6. Профиль пользователя с успешной реализацией XSS атаки

Реализация сервиса безопасности от XSS атаки

Как видно на этапе 3, проблема заключается в том, что переменные переносятся в базу данных такими, какими ввёл их пользователь, без дополнительной обработки. Необходимо осуществить механизм защиты от подобного рода атак. В этом могут помочь HTML-мнемоники – кодовое представление символа в HTML, который начинается со знака амперсанда и завершается точкой с запятой. Теги <script> состоят из треугольных скобок, следовательно, их необходимо заменить на мнемоники. Благодаря этому текст не будет трактоваться браузером как HTML-тег.

В php-сценариях на этапах регистрации и авторизации необходимо реализовать сервис безопасности, включающий в себя вызов функции, которая для переданной строки выполнит фильтрацию и заменит все опасные символы в ней на подходящие HTML-мнемоники. Такая функция в нашем случае называется htmlspecialchars.

Приведем пример кода для обработки поля ФИО пользователя с помощью данной функции:

```
$name=isset($_POST['name'])?htmlspecialch
ars($_POST['name']):'';.
```

Повторим этап 1 и этап 2. Результат авторизации показан на (рис.7).

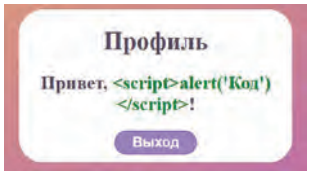


Рис. 7. Профиль пользователя с блокированной XSS атакой

Как видно из (рис. 7), код JavaScript не был выполнен, данные безопасно были извлечены из базы данных. В БД данные в поле ФИО для пользователя выглядят строкой кода:

```
&lt;script&gt;alert(&#039;Код&#039;)&lt;/
script&gt,
```

что является признаком диагностической ошибки веб-приложения, и может служить маркером для мониторинга блокированной XSS атаки.

Таким образом, в коде php реализован сервис безопасности с помощью встроенной функции htmlspecialchars, с его помощью удалось реализовать механизм защиты от XSS атаки. Программный код в виде функции:

```
function sanitizeInput($input) {
return isset($input) ?
htmlspecialchars($input) : “” ;
}
$name = sanitizeInput($_POST['name']).
```

Программный код сервиса безопасности включает встроенную функцию htmlspecialchars с кодом конфигурации и функциями взаимодействия с контейнерами.

Мониторинг XSS атаки и обсуждение результатов

Для мониторинга XSS атаки может служить указанный выше маркер в виде соответствующей диагностической ошибки веб-приложения. Для реализации системы мониторинга веб-приложения на выделенном кластере нами использовано открытое программное обеспечение Kubernetes [15]. В Kubernetes наименьшей единицей является «под» – это абстрактный объект, представляющий собой группу из одного или нескольких контейнеров приложения (в нашем случае, Docker).

Для реализации базовой конфигурации кластера, состоящего из одного пода с четырьмя контейнерами, создан специальный манифест config.yaml. Манифест описывает развертывание веб-приложения, состоящего из веб-сервера nginx, сервера обработки php (php-fpm), базы данных mysql и интерфейса администрирования базы данных phpmyadmin, а также настройку сервиса для доступа к веб-серверу nginx. Развёртывание кластера осуществляется командой:

```
kubectl create -f config.yaml.
```

Для реализации системы мониторинга нами использованы: Prometheus и Grafana [16], а также cAdvisor и Node Exporter [17].

Prometheus – это набор инструментов для мониторинга и оповещения систем с открытым исходным кодом.

Grafana – это платформа для мониторинга, анализа данных и визуализации собранных данных с открытым исходным кодом. Grafana упрощает мониторинг и анализ состояния системы для разработчиков и администраторов. Таким образом, вместе они обеспечивают мощный инструментарий для эффективного контроля и оптимизации работы IT-инфраструктуры любого уровня.

cAdvisor – предоставляет данные по использованию ресурсов и производительности запущенных контейнеров. Формирует метрики в читаемом для Prometheus формате.

Node Exporter – это агент сбора метрик системы, таких как использование процессора, памяти, дисков, сети, а также самостоятельных метрик в виде соответствующей диагностической ошибки веб-приложения.

Для упрощения развёртывания системы мониторинга нами использован Docker Compose, который обеспечивает управление всеми этапами в жизненном цикле определенной службы: запуском, остановкой и перестроением служб; просмотром состояния службы, потоковой передачей журналов. Созданы манифест docker-compose.yml, директория prometheus с манифестом prometheus.yml, директория grafana с манифестом config.monitoring, поддиректория provisioning/dashboards с манифестом dashboard.yml и поддиректория provisioning/datasources с манифестом datasource.yml.

При использовании Docker Compose также потребуется установить переменные в Windows PowerShell, чтобы преобразовать пути Windows, присутствующие в сопоставлениях томов Docker Compose.

На (рис. 8) показана витрина Docker Containers в Grafana, которая отображает информацию о наших контейнерах Docker, запущенных на хостовой машине. Здесь демонстрируются данные о ресурсах, потребляемых каждым контейнером, такие как CPU, память, сеть и дисковое пространство, а также другие метрики, в том числе самостоятельные метрики

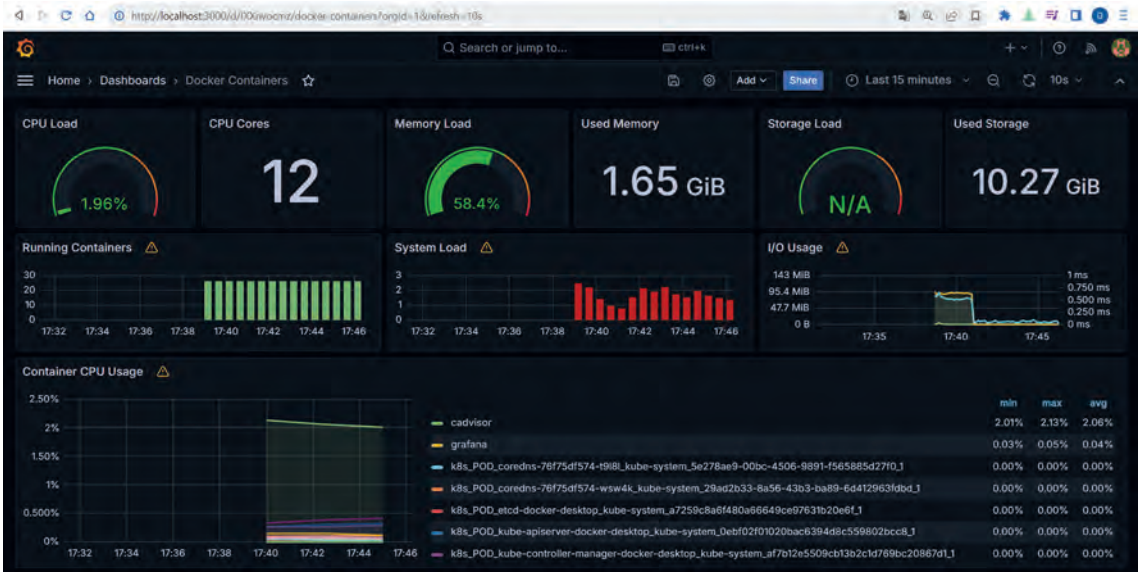


Рис. 8. Витрина Docker Containers в Grafana

в виде соответствующей диагностической ошибки веб-приложения, которые, собираются с помощью инструментов мониторинга.

Пик изменения нагрузки на (рис. 8) в 17:40 является признаком диагностической ошибки веб-приложения, и может служить маркером для мониторинга блокированной XSS атаки. В то же время можно настроить дополнительные метрики, более чувствительные к соответствующей диагностической ошибки веб-приложения.

Выводы

Построена микросервисная архитектура паттерна для защиты веб-приложения от XSS атак для широко круга веб-приложений в облачной инфраструктуре. Разработан паттерн для защиты веб-приложения от XSS атак на основе микросервисов, интегрированных в контейнеры, с учётом сервиса безопасности, включающего механизмы защиты от XSS-атак, которые могут быть использованы как шаблон по построению аналогичных систем безопасности от XSS атак. В коде php реализован сервис безопасности от XSS атак с помощью встроенной функции htmlspecialchars. Программный код сервиса безопасности включает встроенную функцию htmlspecialchars с кодом конфигурации и функциями взаимодействия с контейнерами. Произведена практическая проверка работы сервиса безопасности путем натурной имитации XSS атаки на типовое веб-приложение, которая показала его эффективность в отношении отражения XSS атаки. Для мониторинга XSS-атаки на веб-приложение использовано открытое программное обеспечение. Созданы манифесты для реализации базовой конфигурации кластера, состоящего из одного пода с четырьмя контейнерами. В результате развёрнута система мониторинга XSS атаки и показана возможность диагностировать пики изменения нагрузки, как признаки блокированной XSS атаки, которая может быть использована в системах SIEM.

Литература

1. *Shameer Mohammed, S. Nanthini, N. Bala Krishna, Inumarthi V. Srinivas, Manikandan Rajagopal, M. Ashok Kumar, A new lightweight data security system for data security in the cloud computing, Measurement: Sensors, Volume 29, 2023, 100856.*
2. *S. Achar, Cloud computing security for multi-cloud service providers: controls and techniques in our modern threat landscape, International Journal of Computer and Systems Engineering, 16(9), 2022, 379–384.*
3. *Oludare Isaac Abiodun, Moatsum Alawida, Abiodun Esther Omolara, Abdulatif Alabdulatif, Data provenance for cloud forensic investigations, security, challenges, solutions and future perspectives: A survey, Journal of King Saud University – Computer and Information Sciences, Volume 34, Issue 10, Part B, 2022, 10217–10245.*
4. *Chakraborti, A., Curtmola, R., Katz, J., Nieh, J., Sadeghi, A. R., Sion, R., Zhang, Y., Cloud Computing Security: Foundations and Research Directions. Foundations and Trends in Privacy and Security, 3(2), 2022, 103–213.*
5. *Ukeje, N., Gutierrez, J., Petrova, K., Information security and privacy challenges of cloud computing for government adoption: a systematic review, International Journal of Information Security, Issue 2/2024, 2024, https://doi.org/10.1007/s10207-023-00797-6.*
6. *Fatemeh Khoda Parast, Chandni Sindhav, Seema Nikam, Hadiseh Izadi Yekta, Kenneth B. Kent, Saqib Hakak, Cloud computing security: A survey of service-based models, Computers & Security,Volume 114, 2022, 102580.*
7. *Faizan Younas, Ali Raza, Nisrean Thalji, Laith Abualigah, Raed Abu Zitar, Heming Jia, An efficient artificial intelligence approach for early detection of cross-site scripting attacks, Decision Analytics Journal, Volume 11, 2024, 100466.*
8. *Wenbo Wang, Peng Yi, Huikai Xu, DoubleR: Effective XSS attacking reality detection, Computer Networks,Volume 251, 2024, 110567.*
9. *Abdelhakim Hannousse, Salima Yahiouche, Mohamed Cherif Nait-Hamoud, Twenty-two years since revealing cross-site scripting attacks: A systematic mapping and a comprehensive survey, Computer Science Review, Volume 52, 2024, 100634.*
10. *Josh Hickling, What is DOM XSS and why should you care?, Computer Fraud & Security, Volume 2021, Issue 4, 2021, 6–10.*
11. *Diogo Faustino, Nuno Gonçalves, Manuel Portela, António Rito Silva, Stepwise migration of a monolith to a microservice architecture: Performance and migration effort evaluation, Performance Evaluation, Volume 164, 2024, 102411.*
12. *Hassaan Siddiqui, Ferhat Khendek, Maria Toeroe, Microservices based architectures for IoT systems – State-of-the-art review, Internet of Things, Volume 23, 2023, 100854.*
13. *Hubin Yang, Ruochen Shao, Yanbo Cheng, Yucong Chen, Rui Zhou, Gang Liu, Guoqi Xie, Qingguo Zhou, REDB: Real-time enhancement of Docker containers via memory bank partitioning in multicore systems, Journal of Systems Architecture, Volume 151, 2024, 103135.*
14. *Enrico Cambiaso, Luca Caviglione, Marco Zuppelli, DockerChannel: A framework for evaluating information leakages of Docker containers, SoftwareX, Volume 24, 2023, 101576.*
15. *Gianluca Turin, Andrea Borgarelli, Simone Donetti, Ferruccio Damiani, Einar Broch Johnsen, S. Lizeth Tapia Tarifa, Predicting resource consumption of Kubernetes container systems using resource models, Journal of Systems and Software, Volume 203, 2023, 111750.*
16. *Vladimir Ciric, Marija Milosevic, Danijel Sokolovic, Ivan Milentijevic, Modular deep learning-based network intrusion detection architecture for real-world cyber-attack simulation, Simulation Modelling Practice and Theory, Volume 133, 2024, 102916.*
17. *Miguel Correia, Wellington Oliveira, José Cecílio, Monintainer: An orchestration-independent extensible container-based monitoring solution for large clusters, Journal of Systems Architecture, Volume 145, 2023, 103035.*

PATTERN FOR SECURING WEB APPLICATIONS AGAINST XSS ATTACKS IN CLOUD INFRASTRUCTURE

Korneev N. V.[3], Lazorin D. S.[4]

The purpose of this article: *To develop a template protection mechanism to ensure the security of a web application in the event of a cross-site scripting threat.*

3 Nikolai V. Korneev, Dr.Sc., Professor, Gubkin Russian State University of Oil and Gas (National Research University), Moscow, Russia. E-mail: niccyper@mail.ru
4 Danil S. Lazorin, student, Gubkin Russian State University of Oil and Gas (National Research University), Moscow, Russia. E-mail: lazorindanya@yandex.ru

Research method: *Analysis of the principle of operation of cross-site scripting, in particular stored XSS. Synthesis of a cross-site script using two levels of protection: data encoding at the output and confirmation of input on arrival. Using methods of escaping in Unicode, blocking and applying several encoding levels in the correct order for invalid input of malicious code, operations are proposed to replace dangerous characters with suitable HTML mnemonics performed by full-scale modeling of a Docker-based web application in containerization-enabled environments, its deployment and testing under the threat of cross-site scripting.*

Result: *The analysis of cloud security of web applications is carried out and the relevance of the problem of developing universal template security mechanisms, called patterns for protecting a web application from XSS attacks, is shown. In particular, the principles of cross-site scripting, types of XSS attacks, and template mechanisms for protecting a web application from XSS attacks are considered pattern to protect the web application from XSS attacks. A pattern has been developed to protect a web application from XSS attacks based on microservices, taking into account a security service that includes protection mechanisms. On a practical example of a real web application, 4 containers (nginx, php, mysql, phpmyadmin) are deployed and interaction between them is configured. A registration form and an authorization form for a standard web application have been implemented. An XSS attack was performed on a web application using JavaScript code without a protection mechanism. The php code implements a security service to protect against XSS attacks using the built-in htmlspecialchars function. The program code of the service in the form of a function is given. The security service code includes an htmlspecialchars function with configuration code and interaction with the containers described above. A second XSS attack was performed, as a result of which the JavaScript code was not executed, the data was safely retrieved from the database. As a result, a line of code is obtained in the database, which is a sign of a diagnostic error of the web application, and can serve as a marker to monitor the XSS blocked attack. Open source software Kubernetes, Prometheus, Grafana, cAdvisor and Node Exporter were used to monitor the XSS attack. Manifests have been created to implement a basic cluster configuration consisting of a single pod with four containers. As a result, an XSS attack monitoring system was deployed and the ability to diagnose spikes in load changes as signs of a blocked XSS attack was shown.*

Practical value: *The practical value of the proposed solution includes a template protection mechanism in the form of a pattern. The pattern can be applied to a wide range of web applications, including transferring the developed solution to any industry: fuel and energy, economic and not only, due to the cross-platform nature of the solution itself.*

Keywords: *cloud computing, dataset, template, malicious code, security service, container, diagnostic error, marker, manifest, cluster, XSS attack monitoring system.*

References

1. *Shameer Mohammed, S. Nanthini, N. Bala Krishna, Inumarthi V. Srinivas, Manikandan Rajagopal, M. Ashok Kumar, A new lightweight data security system for data security in the cloud computing, Measurement: Sensors, Volume 29, 2023, 100856.*
2. *S. Achar, Cloud computing security for multi-cloud service providers: controls and techniques in our modern threat landscape, International Journal of Computer and Systems Engineering, 16(9), 2022, 379–384.*
3. *Oludare Isaac Abiodun, Moatsum Alawida, Abiodun Esther Omolara, Abdulatif Alabdulatif, Data provenance for cloud forensic investigations, security, challenges, solutions and future perspectives: A survey, Journal of King Saud University – Computer and Information Sciences, Volume 34, Issue 10, Part B, 2022, 10217–10245.*
4. *Chakraborti, A., Curtmola, R., Katz, J., Nieh, J., Sadeghi, A. R., Sion, R., Zhang, Y., Cloud Computing Security: Foundations and Research Directions. Foundations and Trends in Privacy and Security, 3(2), 2022, 103–213.*
5. *Ukeje, N., Gutierrez, J., Petrova, K., Information security and privacy challenges of cloud computing for government adoption: a systematic review, International Journal of Information Security, Issue 2/2024, 2024, https://doi.org/10.1007/s10207-023-00797-6.*
6. *Fatemeh Khoda Parast, Chandni Sindhav, Seema Nikam, Hadiseh Izadi Yekta, Kenneth B. Kent, Saqib Hakak, Cloud computing security: A survey of service-based models, Computers & Security,Volume 114, 2022, 102580.*
7. *Faizan Younas, Ali Raza, Nisrean Thalji, Laith Abualigah, Raed Abu Zitar, Heming Jia, An efficient artificial intelligence approach for early detection of cross-site scripting attacks, Decision Analytics Journal, Volume 11, 2024, 100466.*
8. *Wenbo Wang, Peng Yi, Huikai Xu, DoubleR: Effective XSS attacking reality detection, Computer Networks,Volume 251, 2024, 110567.*
9. *Abdelhakim Hannousse, Salima Yahiouche, Mohamed Cherif Nait-Hamoud, Twenty-two years since revealing cross-site scripting attacks: A systematic mapping and a comprehensive survey, Computer Science Review, Volume 52, 2024, 100634.*
10. *Josh Hickling, What is DOM XSS and why should you care?, Computer Fraud & Security, Volume 2021, Issue 4, 2021, 6–10.*
11. *Diogo Faustino, Nuno Gonçalves, Manuel Portela, António Rito Silva, Stepwise migration of a monolith to a microservice architecture: Performance and migration effort evaluation, Performance Evaluation, Volume 164, 2024, 102411.*
12. *Hassaan Siddiqui, Ferhat Khendek, Maria Toeroe, Microservices based architectures for IoT systems – State-of-the-art review, Internet of Things, Volume 23, 2023, 100854.*
13. *Hubin Yang, Ruochen Shao, Yanbo Cheng, Yucong Chen, Rui Zhou, Gang Liu, Guoqi Xie, Qingguo Zhou, REDB: Real-time enhancement of Docker containers via memory bank partitioning in multicore systems, Journal of Systems Architecture, Volume 151, 2024, 103135.*
14. *Enrico Cambiaso, Luca Caviglione, Marco Zuppelli, DockerChannel: A framework for evaluating information leakages of Docker containers, SoftwareX, Volume 24, 2023, 101576.*
15. *Gianluca Turin, Andrea Borgarelli, Simone Donetti, Ferruccio Damiani, Einar Broch Johnsen, S. Lizeth Tapia Tarifa, Predicting resource consumption of Kubernetes container systems using resource models, Journal of Systems and Software, Volume 203, 2023, 111750.*
16. *Vladimir Ciric, Marija Milosevic, Danijel Sokolovic, Ivan Milentijevic, Modular deep learning-based network intrusion detection architecture for real-world cyber-attack simulation, Simulation Modelling Practice and Theory, Volume 133, 2024, 102916.*
17. *Miguel Correia, Wellington Oliveira, José Cecílio, Monintainer: An orchestration-independent extensible container-based monitoring solution for large clusters, Journal of Systems Architecture, Volume 145, 2023, 103035.*

ПРОБЛЕМНЫЕ ВОПРОСЫ УПРАВЛЕНИЯ ЗАЩИТОЙ ИНФОРМАЦИИ ОТ УТЕЧКИ ПО ТЕХНИЧЕСКИМ КАНАЛАМ С ПРИМЕНЕНИЕМ МНОГОАГЕНТНЫХ СИСТЕМ

Язов Ю. К.[1], Авсентьев А. О.[2]

DOI: 10.21681/2311-3456-2024-6-85-97

Цель статьи: *раскрыть проблемные вопросы защиты информации от утечки по техническим каналам, возникающим за счет побочных электромагнитных излучений, с применением перспективных многоагентных систем и управления ими, показать необходимость и пути количественной оценки эффективности такой защиты.*

Методы исследования: *применены методы морфологического и функционально-структурного анализа процессов распределенного управления защитой информации от утечки по техническим каналам, а также методы теории вероятностей и теории составных сетей Петри-Маркова в интересах моделирования и оценки эффективности процессов централизованно-децентрализованного управления защитой.*

Полученный результат: *показана актуальность создания многоагентной системы защиты информации от утечки по техническим каналам; отмечена необходимость управления защитой в таких системах, раскрыты особенности централизованно-децентрализованного (смешанного) принципа управления в многоагентной системе на примере защиты речевой информации от утечки по техническим каналам, возникающим за счет побочных электромагнитных излучений радиоэлектронного оборудования в составе объектов информатизации.*

Раскрыты проблемные вопросы построения подсистем управления в составе многоагентных систем защиты информации от утечки по техническим каналам, возникающим за счет побочных электромагнитных излучений, связанных с понятием и формированием показателей эффективности защиты, влиянием управления защитой на ее эффективность, распределения управляющих воздействий по субъектам управления. Приведены составная сеть Петри-Маркова, моделирующая процесс утечки речевой информации по побочным электромагнитным излучениям, и аналитические соотношения для расчета показателя эффективности управления защитой информации в многоагентной системе.

Научная новизна *статьи состоит в том, что в ней впервые поставлена проблема реализации смешанного принципа управления защитой информации от утечки по техническим каналам на основе многоагентной системы и рассмотрены первоочередные методологические аспекты количественной оценки эффективности такой защиты.*

Ключевые слова: *побочное электромагнитное излучение, управление защитой, смешанный принцип управления, эффективность защиты, эффективность управления, мера защиты, частный показатель, математическая модель.*

Введение

На объектах информатизации (ОИ), создаваемых в интересах обеспечения деятельности различных организаций, условия реализации информационных процессов по обработке информации могут значительно отличаться. Это обусловлено, во-первых, различиями форм представления этой информации и ее материальных носителей, во-вторых, использованием для ее обработки различных технических средств и систем, в-третьих, отличиями архитектурных характеристик зданий, сооружений и помещений, в которых эти средства и системы размещаются [1]. В связи с тем, что такого рода информационные процессы реализуются во времени, то в совокупности с указанными обстоятельствами это, с одной стороны, обусловливает специфику динамики этих процессов, а с другой – определяет условия реализации угроз безопасности информации, в том числе угроз ее утечки по техническим каналам (ТКУИ) [1, 2].

Меры защиты информации (ЗИ) от утечки по ТКУИ, как правило, реализуются в составе систем защиты информации (СЗИ), развертываемых на ОИ. Для их выбора и применения в составе СЗИ создается подсистема управления защитой. В настоящее время управление в функционирующих СЗИ осуществляется по централизованному принципу из одного центра управления в составе ОИ. В [3] отмечено, что на больших ОИ, включающих десятки и сотни датчиков и аппаратных или программно-аппаратных элементов средств защиты, состав и настройки которых необходимо изменять в динамике изменения обстановки, централизованное построение и управление системой из-за большого количества процедур анализа и принятия решений по управлению с высокой вероятностью может приводить к неадекватным решениям и, как следствие, к снижению эффективности защиты информации на ОИ. С целью

1 Язов Юрий Константинович, доктор технических наук, профессор, главный научный сотрудник управления ФАУ «ГНИИИ ПТЗИ ФСТЭК России», г. Воронеж, Российская Федерация. E-mail: Yazoff_1946@mail.ru

2 Авсентьев Александр Олегович, кандидат технических наук, доцент кафедры компьютерной безопасности и технической экспертизы ФГКОУ ВО «Воронежский институт Министерства внутренних дел Российской Федерации», г. Воронеж, Российская Федерация. E-mail: aoaao8787@mail.ru

учета условий реализации разнородных процессов обработки информации на ОИ, ее перехвата по ТКУИ и защиты от перехвата в [3] предложено переходить к централизованно-децентрализованному (смешанному) принципу управления СЗИ на основе многоагентной системы защиты информации (МАСЗИ). В этом случае система решений в ходе управления защитой распределяется между агентами МАСЗИ, а сами агенты распределяются по территории ОИ и его элементам. Эффективность защиты информации как степень соответствия результата защиты, цели защиты в этих условиях следует рассматривать с учетом того, насколько эффективно управление МАСЗИ.

Следует отметить, что в настоящее время исследованиям, связанным с разработкой и применением многоагентных систем в других сферах деятельности, уделяется значительное внимание. В основном такие исследования посвящены развитию теории агентов, исследованию математических методов описания их свойств, архитектуры построения, как агентов, так и систем в целом, методов и средств их коммуникации, методов и программных средств поддержки миграции агентов и др. [4;5;6;7], а также[3,4,5]. Однако применение МАСЗИ от утечки по ТКУИ связано с необходимостью решения ряда проблемных вопросов построения таких систем, разработкой подсистем и алгоритмов управления мерами и средствами защиты[6], оценки эффективности ЗИ в МАСЗИ и эффективности управления ею и др., которые до настоящего времени применительно к ЗИ даже не рассматривались.

Данная статья посвящена вопросам управления защитой информации в МАСЗИ от утечки речевой информации по ТКУИ, возникающим за счет побочных электромагнитных излучений (ПЭМИ) радиоэлектронного оборудования в составе ОИ, оценки эффективности такого управления с учетом фактора времени. Конечно, рассмотреть все проблемные вопросы управления защитой в МАСЗИ в одной статье невозможно, поэтому ниже рассматриваются те из них, которые в первую очередь подлежат решению по данной проблеме.

1. Управление защитой информации от утечки по ПЭМИ с применением МАСЗИ и проблемные вопросы его реализации

Предшественниками многоагентных систем можно считать адаптивные системы, которые подстраивались под ситуацию или обстоятельства и адекватным образом меняли свое поведение или характеристики, чтобы обеспечить решение стоящих перед ними задач. Однако многоагентная система, рассматриваемая первоначально как совокупность агентов, выполняющих каждый свои функции (с адаптивными изменениями своих характеристик в ходе функционирования) без централизованного управления ими со стороны администратора системы (типа «оркестра без дирижера»), оказалась значительно сложнее просто адаптивной системы, так как система решений в ней и система управления объектами стали распределенными как по территории, так и по времени.

Агент в составе многоагентной системы — это самостоятельная программная система, «имеющая возможность принимать воздействие из внешнего мира, определяющая свою реакцию на это воздействие и формирующая ответное действие, изменяющая свое поведение с течением времени в зависимости от накопленной информации и извлеченных из нее знаний, обладающая мотивацией и способная после делегирования полномочий пользователем поставить себя на его место и принять решение, соответствующее ситуации»[7]. Создание таких агентов стало возможным за счет внедрения элементов искусственного интеллекта, однако их создание является достаточно сложной задачей.

В связи с изложенным скоро стало ясно, что такие системы с полностью децентрализованным управлением пока создать весьма сложно, а в некоторых случаях и нецелесообразно. Поэтому вполне логичным стал переход к многоагентным системам, в которых реализуется смешанный (централизованно-децентрализованный) принцип управления процессами и объектами. Это, конечно, «откат» к промежуточному варианту многоагентной системы, но он позволяет на основе имеющихся технологий создавать весьма продвинутые системы, в том числе в области ЗИ.

Применительно к МАСЗИ от утечки по ТКУИ в [3] был предложен вариант состава и структуры такой системы, однако при этом практически не затрагивались проблемные вопросы управления защитой в МАСЗИ и тем более оценки влияния такого управления на эффективность защиты.

Под управлением ЗИ на ОИ понимается совокупность целенаправленных воздействий на радиоэлектронное оборудование в составе ОИ и на средства защиты от утечки по ТКУИ, а также команд (указаний,

3 Hua, Y. Formation-containment tracking for general linear multi-agent systems with a tracking-leader of unknown control input / Y. Hua, X. Dong, L. Han, Q. Li, Z. Ren. -Текст : электронный // Systems & Control Letters, vol. 122, pp. 67–76, 2018. URL: https://www. semanticscholar.org/paper/Formation-containment-tracking-for-general-linear-a-Hua-dong/40c82ecb36b79b62925895ef33ed9fa4316fef70 (дата обращения: 15.10.2024).

4 Бежицкая Е. А., Казанцева П. И Многоагентные технологии в задачах управления // Актуальные проблемы авиации и космонавтики. 2018. Т. 2. № 4 (14). С. 289–291.

5 Городецкий В. И., Скобелев П. О. Многоагентные технологии для индустриальных приложений: реальность и перспектива // Труды СПИИРАН, № 6 (55). 2017. С. 11–45.

6 Фуртат И. Б. Адаптивное и робастное управление мультиагентными системами/ И. Б. Фуртат. – СПб: Университет ИТМО, 2016. – 155 с.

7 Бежицкая Е. А., Казанцева П. И. Многоагентные технологии в задачах управления // Актуальные проблемы авиации и космонавтики. 2018. Т. 2. № 4 (14). С. 289–291.

предписаний) подразделениям и должностным лицам на проведение организационных и организационно-технических мероприятий по решению задач ЗИ. Управление осуществляется организационными (организационно-техническими) и техническими мерами защиты. К организационным относятся меры, направленные, например, на поиск и задержание нарушителей на территории ОИ, введение ограничений на посещение ОИ или отдельных помещений и т.п. К организационно-техническим относятся меры организационного характера, реализуемые с применением технических средств, например, поиск с использованием специальной аппаратуры закладочных устройств, установка экранов и заземлений и др. К техническим мерам относятся меры технического характера, реализуемые с применением средств защиты. Цель управления состоит в своевременном (в том числе заблаговременном) применении адекватных мер защиты и достижении тем самым эффективной защиты информации от утечки по ТКУИ.

При применении организационных и организационно-технических мер защиты субъектом управления является орган управления защитой на ОИ (или уполномоченное должностное лицо), а объектом управления – выделенное для реализации мер защиты подразделение или должностные лица.

При применении технических мер защиты объектами управления являются средства защиты (средства постановки помех, экранирования, заземления и др.), а субъектами управления соответствующие интеллектуальные агенты или орган управления.

Управление в МАСЗИ включает в себя следующие действия:

- сбор необходимой для управления защитой информации, касающейся характеристик функционирующих каналов утечки информации за счет ПЭМИ (в том числе для выявления возможных датчиков информации и оценки уровней излучений ПЭМИ, по которым может перехватываться информация, для определения предполагаемого состава, характеристик и размещения технических средств перехвата, для выявления условий и характеристик среды распространения ПЭМИ и др.);
- оценку необходимости и возможности защиты перехватываемой информации и принятие решений по защите;
- выбор и организацию применения адекватных организационных (организационно-технических) и технических мер защиты с оценкой ожидаемой эффективности защиты с их применением;
- выделение, расстановку и включение (с последующим выключением) аппаратных средств защиты, настройку их параметров и контроль функционирования;
- контроль (мониторинг) эффективности защиты информации от утечки по техническим каналам.

Особенностями управления защитой в МАСЗИ являются:

- распределенность функций управления между интеллектуальными агентами системы и центральным органом управления защитой на ОИ;

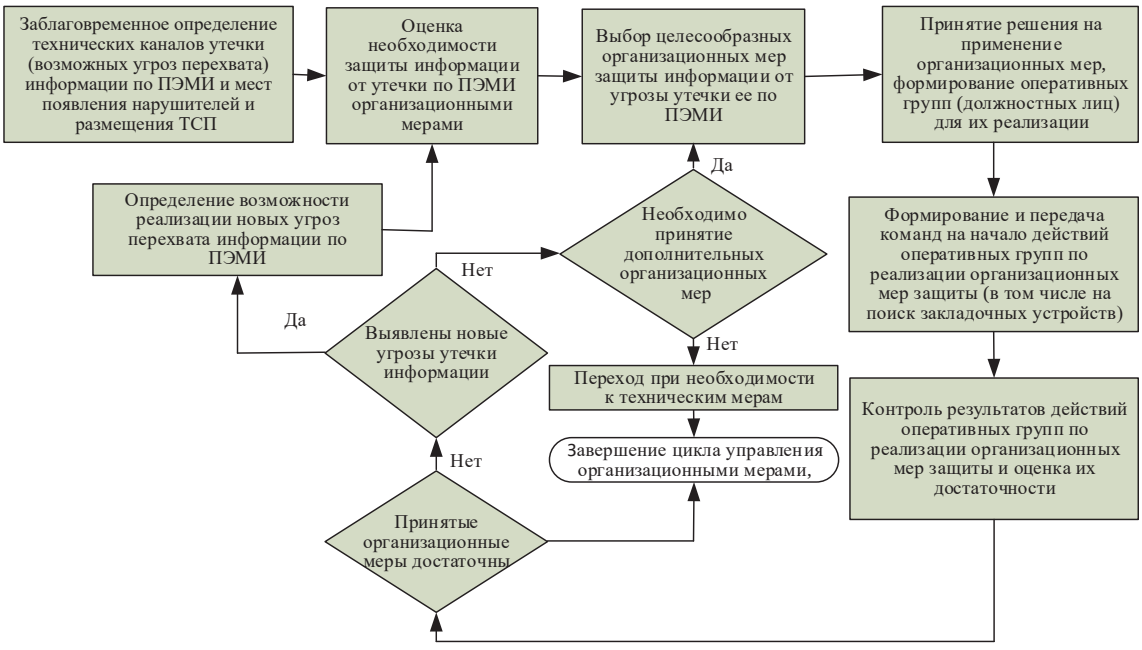


Рис. 1. Цикл управления организационными (организационно-техническими) мерами защиты информации от утечки по ПЭМИ на объекте информатизации

DOI: 10.21681/2311-3456-2024-6-85-97

- различие функций управления, реализуемых центральным органом управления при применении организационных (организационно-технических) и технических мер;
- различие функций управления при принятии заблаговременных и оперативных мер защиты;
- различие функций управления агентами, предназначенными: а) для сбора и обработки информации, необходимой для принятия мер защиты от утечки по ПЭМИ; б) для оценки возможностей и принятия решений по защите информации от утечки по ПЭМИ; г) для выявления функционирующего ТКУИ и подавления ТСП радиопомехами; д) для управления защитой от утечки по ТКУИ, содержащим различные ТСП (например, закладочные устройства, мобильные или стационарные ТСП);
- наличие случайных факторов, которые могут повлиять на управление защитой информации от утечки по ПЭМИ.

Описание циклов управления защитой информации при применении организационных и организационно-технических мер защиты применительно к каналам утечки по ПЭМИ приведено на рис. 1, а технических мер – на рис. 2.

Каждый из рассматриваемых циклов управления включает в себя две части: предварительную и непосредственную. В предварительной части управление заключается в заблаговременном анализе возможных угроз утечки информации по ПЭМИ (выявлении технических каналов утечки), которые могут быть в повседневной деятельности или при проведении различных мероприятий на ОИ (совещаний, сборов, конференций, комиссий и т.п.), в разработке модели действий нарушителя при перехвате ПЭМИ, оценке возможностей перехвата ПЭМИ различными ТСП – закладочными устройствами, мобильными (носимыми или возимыми) ТСП, стационарными средствами, которые могут устанавливаться в соседних с ОИ зданиях, в определении мер защиты, которые должны быть приняты и выполняться ежедневно при повседневной деятельности ОИ и которые могли бы быть приняты дополнительно при проведении на ОИ указанных мероприятий, в оценке ожидаемой эффективности мер защиты при повседневной деятельности на ОИ и др.

В непосредственной части управление защитой включает:

- отслеживание изменений, происходящих на ОИ и существенных для возникновения угроз утечки информации по ПЭМИ при повседневной деятельности;
- уточнение состава и оценку возможностей реализации угроз утечки информации по ПЭМИ в ходе конкретных мероприятий, проведение которых планируется на ОИ, в зависимости от их содержания и сроков проведения;
- уточнение совокупности мер защиты, которые должны применяться на ОИ при проведении очередного запланированного мероприятия на ОИ;

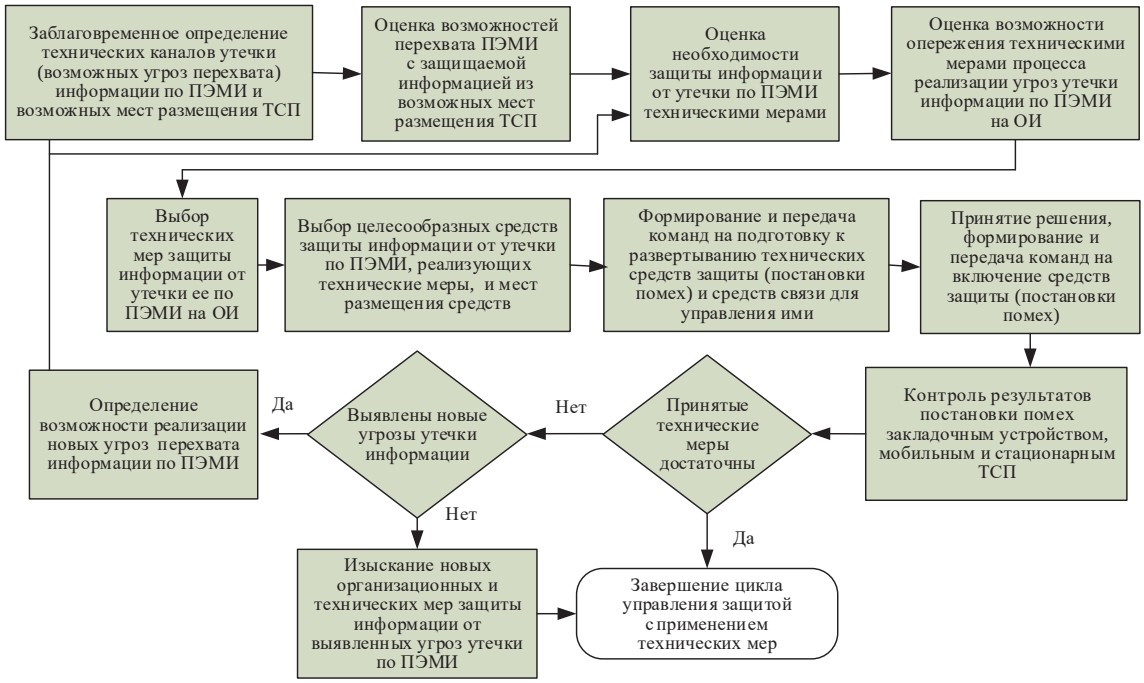


Рис. 2. Цикл управления техническими мерами защиты информации от утечки по ПЭМИ на объекте информатизации

- формирование состава сил и выбор технических средств для реализации организационных и технических мер защиты информации от утечки по ПЭМИ на очередном мероприятии, в том числе сил и средств для возможного усиления защиты при непредусмотренных изменениях условий защиты, касающихся состава и функционирования ОИ, деятельности нарушителей и функционирования ТСП;
- оценку необходимости и сроков применения активных технических средств защиты (постановки помех);
- оценку ожидаемой эффективности организационных и технических мер защиты на очередном подлежащем проведению мероприятии на ОИ;
- отслеживание изменений на ОИ (например, состава и характеристик развертываемого радиоэлектронного оборудования) в ходе проведения очередного мероприятия и в деятельности нарушителей (при наличии новых или отсутствии точных сведений о составе и размещении ТСП, о характеристиках динамики действий нарушителей на территории ОИ и за его пределами и др.);
- формирование и передачу команд на начало действий и отслеживание их выполнения выделенными оперативными группами для поиска на территории ОИ нарушителей и пресечения их деятельности, выявления и ликвидации закладочных устройств;
- формирование и передачу команд на начало применения активных средств защиты (постановки помех) и их выключения по мере надобности;
- контроль и проверка эффективности постановки помех для подавления ТСП информации по ПЭМИ.

При подготовке и реализации действий, как при предварительном, так и при непосредственном управлении защитой, имеет место целый ряд следующих подлежащих разрешению проблемных вопросов, связанных с функционированием и с созданием МАСЗИ.

1. Для создания подсистемы управления в МАСЗИ необходимо разработать линейку интеллектуальных агентов, то есть программных и программно-аппаратных средств на базе элементов искусственного интеллекта, позволяющих управлять техническими средствами защиты на ОИ от утечки по ПЭМИ, другими агентами (датчиками, средствами сбора данных, необходимых для управления, агентами обработки собираемых данных и выработки вариантов решений по ним, агентами передачи данных и т.д.).
2. Необходимо иметь средства защиты, которыми можно управлять с помощью интеллектуальных агентов, то есть включать, выключать, проводить настройку параметров (например, по частоте, мощности, направлению излучения).
3. Необходимо разработать комплекс средств для оснащения органа управления МАСЗИ, позволяющий решать всю совокупность задач, связанных с управлением защитой (например, программных средств для сбора обработки и предоставления данных, необходимых для применения мер и средств защиты, проведения расчетов при подготовке таких данных и др.).

До сих пор указанные проблемные вопросы, касающиеся практической части защиты информации от утечки по ТКУИ с использованием МАСЗИ, даже не ставились.

Функционирование МАСЗИ невозможно без соответствующего методического обеспечения управления защитой. Однако это связано с решением ряда проблемных вопросов и, в частности, касающихся понятий «эффективность защиты» и «эффективность управления защитой». Под эффективностью защиты информации принято понимать «степень соответствия результатов защиты информации поставленной цели»[8]. Эффективность характеризует меру приближения уровня защищенности информации к уровню, определенному целью защиты (чаще всего сегодня цель состоит в выполнении установленных нормативными документами требований). Однако в общем случае может быть и иная цель или несколько иерархически упорядоченных целей (рис. 3).

Так как формулирование целей защиты является прерогативой обладателя информации, то оценка эффективности ее защиты в этом смысле является субъективной (за исключением случая, когда обладателем является государство или ведомство, осуществляющее целеполагание). Многообразие возможных целей и содержания защищаемой информации обусловливает наличие проблемы создания единой методологии оценки эффективности ее защиты.

Еще более сложным оказывается проблемный вопрос оценки эффективности управления защитой. Несмотря на то, что исследования, касающиеся вопросов управления объектами, войсками, организациями и т.д. проводятся уже много десятилетий, методологические аспекты оценки его эффективности применительно к проблематике ЗИ практически не развиты, а сама оценка ограничивается использованием, в основном, качественных частных показателей, например, таких как устойчивость, непрерывность, оперативность и скрытность управления.

Попытки разработки математической модели оценки эффективности защиты информации от утечки по ПЭМИ были предприняты, например, в [2],

8 ГОСТ Р 50922 – 2006 г. Защита информации. Основные термины и определения.

Рис. 3. Иерархия возможных целей защиты информации от утечки по ПЭМИ в организации (на предприятии)

на основе применения количественных показателей. Было показано, что при таком моделировании необходимо использовать аппарат составных сетей Петри-Маркова для учета фактора времени и различных логических условий реализации угроз утечки информации по ТКУИ. Однако в этой работе даже не стоял вопрос разработки математической модели оценки эффективности управления защитой. Вместе с тем следует подчеркнуть, что отсутствие учета фактора времени делает оценку эффективности управления защитой информации несостоятельной. Однако какие-либо математические модели управления защитой от утечки по ТКУИ с учетом фактора времени сегодня отсутствуют.

Кроме этого, эффективность управления защитой с использованием количественных показателей можно оценивать, по крайней мере, двумя путями:

- во-первых, по его влиянию на эффективность защиты информации от утечки. В этом случае оценивается изменение показателя эффективности защиты (или показателя повышения защищенности информации) в результате применения мер защиты.
- во-вторых, по результативности самого управления, то есть достижения цели управления. Такими целями могут быть, например, своевременное применение адекватных мер защиты от утечки по ПЭМИ (оперативность управления) или опережение мерами защиты процесса реализации угрозы утечки информации по ПЭМИ.

Такая многоаспектность оценки должна быть учтена при разработке единого методического обеспечения оценки эффективности управления защитой. При этом при оценке по каждому из указанных путей также имеют место проблемные вопросы методологического характера, которые рассматриваются далее.

2. Система показателей оценки эффективности защиты информации от утечки по ПЭМИ в МАСЗИ и управления защитой

Наряду с тем, что оценка эффективности управления защитой и, соответственно, необходимое для этого методическое обеспечение (см. раздел 1), а в случае оценки его эффективности по влиянию на эффективность защиты также от цели самой защиты, имеет место ряд других важных факторов, подлежащих учету при такой оценке, к которым относятся следующие.

Во-первых, при централизованно-децентрализованном управлении защитой крайне важным становится взаимосвязь иерархически формируемых решений по управлению. Действительно, чтобы принять решение на включение и соответствующую настройку средства постановки помех, необходимо иметь решения органа управления, касающиеся применения средства, объекта воздействия, места размещения средства, направления или сектора постановки помехи. Тогда интеллектуальный агент принимает решения по выбору диапазона частот, времени включения и выключения (в зависимости от, например, наличия ПЭМИ с защищаемой речевой информацией по данным от агентов-датчиков), а также по выдаче команд для проведения необходимых настроек управляемого средства защиты. Сегодня система таких взаимосвязанных решений пока отсутствует. Вместе с тем от нее существенно зависит эффективность управления защитой как интеллектуальными агентами, так защитой в МАСЗИ в целом.

Во-вторых, сегодня отсутствует система показателей оценки эффективности управления защитой в МАСЗИ с использованием централизованно-децентрализованного принципа управления; и тем более математические модели для их расчета не разрабатывались.

Применительно к предварительным действиям, направленным на реализацию организационных и организационно-технических мер защиты, эффективность управления оценивается тем, насколько адекватными являются принятые меры защиты, то есть, по сути, соответствуют эффективности защиты информации на ОИ.

Применительно к техническим мерам эффективность управления может оцениваться как по влиянию на эффективность защиты, так и по достижению частной цели управления. На рис. 4 показан вариант системы таких показателей, включающей в себя интегральные и частные показатели. При этом показатель оценки эффективности управления защитой в МАСЗИ в целом (интегральный показатель) представляет собой функционал, определяемый через вероятности реализации совокупности угроз утечки информации в отсутствии и при применении организационно-технической системы управления. Для оценки функционирования элементов МАСЗИ, а также для учета различных условий и существенных факторов, влияющих на эффективность управления, могут применяться частные показатели, примеры которых показаны на рис. 4. Рассматривая вариант, когда эффективность непосредственного управления оценивается по влиянию на эффективность защиты, необходимо отметить, что в [13] предлагалось с использованием теории рисков применять в качестве интегральных показателей эффективности защиты следующие:

разностный – $\eta_d(t) = 1 - R^{(ЗИ)}(t)$; (1)

относительный – $\eta_r(t) = \frac{R^{(ЗИ)}(t)}{R^{(0)}(t)}, R^{(0)} > 0$; (2)

относительный разностный –

$$\eta_{rd}(t) = \frac{\left|R^{(0)}(t) - R^{(ЗИ)}(t)\right|}{R^{(0)}(t)}, R^{(0)} > 0, \quad (3)$$

где $R^{(ЗИ)}(t)$ и $R^{(0)}$ – риски при реализации угроз утечки информации в условиях применения и отсутствия мер защиты соответственно, при этом для u-й угрозы риск ее реализации определяется как $R_u(t) = \overline{\zeta_u} \cdot P_u(t)$, $\overline{\zeta_u}$ – математическое ожидание ущерба, наносимого при ее реализации, а $P_u(t)$ – вероятность реализации u-й угрозы. Наиболее удобным и нашедшим применение в практике анализа угроз стал относительный разностный показатель.

В условиях управления защитой вероятность парирования угрозы, как правило, ниже, чем при его отсутствии. Тогда по аналогии эффективность управления защитой от u-й угрозы может быть оценена интегральным относительным разностным показателем, который с учетом того, что меры защиты и управление ими, если не исключают возможности реализации угрозы, не влияют на возможный ущерб от ее реализации, рассчитывается по формуле:

$$\eta_{rd}^{(u)}(t) = \frac{\left|P_{u0}^{(ЗИ)}(t) - P_{uy}^{(ЗИ)}(t)\right|}{P_{u0}^{(ЗИ)}(t)}, \quad (4)$$

где $P_{u0}^{(ЗИ)}(t)$ и $P_{uy}^{(ЗИ)}(t)$ – вероятности реализации u-й угрозы в условиях применения мер защиты без управления и с управлением ими соответственно.

Если на ОИ выявлено U угроз утечки информации, реализуемых с вероятностями $P_{u0}^{(ЗИ)}(t)$ и $P_{uy}^{(ЗИ)}(t)$, и все

Рис. 4. Система показателей эффективности управления защитой в МАСЗИ по влиянию на эффективность защиты[9]

9 Здесь не учитываются показатели оценки устойчивости, непрерывности и скрытности управления, которые в общем случае также нужно количественно оценивать для учета факторов возможного негативного воздействия на систему управления со стороны нарушителя

угрозы парируются, то показатель эффективности управления рассчитывается следующим образом:

$$\eta_{rd}^{(u)}(t) = \frac{\left| \prod_{u=1}^{U} P_{u0}^{(ЗИ)}(t) - \prod_{u=1}^{U} P_{uy}^{(ЗИ)}(t) \right|}{\prod_{u=1}^{U} P_{u0}^{(ЗИ)}(t)}. \quad (5)$$

Если необходимо парировать и парируются только k угроз из U, то

$$\eta_{rd}^{(u)}(t) = \left| \frac{\frac{1}{k!} \cdot \frac{d^k}{ds^k} \left\{ \prod_{u=1}^{U} \left[1 - P_{u0}^{(ЗИ)}(t) + s \cdot P_{u0}^{(ЗИ)}(t) \right] - \prod_{u=1}^{U} \left[1 - P_{uy}^{(ЗИ)}(t) + s \cdot P_{u0}^{(ЗИ)}(t) \right] \right\}}{\frac{1}{k!} \cdot \frac{d^k}{ds^k} \left\{ \prod_{u=1}^{U} \left[1 - P_{u0}^{(ЗИ)}(t) + s \cdot P_{u0}^{(ЗИ)}(t) \right] \right\}} \right|_{s=0}. \quad (6)$$

Однако расчет такого показателя обусловливает необходимость разработки соответствующих математических моделей для оценки вероятностей реализации каждой угрозы. Ниже предлагается подход к оценке такого показателя эффективности с применением математических моделей, разрабатываемых на основе аппарата составных сетей Петри-Маркова [8, 9].

3. Математическая модель оценки эффективности управления средствами защиты информации от утечки по ПЭМИ

Рассмотрим часто встречающуюся на практике ситуацию, когда необходимо осуществить защиту речевой информации в ходе проведения некоторого мероприятия (совещания, сбора, конференции и т.п.) от утечки по ПЭМИ с применением средства постановки помех. Мероприятие проводится в ограниченный период времени, а конфиденциальная информация может быть перехвачена по ПЭМИ при эпизодическом появлении нарушителя на территории ОИ и развертывания, например, мобильного (на автомобиле) средства перехвата.

Управление средством защиты может осуществляться или органом управления МАСЗИ, или интеллектуальным агентом. И тот, и другой субъект управления, по сути, осуществляет одни и те же действия в ходе управления, при этом полагается, что вся первоначально необходимая информации для управления (решение органа управления, касающееся применения средства защиты, объект воздействия, место размещения средства защиты, направление или сектор постановки помехи) имеется. В ходе постановки помех на интеллектуальный агент в составе МАСЗИ могут поступать данные от органа управления, касающиеся появления вероятного нарушителя с мобильным средством перехвата ПЭМИ, от агентов-датчиков – возникновения ПЭМИ, по которому может перехватываться конфиденциальная информация с началом мероприятия, а также данные о диапазоне частот, в котором обнаружено ПЭМИ, виде модуляции, уровне излучения и др. В результате получения новых данных средство постановки помех перестраивается интеллектуальным агентом или непосредственно органом управления МАСЗИ. Элементы МАСЗИ, которые задействуются в ходе оперативного управления средством защиты, могут функционировать как параллельно друг другу, так и последовательно, при этом крайне важным становится учет фактора времени, без чего оценка эффективности управления оказывается несостоятельной. Как показано в [2], для моделирования таких процессов целесообразно использовать аппарат составных сетей Петри-Маркова (ССПМ) [9]. Графы ССПМ для случаев, когда отсутствует и имеется управление защитой, приведены на рис. 5[10].

Математическое ожидание времени срабатывания ССПМ при отсутствии управления мерами защиты (перемещения процесса в состояние 5) определяется из соотношения:

$$\overline{\tau_{u0}^{(ЗИ)}} = \begin{cases} \overline{\tau_{01}} + \overline{\tau_{32}} + \dfrac{\overline{\tau_{44}}}{P_{ПЭМИ}}, & \text{если нарушитель обнаруживает ПЭМИ в условиях помех с вероятностью } P_{ПЭМИ}; \\ \overline{\tau_{01}} + \overline{\tau_{33}}, & \text{если нарушитель, обнаружив помеху, меняет свое местоположение и продолжает перехват ПЭМИ без помех.} \end{cases} \quad (7)$$

где $\overline{\tau_{32}}$, $\overline{\tau_{33}}$ и $\overline{\tau_{44}}$ – математические ожидания времен перемещения процесса[11] соответственно (см. рис. 5а) по дугам (3,2z), (3, 3z), и (4,4z); $\overline{\tau_{01}}$ – математическое ожидание времени срабатывания перехода 1z,

$$\overline{\tau_{01}} = \overline{\tau_{00}} + \frac{\overline{\tau_{11}}^2 + \overline{\tau_{11}} \cdot \overline{\tau_{21}} + \overline{\tau_{11}}^2}{\overline{\tau_{11}} + \overline{\tau_{21}}}; \quad (8)$$

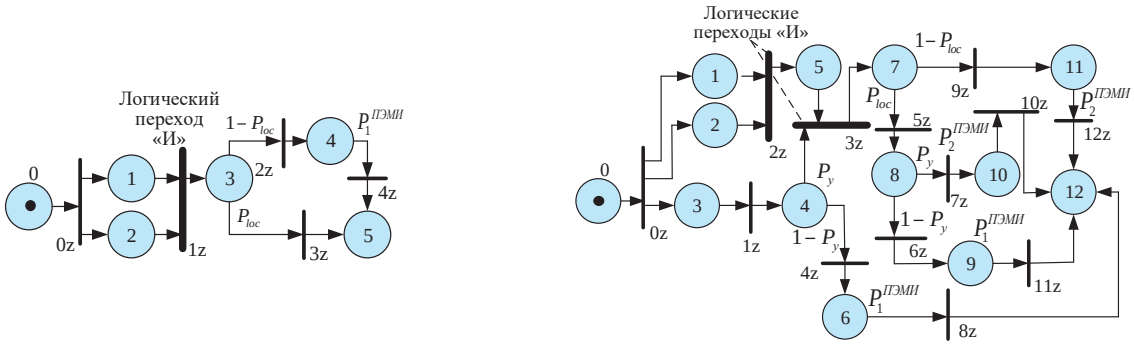
$\overline{\tau_{00}}$, $\overline{\tau_{11}}$ и $\overline{\tau_{21}}$, – математические ожидания времен перемещения процесса соответственно по дугам (0,0z), (1, 1z) и (2,1z).

Вероятность реализации угрозы в условиях отсутствия управления защитой при экспоненциальном приближении для распределений случайных времен переходов моделируемого процесса из состояний в переходы [12] определяется из соотношения:

$$P_{u0}^{(ЗИ)}(t) = (1 - P_{loc}) \cdot \left[1\text{-}\exp\left(- \frac{t}{\overline{\tau_{01}} + \overline{\tau_{32}} + \dfrac{\overline{\tau_{44}}}{P_1^{ПЭМИ}}} \right) \right] + P_{loc} \cdot \left[1 - \exp\left(- \frac{t}{\overline{\tau_{01}} + \overline{\tau_{33}}} \right) \right] \quad (9)$$

10 Время перемещения процесса из перехода в состояние считается в теории ССПМ мгновенным, а из позиции в переход конечным и случайным. Номер позиции обозначается цифрой, а перехода – цифрой с буквой z.

11 Здесь первая цифра указывает номер позиции, а вторая – номер перехода. Если какой-либо из номеров или оба номера являются двузначными, то они разделяются запятой.



а)

0 – начато мероприятие на ОИ, ожидается появление нарушителя на территории ОИ, подготовлено развертывание средства защиты, сформирована информация о возможном нарушителе, средстве перехвата и вероятном месте его развертывания;

1 – средство защиты развернуто и с началом мероприятия включено;

2 – нарушитель появился на территории ОИ, развернул средство перехвата и начал поиск ПЭМИ;

3 – нарушитель обнаружил ПЭМИ с защищаемой речевой информацией и помехи приемнику, но с вероятностью $1 - P_{loc}$ не стал менять своего местоположения, а с вероятностью P_{loc} сменил местоположение и пытается перехватить ПЭМИ в условиях, по сути, отсутствия помех;

4 – нарушитель, не меняя местоположения, начал перехват ПЭМИ в условиях помех с вероятностью $P_1^{\text{ПЭМИ}}$;

5 – угроза перехвата ПЭМИ реализована;

0z – передача команды на развертывание средства защиты и передача ориентировочной информации о нарушителе;

1z – логический переход с логикой «И», срабатываемый, если осуществлена настройка средства защиты, нарушитель предположительно появился на территории и начал поиск ПЭМИ, на ОИ начались мероприятия и возникла возможность перехвата речевой информации по ПЭМИ нарушителем;

2z – нарушитель настраивает ТСП для перехвата ПЭМИ с конфиденциальной информацией в условиях помех;

3z – нарушитель меняет свое местоположение и развертывает ТСП, перехватывает ПЭМИ с нового местоположения;

4z – с вероятностью $P_1^{\text{ПЭМИ}}$ нарушитель перехватывает ПЭМИ в условиях помех.

б)

0 – начато мероприятие на ОИ, ожидается появление нарушителя на территории ОМ, подготовлено средство защиты, собрана информация о возможном нарушителе и средстве перехвата;

1 – включены агенты-датчики для выявления нарушителя на территории с применением средств видеонаблюдения и для отслеживания появления ПЭМИ, а также интеллектуальные агенты подготовки данных, необходимых для постановки помехи;

2 – средство защиты развернуто и готово к включению;

3 – нарушитель появился на территории ОИ, развернул средство перехвата и начал поиск ПЭМИ, включена МАСЗИ;

4 – агентами-датчиками с вероятностью P_y выявлено место расположения предполагаемого нарушителя на территории ОИ, получены данные для применения средства постановки помех для передачи на интеллектуальный агент управления средством защиты (реализован цикл управления защитой) и с вероятностью $1 - P_y$ полный цикл управления защитой сорван;

5 – включено средство защиты (постановки помех);

6 – нарушитель с вероятностью $P_1^{\text{ПЭМИ}}$ перехватил ПЭМИ в условиях помех;

7 – нарушитель обнаружил ПЭМИ с защищаемой речевой информацией и наличие помех приемнику, при этом с вероятностью P_{loc} принимает решение сменить, а с вероятностью $1 - P_{loc}$ остаться на том же месте и продолжить попытки перехвата;

8 – нарушитель с вероятностью P_{loc} сменил местоположение, агентами-датчиками начат поиск нового местоположения нарушителя на территории ОИ;

9 – агентами-датчиками с вероятностью $1 - P_y$ не удалось реализовать полный цикл управления защитой, постановка помех начата по неточным данным, нарушитель начал перехват ПЭМИ с вероятностью $P_1^{\text{ПЭМИ}} \geq P_2^{\text{ПЭМИ}}$;

10 – агентами-датчиками с вероятностью P_y реализован цикл управления постановкой помех по выявленным новым данным о нарушителе, средство защиты настроено по новым данным и включено, нарушитель с вероятностью $P_2^{\text{ПЭМИ}}$ перехватил ПЭМИ в условиях помех;

11 – нарушитель, не меняя местоположения, продолжил с вероятностью $P_2^{\text{ПЭМИ}}$ перехват информации по ПЭМИ в условиях помех;

12 – угроза перехвата речевой информации по ПЭМИ в ходе проведения мероприятия на ОИ реализована;

0z – передача команды на развертывание средства защиты, формирование ориентировочной информации о нарушителе и средстве перехвата;

1z – выявление нарушителя на территории средствами наблюдения, проведение расчетов, необходимых для постановки помех, передача команды на применение средства защиты;

2z – логический переход с логикой «И», срабатывающий, если развернуто средство защиты, нарушитель появился на территории и начал поиск ПЭМИ;

3z – логический переход с логикой «И», срабатывающий, если с вероятностью P_y реализован цикл управления средством защиты и началась постановка помех;

4z и **6z** – не подготовлены с вероятностью $1 - P_y$ необходимые данные и дана команда на подавление средства перехвата помехами по ориентировочным сведениям; нарушитель перехватывает ПЭМИ с вероятностью $P_1^{\text{ПЭМИ}} \geq P_2^{\text{ПЭМИ}}$;

5z – осуществляется поиск нарушителя на территории ОИ и определение данных для постановки помех;

7z – осуществляется с вероятностью P_y поиск нарушителя и уточнение данных о нем;

8z и **11z** – осуществляется перехват ПЭМИ нарушителем с вероятностью $P_1^{\text{ПЭМИ}}$;

10z и **12z** – осуществляется перехват ПЭМИ нарушителем с вероятностью $P_2^{\text{ПЭМИ}}$.

Рис. 5. Граф составной сети Петри-Маркова для моделирования процесса реализации угрозы: а) при отсутствии; б) при наличии управления средством постановки помех для защиты информации от утечки по ПЭМИ

При наличии управления защитой (рис.5б), которое характеризуется вероятностью P_y того, что цикл управления средством защиты будет своевременно реализован (оценивающий, по сути, оперативность управления защитой), математические ожидания времени реализации угрозы по веткам графа ССПМ (0 → 4z)), (0 → 6z), (0 → 9z), (0 → 10z) определяется следующим образом:

$$\overline{\tau_{08}} = \overline{\tau_{00}} + \overline{\tau_{14}} + \overline{\tau_{44}} + \frac{\overline{\tau_{68}}}{P_1^{ПЭМИ}};$$

$$\overline{\tau_{0,10}} = \overline{\tau_{03}} + \overline{\tau_{75}} + \overline{\tau_{87}} + \frac{\overline{\tau_{10,10}}}{P_2^{ПЭМИ}};$$

$$\overline{\tau_{0,11}} = \overline{\tau_{03}} + \overline{\tau_{75}} + \overline{\tau_{86}} + \frac{\overline{\tau_{9,11}}}{P_1^{ПЭМИ}};$$

$$\overline{\tau_{0,12}} = \overline{\tau_{03}} + \overline{\tau_{79}} + \frac{\overline{\tau_{11,12}}}{P_2^{ПЭМИ}}, \quad (10)$$

где $\overline{\tau_{03}}$ – математическое ожидание времени срабатывания логического перехода 3z,

$$\overline{\tau_{03}} = \frac{(\overline{\tau_{02}}+\overline{\tau_{53}})^2+(\overline{\tau_{02}}+\overline{\tau_{53}})\cdot(\overline{\tau_{00}}+\overline{\tau_{31}}+\overline{\tau_{43}})+(\overline{\tau_{00}}+\overline{\tau_{31}}+\overline{\tau_{43}})^2}{(\overline{\tau_{02}}+\overline{\tau_{53}}+\overline{\tau_{00}}+\overline{\tau_{31}}+\overline{\tau_{43}})}, \quad (11)$$

$\overline{\tau_{02}}$ – математическое ожидание времени срабатывания логического перехода 2z,

$$\overline{\tau_{02}} = \overline{\tau_{00}} + \frac{\overline{\tau_{12}}^2 + \overline{\tau_{12}}\cdot\overline{\tau_{22}} + \overline{\tau_{22}}^2}{\overline{\tau_{12}} + \overline{\tau_{22}}}. \quad (12)$$

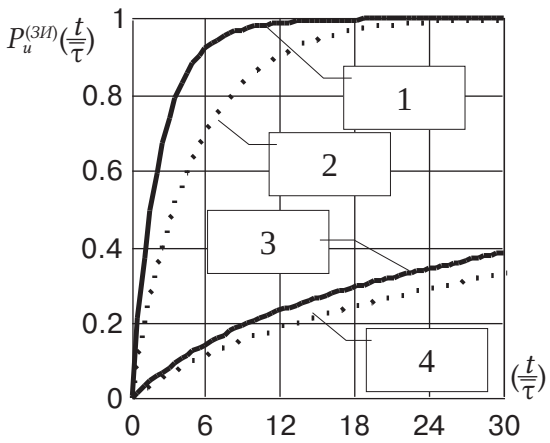
Тогда вероятность реализации угрозы находится из соотношения:

$$P_{u0}^{(ЗИ)}(t) = (1 - P_y)\cdot\left(1 - e^{\frac{t}{\overline{\tau_{08}}}}\right) + P_y\cdot\left\{P_{loc}\cdot\left[P_y\cdot\left(1 - e^{\frac{t}{\overline{\tau_{0.10}}}}\right) + (1 - P_y)\cdot\left(1 - e^{\frac{t}{\overline{\tau_{0.11}}}}\right)\right] + (1 - P_{loc})\cdot\left(1 - e^{\frac{t}{\overline{\tau_{0.12}}}}\right)\right. \quad (13)$$

В графическом виде зависимости вероятностей реализации угрозы при отсутствии и наличии управления защитой в случае, когда все математические ожидания времен перемещения процесса по сети из состояний в переходы примерно равны $\overline{\tau}$, приведены на рис. 6 и 7, а зависимость показателя эффективности управления, определяемого по формуле (4), от вероятности P_y при различных значениях показателей $P_{ПЭМИ}$ и P_{loc} – на рис. 8.

Анализ полученных зависимостей показывает следующее:

- с увеличением интервала времени вероятность реализации угрозы утечки ПЭМИ возрастает, а эффективность управления защитой падает;
- значение вероятности перемещения нарушителя по территории несущественно влияет на вероятность реализации угрозы утечки ПЭМИ из-за возможного повторного обнаружения нарушителя и постановки эффективных помех его приемнику;
- с уменьшением вероятности перехвата ПЭМИ эффективность управления возрастает;
- с уменьшением вероятности успешной реализации цикла управления защитой эффективность управления падает, однако не достигает нуля, так как остается вероятность того, что ПЭМИ в условиях фактически отсутствия управления будет перехватываться.



$P_1^{ПЭМИ} = 0{,}3;\ P_2^{ПЭМИ} = 0{,}01.$

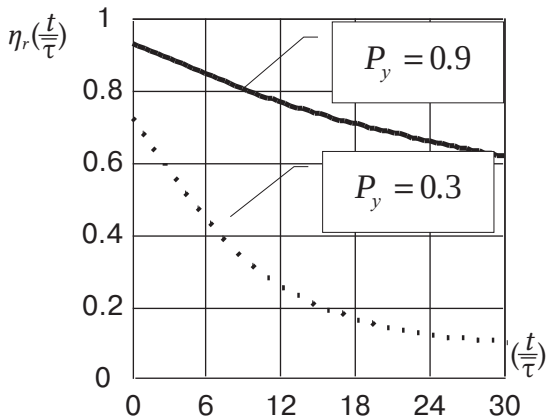
1 – управление защитой отсутствует, $P_{loc} = 0{,}9$;

2 – управление защитой отсутствует, $P_{loc} = 0{,}1$;

3 – вероятность реализации цикла управления $P_y = 0{,}9$, $P_{loc} = 0{,}9$;

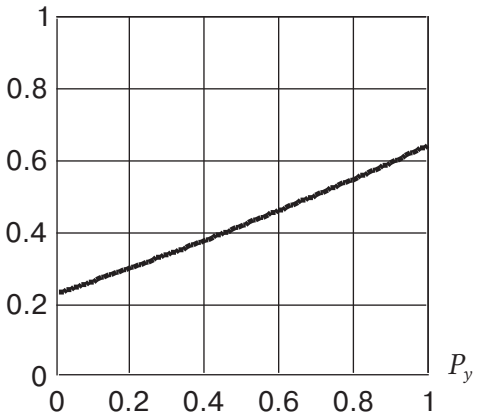
4 – вероятность реализации цикла управления $P_y = 0{,}9$, $P_{loc} = 0{,}1$;

Рис. 6. Зависимость вероятности реализации угрозы перехвата ПЭМИ от времени без управления и при наличии управления защитой



$P_{loc} = 0{,}9;\ P_1^{ПЭМИ} = 0{,}3;\ P_2^{ПЭМИ} = 0{,}01.$

Рис. 7. Зависимость показателя эффективности управления от времени при разных вероятностях реализации цикла управления



$$P_{loc} = 0{,}9;\ P_1^{ПЭМИ} = 0{,}3;\ P_2^{ПЭМИ} = 0{,}01;\ (\tfrac{t}{\overline{\tau}}) = 5.$$

Рис. 8. Зависимость показателя эффективности управления от вероятности реализации цикла управления

4. Алгоритм расчета частного показателя оперативности управления защитой информации в МАСЗИ

Введенный выше показатель оперативности управления защитой информации от утечки по ПЭМИ P_y (своевременной реализации цикла управления) представляет собой вероятность того, что случайное время τ_y, необходимое для подготовки за время $\tau_{дан}$ исходных данных, принятые за время $\tau_{реш}$ решения на применение средства защиты, подготовка и передачи за время $\tau_{ком}$ соответствующих команд, настройки и включения за время $\tau_{ср}$ средства защиты, окажется меньше случайного времени $\tau_{ТСП}$, необходимого для развертывания ТСП нарушителем за время $\tau_р$, поиска ПЭМИ и перехвата речевого сообщения за время $\tau_{пер}$ в ходе проведения мероприятия на ОИ, содержащем конфиденциальную информацию, то есть:

$$\tau_y = \tau_{дан} + \tau_{реш} + \tau_{ком} + \tau_{ср};\ \tau_{ТСП} = \tau_р + \tau_{поиск} + \tau_{пер}. \quad (14)$$

Рассмотрим случайную величину $y = \tau_{ТСП} - \tau_y$. Если $y > 0$, то цикл управления защитой будет успешно реализован. Пусть плотности распределения вероятностей для случайных величин τ_y и $\tau_{ТСП}$ равны соответственно $f_y(x)$ и $f_{ТСП}(x)$. Тогда с учетом положительно определенных значений времен плотность распределения случайной величины y, определяется следующим образом:

$$f_y(y) = \frac{1}{\overline{\tau_y} + \overline{\tau_{ТСП}}} \cdot e^{\frac{y}{\overline{\tau_{ТСП}}}} + \frac{1}{\overline{\tau_y} + \overline{\tau_{ТСП}}} \cdot \delta(y),\ y > 0\ , \quad (14)$$

где $\overline{\tau_y}$ и $\overline{\tau_{ТСП}}$ – математические ожидания времен τ_y и $\tau_{ТСП}$ соответственно; $\delta(y)$ – дельта-функция (функция Дирака).

Тогда усредненная за время t вероятность P_y определяется по формуле:

$$\overline{P_y} = \frac{\overline{\tau_{ТСП}}}{\overline{\tau_y} + \overline{\tau_{ТСП}}}. \quad (15)$$

Рассмотренный алгоритм расчета показателя оперативности управления защитой соответствует простому управлению, реализуемому или соответствующим агентом управления в МАСЗИ, или органом управления, когда не учитывается этапность процедуры (каскадность и иногда цикличность) управления, связанной с согласованием решений по управлению средством защиты между агентами и органом управления, выбором одного из нескольких возможных решений по установленным критериям, содержанием задач, решаемых в ходе управления, таких как поиск ТСП, проведение расчетов на подавление помехами ТСП, наблюдение за территорией и распознавание объектов и их действий, определение рационального размещения средств постановки помех на территории ОИ и др.

Это обусловливает необходимость разработки комплекса алгоритмов управления для обеспечения функционирования всех интеллектуальных агентов в составе МАСЗИ и соответствующих алгоритмов функционирования всех программных и программно-аппаратных средств – объектов управления. Проблемность этих вопросов определяется тем, что управление в данном случае реализуется на основе смешанного принципа управления, а решения по нему могут приниматься: а) самостоятельно каждым интеллектуальным агентом; б) согласованно несколькими агентами на одном уровне иерархии в системе управления защитой; в) путем согласования решения нижестоящего уровня с вышестоящим уровнем иерархии МАСЗИ. Каких-либо исследований по разработке соответствующих критериев и алгоритмов принятия решений при реализации смешанного принципа управления в многоагентных системах в интересах защиты информации от утечки по ТКУИ до сих пор практически не проводилось. Разработка указанных алгоритмов связана с созданием математических моделей управления, включающих в себя в качестве составных частей математические модели: сбора и обработки данных, получаемых от агентов-датчиков через подсистему связи в составе МАСЗИ с учетом фактора времени; принятие иерархически упорядоченных и согласованных оперативных решений для интеллектуальных агентов и органа управления при реализации смешанного (централизованно-децентрализованного) принципа управления.

Заключение

1. На больших ОИ, включающих десятки и сотни датчиков и аппаратных или программно-аппаратных элементов средств защиты, состав и настройки которых необходимо изменять в динамике изменения обстановки, централизованное построение и управление системой защиты из-за большого количества

процедур анализа и принятия решений по управлению может приводить к неадекватным решениям и, как следствие, к снижению эффективности защиты информации на ОИ. Парирование этих сложностей может быть достигнуто путем перехода к централизованно-децентрализованному (смешанному) принципу управления СЗИ на основе многоагентной системы защиты информации.

2. При подготовке и реализации действий, выполняемых при управлении защитой от утечки по ПЭМИ с применением МАСЗИ, имеет место целый ряд проблемных вопросов, связанных с функционированием и созданием МАСЗИ и касающихся разработки линейки интеллектуальных агентов на базе элементов искусственного интеллекта, средств защиты, которыми можно управлять с помощью интеллектуальных агентов, комплекса программных средств для оснащения органа управления МАСЗИ, позволяющего решать всю совокупность задач, связанных с управлением защитой.

3. Функционирование МАСЗИ невозможно без соответствующего методического обеспечения управления защитой. Сегодня такое обеспечение отсутствует, а его создание связано с проблемными вопросами разработки математических моделей оценки эффективности управления защитой на основе смешанного принципа управления с учетом фактора времени и различных логических условий, математических моделей управления защитой от утечки от ТКУИ, включающих в себя в том числе алгоритмы мониторинга обстановки и обработки данных, упорядоченную совокупность возможных решений, критериев и алгоритмов их принятия в МАСЗИ при выборе целесообразного состава мер защиты.

4. Для оценки эффективности управления защитой информации от утечки по ПЭМИ в МАСЗИ предложено использовать систему количественных показателей, позволяющих определить влияние управления защитой на ее эффективность. Для расчета таких показателей с учетом фактора времени и логических условий, определяющих процесс реализации угроз утечки информации по ПЭМИ целесообразно использовать аппарат составных сетей Петри-Маркова. Приведен пример применения этого аппарата для количественной оценки эффективности управления защитой информации от утечки по ПЭМИ при проведении на ОИ мероприятия (совещания, сбора или конференции).

Литература

1. *Авсентьев О. С., Кругов А. Г., Шелупанова П. А. Функциональные модели процессов реализации угроз утечки информации за счет побочных электромагнитных излучений объектов информатизации // Доклады ТУСУР. – 2020. – Т. 22, № 1. – С. 29–39.*
2. *Avsentiev O. S., Avsentiev A. O., Krugov A. G., Yazov Yu. K. Simulation of processes for protecting voice information objects against leakage through the spurious electromagnetic radiation channels using the Petri-Markov nets // Journal of Computational and Engineering Mathematics. – 2021. Vol. 8. – № 2. – P. 3–24.*
3. *Язов, Ю. К., Авсентьев А. О. Пути построения многоагентной системы защиты информации от утечки по техническим каналам // Вопросы кибербезопасности. 2022. № 5(51). С. 2–13. DOI:10.21681/2311-3456-2022-5-2-13*
4. *Wang, H. Multiagent hierarchical cognition difference policy for multiagent cooperation/ H. Wang., J. Yi., Z. Pu., Z. Liu. – Текст: электронный // Algorithms. – 2021. Т. 14. № 3. – DOI: 10.3390/a14030098*
5. *Wang, L. Distributed continuous-time containment control of heterogeneous multiagent systems with nonconvex control input constraints / Wang L., Li X., Zhang Y. – Текст : электронный // Complexity. 2022. Т. 2022. С. 7081091. – DOI: 10.1155/2022/7081091*
6. *Грушо Н. А., Тимонина Е. Е. Сравнение архитектур многоагентных систем // Информационные технологии. — Москва. — 2019. Т. 25. № 5. С. 293–299.*
7. *Кошелев Д. А., Корж Т. В. Возможность применения многоагентной системы для обнаружения внедрения и атак // Сборник трудов XXV Международной научно-технической конференции, посвященной 160-летию со дня рождения А.С. Попова: Радиолокация, навигация, связь. В 6 томах. 2019. С. 106–113.*
8. *Язов Ю. К., Соловьев С. В. Методология оценки эффективности защиты информации в информационных системах от несанкционированного доступа: монография/ Ю. К. Язов, Санкт-Петербург: Наукоемкие технологии, 2023. – 258с.*
9. *Язов Ю. К. Основы теории составных сетей Сети Петри-Маркова и их применение для моделирования процессов реализации угроз безопасности информации в информационных системах: монография / Ю. К. Язов, А. В. Анищенко, А.С . Суховерхов. – Санкт-Петербург: Сциентиа, 2024. – 196 с.*

PROBLEMATIC ISSUES OF INFORMATION PROTECTION MANAGEMENT AGAINST LEAKAGE THROUGH TECHNICAL CHANNELS USING MULTI-AGENT SYSTEMS

Yazov Yu. K.[12], Avsentiev A. O.[13]

The purpose of the article *is to reveal the problematic issues of information protection from leakage through technical channels arising from side electromagnetic radiation, using promising multi-agent systems and its management, to show the need and ways to quantify the effectiveness of such protection.*

Research methods: *methods of morphological and functional-structural analysis of the processes of distributed information security management against leakage through technical channels, as well as methods of probability theory and Petri-Markov network theory are applied in the interests of modeling and evaluating the effectiveness of centrally decentralized security management processes.*

The result obtained: *the relevance of creating a multi-agent information protection system against leakage through technical channels is shown; the need for protection management in such systems is noted, the features of a centrally decentralized (mixed) control principle in a multi-agent system are revealed by the example of protecting speech information from leakage through technical channels arising from side electromagnetic radiation of radioelectronic equipment in the-the rate of objects of informatization.*

The problematic issues of building control subsystems as part of multi-agent information protection systems against leakage through technical channels arising from side electromagnetic radiation related to the concept and formation of protection efficiency indicators, the influence of protection management on its effectiveness, and the distribution of control actions among management entities are disclosed. A composite Petri-Markov network modeling the process of leakage of speech information by side electromagnetic radiation and analytical relations for calculating the indicator of the effectiveness of information security management in a multi-agent system are presented.

The scientific novelty *of the article lies in the fact that for the first time it poses the problem of implementing a mixed principle of managing information protection from leakage through technical channels based on a multi-agent system and considers the priority methodological aspects of quantifying the effectiveness of such protection.*

Keywords: *side electromagnetic radiation, protection management, mixed control principle, protection efficiency, management efficiency, protection measure, private indicator, mathematical model.*

References

1. *Avsent'ev O. S., Krugov A. G., Shelupanova P. A. Funkcional'nye modeli processov realizacii ugroz utechki informacii za schet pobochnyh jelektromagnitnyh izluchenij ob#ektov informatizacii // Doklady TUSUR. – 2020. – T. 22, № 1. – S. 29–39.*
2. *Avsentiev O. S., Avsentiev A. O., Krugov A. G., Yazov Yu. K. Simulation of processes for protecting voice information objects against leakage through the spurious electromagnetic radiation channels using the Petri-Markov nets // Journal of Computational and Engineering Mathematics. – 2021. Vol. 8. – № 2. – P. 3–24.*
3. *Jazov, Ju. K., Avsent'ev A. O. Puti postroenija mnogoagentnoj sistemy zashhity informacii ot utechki po tehnicheskim kanalam // Voprosy kiberbezopasnosti. 2022. № 5(51). S. 2–13. DOI:10.21681/2311-3456-2022-5-2-13*
4. *Wang, H. Multiagent hierarchical cognition difference policy for multiagent cooperation/ H. Wang., J. Yi., Z. Pu., Z. Liu. – Tekst : jelektronnyj // Algorithms. — 2021. T. 14. № 3. — DOI: 10.3390/a14030098.*
5. *Wang, L. Distributed continuous-time containment control of heterogeneous multiagent systems with nonconvex control input constraints / Wang L., Li X., Zhang Y. – Tekst : jelektronnyj // Complexity. 2022. T. 2022. S. 7081091. – DOI: 10.1155/2022/7081091*
6. *Grusho N. A., Timonina E. E. Sravnenie arhitektur mnogoagentnyh sistem // Informacionnye tehnologii. – Moskva. – 2019. T. 25. № 5. S. 293–299.*
7. *Koshelev D. A., Korzh T. V. Vozmozhnost' primenenija mnogoagentnoj sistemy dlja obnaruzhenija vnedrenija i atak // Sbornik trudov XXV Mezhdunarodnoj nauchno-tehnicheskoj konferencii, posvjashhennoj 160-letiju so dnja rozhdenija A. S. Popova: Radiolokacija, navigacija, svjaz'. V 6 tomah. 2019. S. 106–113.*
8. *Jazov Ju. K., Solov'ev S. V. Metodologija ocenki jeffektivnosti zashhity informacii v informacionnyh sistemah ot nesankcionirovannogo dostupa: monografija / Ju. K. Jazov, Sankt-Peterburg: Naukoemkie tehnologii, 2023. – 258s.*
9. *Jazov Ju. K. Osnovy teorii sostavnyh setej Seti Petri-Markova i ih primenenie dlja modelirovanija processov realizacii ugroz bezopasnosti informacii v informacionnyh sistemah: monografija / Ju. K. Jazov, A. V. Anishhenko, A. S. Suhoverhov. – Sankt - Peterburg: Scientia, 2024. – 196 s.*

12 Yuri K. Yazov, Dr.Sc., Professor, Chief Researcher of the Department of the FAA «GNII PTZI FSTEC of Russia», Voronezh, Russian Federation. E-mail: Yazoff_1946@mail.ru
13 Alexander A. Gorbachev , Ph.D. Krasnodar Higher Military Orders of Zhukov and the October Revolution Red Banner School named after General of the Army S. M. Shtemenko, Krasnodar, Russia. E-mail: infosec23.00@mail.ru

АЛГЕБРАИЧЕСКИЙ АЛГОРИТМ ЭЦП С ДВУМЯ СКРЫТЫМИ ГРУППАМИ

Молдовян Н. А.[1], Петренко А. С.[2]

DOI: 10.21681/2311-3456-2024-6-98-107

Цель работы: *повышение производительности алгебраических алгоритмов ЭЦП с усиленной рандомизацией подписи.*

Метод исследования: *применение двух скрытых коммутативных групп для усиления рандомизации подписи в алгебраических алгоритмах ЭЦП на конечных некоммутативных ассоциативных алгебрах (КНАА). Известные результаты по изучению декомпозиции четырехмерных КНАА как конечных колец на множество коммутативных подколец используются для вычисления параметров алгоритма ЭЦП с двумя скрытыми коммутативными группами. Применение проверочного уравнения с многократным вхождением подгоночного элемента подписи, представляющего собой вектор S, вычисляемый по двум некоммутативным элементам из разных скрытых групп. Задание операции возведения в степень, вычисляемую как значение хеш-функции от S. В качестве алгебраического носителя алгоритма ЭЦП используются КНАА, заданные по прореженным таблицам умножения базисных векторов.*

Результаты исследования: *впервые механизм усиления рандомизации реализован в алгебраическом алгоритме ЭЦП без использования удвоения проверочного уравнения. Разработанный алгоритм ЭЦП отличается использованием двух скрытых групп для вычисления случайного вектора-фиксатора, по которому вычисляется рандомизирующий элемент генерируемой подписи. Последнее обеспечивает усиление рандомизации не только для значений подписи, но и для значений вектора фиксатора. Благодаря этому существенно повышается потенциально достижимый уровень стойкости. Достаточность выполнения проверки подлинности ЭЦП по одному проверочному уравнению обеспечивается использованием следующих двух приемов: 1) многократным вхождением подгоночного элемента подписи S в проверочное уравнение и 2) использованием значения хеш-функции, зависящего от вектора S, в качестве значения степени одной из операций экспоненциирования, выполняемой в ходе процедуры проверки подлинности подписи. Выполнен анализ стойкости к прямой атаке и к атаке на основе многих известных подписей.*

Научная и практическая значимость результатов статьи состоит в повышении производительности алгебраических алгоритмов ЭЦП с двумя скрытыми коммутативными группами, представляющими, благодаря малым размерам подписи и открытого ключа, интерес для разработки практичных постквантовых стандартов ЭЦП.

Ключевые слова: *конечная некоммутативная алгебра; ассоциативная алгебра; вычислительно трудная задача; скрытая коммутативная группа; цифровая подпись; рандомизация подписи; постквантовая криптография.*

Введение

Разработка практичных постквантовых алгоритмов электронной цифровой подписи (ЭЦП) является одной из актуальных задач в области прикладной и криптографии [1, 2]. Постквантовые криптоалгоритмы должны быть основаны на вычислительно сложных задачах, которые отличны от задачи дискретного логарифмирования (ЗДЛ) и задачи факторизации (ЗФ), поскольку для решения ЗДЛ и ЗФ на квантовом вычислителе известны полиномиальные алгоритмы[3]. Разрабатываются постквантовые двухключевые криптосхемы на группах [3], алгебраических решетках [4], кодах [5], хеш-функциях [6], труднообратимых отображениях [7,8] и некоммутативных алгебрах [9,10].

Большое внимание со стороны криптографического сообщества уделяется разработке постквантовых алгоритмов с открытым ключом на нелинейных трудно обратимых отображениях с секретной лазейкой, стойкость которых основана на вычислительной сложности решения больших систем степенных уравнений в конечных полях [11,12]. Такой интерес связан с тем, что использование квантового компьютера для нахождения решений таких систем не является эффективным. Существенным недостатком алгоритмов указанного типа, включая алгоритмы ЭЦП, является чрезвычайно большой размер открытого ключа [13, 14]. При этом обеспечивается малый размер цифровой подписи. Для устранения данного недостатка недавно была предложена концепция задания трудно обратимого отображения как операции экспоненцирования в векторных конечных полях [15, 16]. Однако и в рамках данной концепции, позволяющей уменьшить размер открытого ключа

1 Молдовян Николай Андреевич, доктор технических наук, главный научный сотрудник лаборатории проблем компьютерной безопасности Санкт-Петербургского федерального исследовательского центра Российской академии наук. ORCID: https://orcid.org/0000-0002-4483-5048. Scopus Author ID: 6603837461. E-mail: nmold@mail.ru

2 Петренко Алексей Сергеевич, младший научный сотрудник лаборатории проблем компьютерной безопасности Санкт-Петербургского федерального исследовательского центра Российской академии наук, Санкт-Петербург, Россия. ORCID: https://orcid.org/0000-0002-9954-4643. Scopus Author ID: 57200260915. E-mail: a.petrenko1999@rambler.ru

3 Yan S. Y. Quantum Computational Number Theory. – Springer. 2015. – 252 p.

в десятки раз, его размер существенно превышает этот параметр для многих других известных постквантовых алгоритмов ЭЦП.

Новый подход к построению алгоритмов ЭЦП, стойкость которых базируется на вычислительной сложности решения больших систем степенных уравнений, предложен в работах [17–20] и позволяет обеспечить малые размеры подписи и открытого ключа. В этом подходе в качестве алгебраического носителя используются конечные некоммутативные ассоциативные алгебры (КНАА), а базовой операцией в процедурах генерации и верификации подписи является операция возведения в степень большого размера. При этом открытый ключ формируется как совокупность векторов, вычисляемых по некоторому набору элементов скрытой (секретной) коммутативной группы, содержащейся в КНАА. Фактически элементы открытого ключа представляют собой замаскированные элементы скрытой группы. Маскировка выполняется путем умножения слева и справа на секретные векторы.

Характерной особенностью алгебраических алгоритмов ЭЦП [17–20] является использование подгоночного элемента подписи в виде вектора $\mathbf{S}$, вычисляемого как замаскированный случайно выбираемый элемент скрытой группы и входящего в проверочное уравнение (уравнение верификации) в качестве множителя. Эта особенность обуславливает потенциальную возможность фальсификации подписи с использованием значения $\mathbf{S}$ в качестве подгоночного параметра атаки. Для устранения такой атаки используется уравнение верификации с двукратным или многократным вхождением множителя $\mathbf{S}$, при котором решение проверочного уравнения относительно неизвестного вектора $\mathbf{S}$ является вычислительно невыполнимым.

Однако в работах [21, 22] была показана неполнота рандомизации в алгебраических алгоритмах ЭЦП со скрытой группой, обусловливающая потенциальную возможность вычисления части секретного ключа по некоторой совокупности известных подлинных подписей. Это приводит к существенному снижению уровня стойкости. Для устранения этого недостатка в работах [21, 22] предложены способы усиления рандомизации подписи, которые потребовали использования удвоенного проверочного уравнения, что существенно увеличило вычислительную сложность процедур генерации и верификации ЭЦП, т.е. привело к значительному снижению производительности алгоритмов ЭЦП со скрытой группой.

Формализация цели исследования

Рандомизация подписи в алгебраических алгоритмах ЭЦП, представленных в работах [17–20], обеспечивается выбором случайного вектора $\mathbf{H}$ из скрытой коммутативной группы и вычислением подгоночного элемента подписи $\mathbf{S}$ по формуле

$$\mathbf{S} = \mathbf{DHF}, \quad (1)$$

где $\mathbf{D}$ и $\mathbf{F}$ секретные маскирующие множители (элементы секретного ключа). С каждой подписью связано уникальное значение $\mathbf{H}$, однако, как замечено в [21], последнее выбирается из существенно ограниченного подмножества векторов, входящих в КНАА. Поскольку векторы $\mathbf{D}$ и $\mathbf{F}$ являются фиксированными, вектор $\mathbf{S}$ принимает очень малую долю возможных значений в КНАА. Коммутативное кольцо, включающее скрытую группу, может быть описано математическими формулами с числом скалярных переменных μ, существенно меньшей размерности m алгебры, используемой в качестве алгебраического носителя алгоритма ЭЦП. Значения указанных скалярных переменных являются случайными и неизвестными для каждой подписи, вычисленной владельцем открытого ключа по значениям подписываемого документа и его личного секретного ключа.

Таким образом, одна известная подпись позволяет записать m скалярных степенных уравнений, выражающих координаты вектора $\mathbf{S}$ через $2m$ фиксированных (для всех известных подписей) скалярных неизвестных (которыми являются координаты секретных векторов $\mathbf{D}$ и $\mathbf{F}$) и μ уникальных скалярных неизвестных. Для числа z известных подписей может быть составлена система, включающая mz степенных уравнений с $2m$ фиксированными неизвестными и μz уникальными скалярными неизвестными. Поскольку $\mu < m$, с увеличением значения число неизвестных (равное $2m + \mu z$) растет медленнее числа уравнений и при некотором z система будет иметь ограниченное число решений, которые могут быть найдены. Очевидно, что построенная таким образом система будет совместной для произвольного значения z. Ввиду того, что имеем дело со степенными уравнениями в общем случае будем иметь различные решения системы для произвольных значений z. При малых значениях z имеем очень большое число решений. Принимая в качестве критерия получения достаточно ограниченного числа решений равенство числа уравнений и неизвестных в системе, можно оценить требуемое число z_0 известных подписей и вычислительную сложность нахождения элементов $\mathbf{D}$ и $\mathbf{F}$ секретного ключа.

В соответствии с этим критерием имеем $mz = 2m + \mu z$, откуда получаем $z_0 = 2m / (m - \mu)$. Для четырехмерных КНАА известно их разбиение (как конечного некоммутативного кольца) на множество конечных коммутативных подколец [23, 24], из которого имеем конкретные значения $\mu = 2$ и $z_0 = 4$. Для этого случая вычислительная сложность

атаки на основе z_0 известных подписей определяется сложностью решения системы из $4z_0 = 16$ степенных уравнений в конечном поле, над которым задана КНАА, используемая в качестве алгебраического носителя. Используя оценки [25] (см. табл. 1 в [25]) сложности решения систем степенных уравнений, получаем уровень стойкости к данной атаке менее 2^{80}, что существенно меньше стойкости алгоритмов [17–20] к прямой атаке при их реализации на четырехмерных КНАА. Хотя атака на основе известных подписей не приводит к нахождению всех элементов секретного ключа, следует принять во внимание существенное снижение стойкости за счет того, что секретные векторы D и F становятся известными.

В работах [21, 22] также показано, что для оценивания полноты рандомизации следует принимать во внимание и значение случайного вектора-фиксатора $\mathbf{R}$, вычисляемого в процессе генерации подписи по формуле

$$\mathbf{R} = \mathbf{A}\mathbf{H}'\mathbf{B}, \quad (2)$$

где $\mathbf{H}'$ – случайный вектор, принадлежащий скрытой группе, $\mathbf{A}$ и $\mathbf{B}$ – секретные векторы. Действительно, значение $\mathbf{R}$ вычисляется в ходе процедуры верификации ЭЦП, поэтому формула (2) дает дополнительные уравнения, связанные с известной подписью, а также дополнительные фиксированные (координаты векторов $\mathbf{A}$ и $\mathbf{B}$) и уникальные (связанные с вектором $\mathbf{H}'$) скалярные неизвестные. В зависимости от конкретного алгоритма ЭЦП со скрытой группой для построения системы степенных уравнений, решаемой в рамках атаки на основе известных подписей, для минимизации сложности атаки может быть использована формула (1) и/или (2).

В работах [21, 22] предложены способы усиления рандомизации, требующие использования приема удвоения проверочного уравнения (по аналогии с реализацией алгоритмов [26] со скрытой группой, основанных на вычислительной сложности скрытой задачи дискретного логарифмирования), за счет чего существенно снижается производительность процедур генерации и верификации ЭЦП. В способах [21,22] векторы $\mathbf{R}$ и $\mathbf{S}$ вычисляются по формулам с использованием случайного вектора $\mathbf{V}$, что устраняет возможность использования проверочного уравнения с многократным вхождением вектора $\mathbf{S}$.

В данной работе решается задача разработки способа усиления рандомизации подписи, сохраняющего возможность использования одного проверочного уравнения с многократным вхождением вектора $\mathbf{S}$. Благодаря последнему достигается повышение производительности алгебраических алгоритмов ЭЦП с усиленной рандомизацией подписи. В основу способа положена идея использования двух скрытых коммутативных групп в четырехмерной КНАА (с глобальной двухсторонней единицей), заданной над конечным простым полем $GF(p)$, таких, что элементы одной из них не коммутируют с элементами другой, и вычисления подгоночного элемента подписи $\mathbf{S}$ по формуле

$$\mathbf{S} = \mathbf{D}\mathbf{P}^b\mathbf{H}^n\mathbf{F}, \quad (3)$$

а вектора-фиксатора $\mathbf{R}$ – по формуле

$$\mathbf{R} = \mathbf{A}\mathbf{H}^k\mathbf{P}^t\mathbf{B}, \quad (4)$$

где $\mathbf{P}$ – генератор первой скрытой (циклической) группы порядка $p^2 - 1$; $\mathbf{H}$ – генератор второй скрытой (циклической) группы простого порядка q, содержащей единственный скалярный вектор в виде двухсторонней глобальной единицы $\mathbf{E}$, используемой КНАА; $b,t < p^2 - 1$ и $n,k < q$ – случайные натуральные числа. Использование КНАА размерности $m = 4$ связано с тем, что информация о разбиении КНАА как конечного кольца на коммутативные подкольца имеет существенное значение для обоснования выбора параметров алгоритма ЭЦП, реализующего разработанный способ усиления рандомизации, и обоснования достаточности достигаемого усиления рандомизации подписи.

Вычисление векторов $\mathbf{S}$ и $\mathbf{R}$ по формулам (2) и (3) обеспечивает высокий уровень стойкости к атакам на основе известных подписей благодаря следующим утверждениям:

1) порядок поля $GF(p)$ выбирается равным $p = 2q + 1$, где q – простое число, за счет чего как множество векторов $\mathbf{P}^i\mathbf{H}^j$ (обозначим его как $\{\mathbf{P}|\mathbf{H}\}$), так и множество векторов $\mathbf{H}^j\mathbf{P}^i$ (обозначим его как $\{\mathbf{H}|\mathbf{P}\}$) при всевозможных степенях i и j включает $\approx p^3$ различных значений КНАА, используемой в качестве алгебраического носителя алгоритма ЭЦП;
2) каждое из множеств $\{\mathbf{P}|\mathbf{H}\}$ и $\{\mathbf{H}|\mathbf{P}\}$ вычислительно невозможно описать формулой с тремя скалярными переменными, из-за чего в рамках атаки на основе известных подписей атакующий вынужден рассматривать векторы $\mathbf{P}^b\mathbf{H}^n$ и $\mathbf{H}^k\mathbf{P}^t$, фигурирующие в формулах (3) и (4), как псевдослучайные векторы, каждый из которых вносит четыре уникальных скалярных неизвестных.

1. Свойства используемого алгебраического носителя

Конечная алгебра размерности m представляет собой m-мерное векторное пространство, заданное над конечным полем, с дополнительно определенной операцией векторного умножения, обладающей свойствами замкнутости и дистрибутивности слева и справа относительно операции сложения векторов. Вектор $\mathbf{V} = (v_0, v_1, v_2, v_3)$ можно представить как сумму

однокомпонентных векторов $\mathbf{V} = v_0e_0 + v_1e_1 + v_2e_2 + +v_3e_3$. Операция умножения векторов $\mathbf{A} = \sum_{i=0}^{m-1} a_i\mathbf{e}_i$ и $\mathbf{B} = \sum_{j=0}^{m-1} b_j\mathbf{e}_j$, где $\mathbf{e}_i$ – базисные векторы, может быть определена по следующей формуле:

$$\mathbf{AB} = \sum_{ji=0}^{m-1}\sum_{j=0}^{m-1} a_ib_j(\mathbf{e}_i\mathbf{e}_j), \quad (5)$$

где каждое из всевозможных произведений пар базисных векторов заменяется на однокомпонентный вектор по правилу, задаваемому некоторой таблицей умножения базисных векторов (ТУБВ). Для построения алгоритмов ЭЦП с операциями экспоненциирования в степень большого размера требуется использовать алгебры с ассоциативным умножением (это свойство позволяет применить способ быстрого возведения в степень, основанный на процессе последовательного возведения в квадрат).

В данной статье в качестве алгебраического носителя разрабатываемого алгоритма используется четырехмерная КНАА с глобальной двухсторонней единицей $\mathbf{E} = (1, 1, 0, 0)$, заданная над простым конечным полем $GF(p)$ с характеристикой в виде простого 128-битного числа вида $p = 2q + 1$, где q есть простое 127-битное число. Операция умножения четырехмерных векторов задается по прореженной ТУБВ (см. табл. 1), а именно по ТУБВ, в половине ячеек которой присутствует структурная константа с нулевым значением. Выбор КНАА размерности $m = 4$, заданной по прореженной табл. 1 связано с тем, что для выполнения одной операции векторного умножения требуется выполнить всего восемь операций умножения в поле $GF(p)$.

Таблица 1.

Прореженная ТУБВ для задания умножения четырехмерных векторов ($\lambda \neq 0$) [9]

·	$\mathbf{e}_0$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$
$\mathbf{e}_0$	$\mathbf{e}_0$	0	0	$\mathbf{e}_3$
$\mathbf{e}_1$	0	$\mathbf{e}_1$	$\mathbf{e}_2$	0
$\mathbf{e}_2$	$\mathbf{e}_2$	0	0	$\lambda\mathbf{e}_1$
$\mathbf{e}_3$	0	$\mathbf{e}_3$	$\lambda\mathbf{e}_0$	0

Декомпозиция этой КНАА как конечного некоммутативного кольца на коммутативные подкольца детально изучена в работе [9] и описывается следующим образом:

1. Множество четырехмерных векторов, содержащихся в рассматриваемой КНАА, разбивается на $\eta = p^2 + p + 1$ коммутативных подколец порядка p^2.
2. Последние пересекаются строго в множестве векторов вида $\mathbf{L} = \alpha\mathbf{E}$, где $\mathbf{E}$ – единичный вектор (глобальная двухсторонняя единица) и $\alpha \in GF(p)$.
3. Имеются только три типа коммутативных подколец порядка p^2:
   3.1. Подкольца (их число $\approx p^2/2$), изоморфные полю $GF(p^2)$ и включающие циклическую мультипликативную группу порядка $\Omega_1 = p^2 - 1$ (группу типа Γ_1).
   3.2. Подкольца (их число $\approx p^2/2$), мультипликативная группа которых (группа типа Γ_2) порождается базисом, включающим два вектора одинакового порядка $p - 1$. Порядок групп типа Γ_2 равен $\Omega_2 = (p_2 - 1)^2$.
   3.3. Подкольца (их число равно $p + 1$), мультипликативная группа которых (группа типа Γ_3) имеет циклическое строение и порядок $\Omega_3 = p(p - 1)$.
4. Произвольный фиксированный представитель $\mathbf{C} = (c_0, c_1, c_2, c_3)$ некоторого подкольца, который отличен от скалярного вектора, позволяет описать все элементы подкольца $\mathbf{V}$ с помощью формулы, включающей две скалярные переменные $d,h \in GF(p)$ и координаты представителя $\mathbf{C}$. Для подколец с мультипликативной группой типа Γ_1 и Γ_2 указанная формула имеет вид (см. формулу (8) в работе [9]):

$$\mathbf{V} = (v_0,v_1,v_2,v_3) = (d,\ d + h(c_1 - c_0)c_2^{-1},\ h,\ hc_3c_2), \quad (6)$$

Векторы $\mathbf{A}$ и $\mathbf{B}$ называются коммутативными, если $\mathbf{AB} = \mathbf{BA}$, и некоммутативными, если $\mathbf{AB} \neq \mathbf{BA}$. Докажем следующее утверждение, обосновывающее выбор векторов $\mathbf{P}$ и $\mathbf{H}$, входящих в формулы (3) и (4) как генераторы двух скрытых коммутативных групп.

Утверждение 1. Пусть в четырехмерной КНАА умножение задано по табл. 1 над полем $GF(p)$ при простом $p = 2q + 1$, где q есть простое число, и вектор $\mathbf{H}$ является генератором циклической подгруппы группы типа Γ_2, имеет порядок равный q и отличен от скалярного вектора. Тогда векторы $\mathbf{H}^x$ при $0 < x < q$ являются нескалярными векторами.

Доказательство. Предположение, что при некотором x, таком, что $0 < x < q$, $\mathbf{H}^x$ равно скалярному вектору $\mathbf{L}$, приводит к противоречию: существует натуральное число $x' = x^{-1} \bmod q$, для которого имеем $\{\mathbf{H}^x = \mathbf{L}\} \Rightarrow \{\mathbf{H} = \mathbf{L}^{x'} = \mathbf{L}'\}$, где $\mathbf{L}'$ скалярный вектор.

Утверждение 2. Пусть вектор $\mathbf{P}$ является генератором циклической группы типа Γ_1, а вектор $\mathbf{H}$ является генератором циклической подгруппы группы типа Γ_2, имеет порядок равный q, и отличен от скалярного вектора. Тогда векторы $\mathbf{P}$ и $\mathbf{H}$ некоммутативны и каждое из произведений $\mathbf{P}^i\mathbf{H}^j$ и $\mathbf{H}^j\mathbf{P}^i$ при $i = 1, 2, \ldots p^2 - 1$ и $j = 1, 2, \ldots q$ принимает $(p^2 - 1)q$ различных значений в КНАА, в которой умножение задано по табл. 1 над полем $GF(p)$ при простом $p = 2q + 1$, где q есть простое число.

Доказательство. Скалярный вектор имеет порядок, равный делителю числа $p - 1$, поэтому не может быть генератором циклической группы порядка $p^2 - 1$, т. е. P является нескалярным вектором. Поэтому векторы $\mathbf{P}$ и $\mathbf{H}$ некоммутативны как несклярные векторы, принадлежащие разным коммутативным подкольцам порядка p^2. Пусть при некоторых целых неотрицательных числах $i,k < p^2 - 1$ и $j,t < q$ имеем $\mathbf{P}^i\mathbf{H}^j = \mathbf{P}^k\mathbf{H}^t$. Тогда $\{\mathbf{P}^i\mathbf{H}^j = \mathbf{P}^k\mathbf{H}^t\} \Rightarrow \{\mathbf{P}^{i-k} = \mathbf{H}^{j-t}\}$. Поскольку $\mathbf{H}$ является несклярным вектором, то все его степени $\mathbf{H}^x$ при $0 < x < q$ являются нескалярными векторами (см. утверждение 1), а пересечение различных подколец рассматриваемой КНАА имеет место только в множестве скалярных векторов, равенство $\mathbf{P}^{i-k} = \mathbf{H}^{j-t}$ возможно только в случае $\mathbf{H}^{j-t} = \mathbf{E}$, из чего следует $\mathbf{P}^{i-k} = \mathbf{E}$. Из последних двух равенств имеем $\{j \equiv t \bmod q\} \Rightarrow \{j = t\}$ и $\{i \equiv k \bmod (p^2 - 1)\} \Rightarrow$ $\Rightarrow \{i = k\}$. Таким образом, каждая уникальная пара значений (i, j) задает уникальный вектор $\mathbf{P}^i\mathbf{H}^j$, т. е. число последних равно $(p^2 - 1)q$. Аналогично доказывается, что вектор $\mathbf{H}^j\mathbf{P}^i$ также принимает $(p^2 - 1)q$ разных значений.

2. Оценка стойкости к атакам на основе известных подписей

Формулы (3) и (4) описывают разработанный способ рандомизации подписи и определяют стойкость реализующих его конкретных алгебраических алгоритмов ЭЦП к атаке на основе известных подписей. Дадим общую оценку достигаемого уровня стойкости к указанной атаке в случае использования четырехмерных КНАА в качестве алгебраического носителя. Следует рассмотреть случаи использования формул (3) и/или (4) для составления системы уравнений, из которой вычисляются элементы секретного ключа.

Случай использования формулы (3). При наличии z известных подписей с подгоночными элементами $\mathbf{S}_i = \mathbf{D}\mathbf{P}_i\mathbf{H}_i\mathbf{F}$ (где $i = 1,2,\ldots z$; $\mathbf{P}_i$ и $\mathbf{H}_i$ – случайные некоммутативные векторы, выбираемые из двух разных скрытых групп) имеем систему из $4z$ скалярных степенных уравнений в поле $GF(p)$, составленных для координат векторов $\mathbf{S}_i$. Неизвестные секретные векторы $\mathbf{D}$ и $\mathbf{F}$ задают 8 фиксированных скалярных неизвестных (которыми являются координаты $\mathbf{D}$ и $\mathbf{F}$). Сделаем сильное предположение в пользу атакующего, состоящее в том, что он нашел способ описания множества векторов $\{\mathbf{P}|\mathbf{H}\}$ мощности $\approx p^3$ по фиксированным координатам некоторого представителя $\mathbf{C}$ этого множествами и тройку скалярных значений $t,v,u \in GF(p)$. Поскольку скрытые группы являются секретными, то координаты вектора $\mathbf{C}$ являются неизвестными, т. е. имеем еще четыре фиксированных неизвестных. При этом каждое уравнение в системе вносит $\mu = 3$ уникальных скалярных неизвестных. Таким образом, имеем 12 фиксированных неизвестных и $3z$ уникальных. По критерию равенства числа неизвестных и числа уравнений составляем выражение

$$4z = 12 + 3z, \qquad (7)$$

из которого получаем нужное для выполнения атаки число известных подписей $z_0 = 12$. Это значение z_0 соответствует 48 степенным уравнениям, входящим в решаемую в ходе атаки систему, и уровню стойкости к данной атаке $>2^{128}$ (см. табл. 1 в [25]).

Сделанное в пользу атакующего допущение является очень сильным. На самом деле описание выбора случайного вектора из множества $\{\mathbf{P}|\mathbf{H}\}$ через случайные три скалярные неизвестные t, v и u, видимо, является вычислительно нереализуемым, поскольку в общем случае произведение $\mathbf{P}^b\mathbf{H}^n$ со случайными степенями b и n в общем случае генерируют векторы, принадлежащие разным подкольцам порядка p^2, что потребует рассмотрения четырех уникальных скалярных неизвестных, связанных с каждой подписью. В этом случае число неизвестных в решаемой системе будет превосходить число уравнений, т.е., если будет возможным, приемлемо ограниченное число решений может быть получено для числа известных подписей $z_0 >> 12$. Последнее приводит к оценке уровня стойкости $> 2^{256}$.

Случай использования формулы (4). Полностью аналогичен случаю составления решаемой системы степенных уравнений по формуле (3), включая приведенные значения уровня стойкости.

Случай использования формул (3) *и* (4). С известными подписями связаны векторы $\mathbf{S}_i = \mathbf{D}\mathbf{P}_i\mathbf{H}_i\mathbf{F}$ и вычисляемые по проверочному уравнению векторы $\mathbf{R}_i = \mathbf{A}\mathbf{H}'_i\mathbf{P}'_i\mathbf{B}$ (где $i = 1,2,\ldots z$; $\mathbf{P}_i\mathbf{H}_i$ – случайные векторы, выбираемые множества $\{\mathbf{P}|\mathbf{H}\}$; $\mathbf{H}'_i\mathbf{P}'_i$ – случайные векторы, выбираемые множества $\{\mathbf{H}|\mathbf{P}\}$), по которым составляется система из $8z$ скалярных степенных уравнений, составленных для координат векторов $\mathbf{S}_i$ и $\mathbf{R}_i$. Неизвестные секретные векторы $\mathbf{A}$, $\mathbf{B}$, $\mathbf{D}$ и $\mathbf{F}$ задают 16 фиксированных неизвестных (координаты этих векторов). Расширяя указанное выше предположение в пользу атакующего на выбор случайного вектора из множества $\{\mathbf{H}|\mathbf{P}\}$ мощности $\approx p^3$ по фиксированным координатам некоторого представителя $\mathbf{C}'$ этого множества и тройку скалярных значений t', v', $u' \in GF(p)$, получаем восемь дополнительных фиксированных неизвестных (координаты векторов $\mathbf{C}$ и $\mathbf{C}'$) и $\mu = 6$ уникальных скалярных неизвестных. Таким образом, имеем 24 фиксированных неизвестных и $6z$ уникальных. В соответствии с критерием равенства числа неизвестных и числа уравнений имеем соотношение $8z = 24 + 6z$, из которого вычисляем $z_0 = 12$, что прямолинейно соответствует 96 степенным уравнениям, входящим в систему, и уровню стойкости к рассматриваемой атаке $>2^{256}$ (см. табл. 1 в [25]).

Однако, легко заметить, что «по построению» полученная система распадается на две независимые системы по 48 уравнений в каждой, причем последние полностью идентичны системам, возникающим в случаях проведения атаки с использованием только формулы (3) или (4).

Таким образом, рассмотренные варианты атаки на основе известных подписей имеют вычислительную сложность не менее 2^{128} в модели атаки с сильным допущением в пользу атакующего. В случае атаки без такого допущения ожидаемый уровень стойкости к данной атаке составляет не менее 2^{256}.

3. Постквантовый алгоритм ЭЦП

При генерации элементов секретного ключа в разработанном алгебраическом алгоритме ЭЦП используется следующее условие обратимости четырехмерного вектора $\mathbf{V} = (v_0, v_1, v_2, v_3)$ как элемента используемой в качестве алгебраического носителя КНАА, задаваемое используемой ТУБВ [9]:

$$v_0, v_1 \neq \lambda v_2 v_3 \qquad (8)$$

Формирование секретного ключа выполняется как генерация случайных натуральных чисел $x < p - 1$, $u < p - 1$ и $w < p - 1$ и случайных обратимых векторов $\mathbf{A}$, $\mathbf{B}$, $\mathbf{D}$, $\mathbf{F}$, $\mathbf{H}$ и $\mathbf{P}$, которые попарно некоммутативны (с учетом строения используемой КНАА вероятность того, что пять случайных векторов будут обратимы и попарно некоммутативны, близка к единице). При этом вектор $\mathbf{H}$ является нескалярным, имеет порядок, равный 127-битному простому числу q, и принадлежит подкольцу, содержащему мультипликативную группу типа Γ_2, а вектор $\mathbf{P}$ имеет порядок $p^2 - 1$. Легко показать, что случайный вектор $\mathbf{H}$ (вектор $\mathbf{P}$) удовлетворяет указанным условиям с вероятностью ≈0,25 (≈0,1), а размер секретного ключа составляет ≈370 байт порядка.

Открытый ключ вычисляется по секретному в виде набора из восьми векторов $\mathbf{Y}$, $\mathbf{N}$, $\mathbf{Z}$, $\mathbf{T}$, $\mathbf{V}$, $\mathbf{X}$, $\mathbf{K}$ и $\mathbf{U}$ (с общим размером 512 байт) по следующим формулам:

$$\mathbf{Y} = \mathbf{AHA}^{-1};\ \mathbf{N} = \mathbf{AH}^{u}\mathbf{P}^{wx}\mathbf{D}^{-1};$$
$$\mathbf{Z} = \mathbf{DPD}^{-1};\ \mathbf{T} = \mathbf{F}^{-1}\mathbf{H}^{x}\mathbf{A}^{-1}; \qquad (9)$$

$$\mathbf{V} = \mathbf{AH}^{w}\mathbf{P}^{x}\mathbf{D};\ \mathbf{X} = \mathbf{F}^{-1}\mathbf{H}^{u}\mathbf{F};\ \mathbf{K} = \mathbf{FH}^{x}\mathbf{P}^{u}\mathbf{B};\ \mathbf{U} = \mathbf{B}^{-1}\mathbf{PB}. \qquad (10)$$

Предполагается, что при генерации и верификации подписи используется некоторая коллизионно стойкая 256-битная хеш-функция Φ, которая является частью рассматриваемой постквантовой схемы ЭЦП.

Алгоритм генерации ЭЦП.

Процедура генерации ЭЦП к документу M включает следующие шаги:

1. Сгенерировать случайные натуральные числа $k < q$ и $t < p^2 - 1$ и вычислить значение вектора-фиксатора $\mathbf{R}$ по формуле (4): $\mathbf{R} = \mathbf{AH}^{k}\mathbf{P}^{t}\mathbf{B}$.
2. Вычислить хеш-значение от документа M с присоединенными к нему рандомизирующим вектором $\mathbf{R}$: $e = e_1 || e_2 = \Phi(M, \mathbf{R})$, где 256-битное хеш-значение e представлено в виде конкатенации двух 128-битных натуральных чисел e^1 и e^2.
3. Вычислить натуральную степень b: $b = -(wx + e) \bmod (p^2 - 1)$.
4. Вычислить натуральную степень n: $n = -(ue_2 + x) \bmod q$.
5. По формуле (3) вычислить вектор $\mathbf{S}$ (подгоночный элемент генерируемой подписи): $\mathbf{S} = \mathbf{DP}^{b}\mathbf{H}^{n}\mathbf{F}$.
6. Вычислить вспомогательный рандомизирующий элемент ЭЦП ρ по формуле $\rho = \Phi(\mathbf{S})$.
7. Вычислить первый вспомогательный подгоночный элемент ЭЦП в виде натурального 127-битного числа s по формуле $s = e_1^{-1}k - u - n - x - e_1^{-1}w \bmod q$.
8. Вычислить второй вспомогательный подгоночный элемент ЭЦП в виде натурального 256-битного числа σ по формуле $\sigma = t - x - \rho - b - u \bmod (p^2 - 1)$.

Сгенерированная ЭП к документу M представляет собой четверку значений

$(e, s, \sigma, \mathbf{S})$ с общим размером ≈144 байт. Вычислительную сложность процедуры вычисления ЭЦП можно оценить как две операции возведения четырехмерных векторов в 256-битную степень ($\mathbf{P}^t$ и $\mathbf{P}^b$) и две операции возведения в 128-битную степень ($\mathbf{H}^k$ и $\mathbf{H}^n$), т. е. как ≈9200 операций умножения в поле $GF(p)$.

Алгоритм верификации ЭЦП.

Проверка подлинности подписи $(e, s, \sigma, \mathbf{S})$ к документу M осуществляется с использованием 512-байтного открытого ключа $(\mathbf{Y}, \mathbf{N}, \mathbf{Z}, \mathbf{T}, \mathbf{V}, \mathbf{X}, \mathbf{K}, \mathbf{U})$ по следующему алгоритму:

1. Вычислить 256-битное натуральное число ρ:$\rho = \Phi(\mathbf{S})$.
2. Вычислить вектор $\mathbf{R}'$ по следующей формуле (проверочное уравнение):

$$\mathbf{R}' = (\mathbf{Y}^{s}\mathbf{NZ}^{e}\mathbf{ST})^{e_1}\mathbf{VZ}^{\rho}\mathbf{SX}^{e_2}\mathbf{KU}^{\sigma}. \qquad (11)$$

3. Вычислить хеш-функцию от документа M с присоединенным к нему вектором $\mathbf{R}'$: $\varepsilon_1 || \varepsilon_2 = \Phi(M, \mathbf{R}')$, где 256-битное хеш-значение представлено в виде конкатенации двух 128-битных чисел ε_1 и ε_2.
4. Если одновременно выполняются равенства $\varepsilon_1 = e_1$ и $\varepsilon_2 = e_2$, то подпись принимается как подлинная, иначе она отвергается как ложная.

Вычислительную сложность алгоритма верификации подписи можно оценить как три операции возведения четырехмерных векторов в 256-битную степень ($\mathbf{Z}^{e}$, $\mathbf{Z}^{\rho}$ и $\mathbf{U}^{\sigma}$) и три операции возведения четырехмерных векторов в 128-битную степень ($\mathbf{Y}^{s}$, $(\mathbf{Y}^{s}\mathbf{NZ}^{e}\mathbf{ST})^{e_1}$ и $\mathbf{X}^{e_2}$), для чего надо осуществить

≈13800 операций умножения в поле $GF(p)$. Подставляя в проверочное уравнение (11) элементы открытого ключа, выраженные через элементы секретного ключа, легко доказать корректность работы предложенного алгоритма ЭЦП.

Доказательство корректности схемы ЭЦП.

Подставляя в проверочное уравнение (11) элементы открытого ключа, выраженные через элементы секретного ключа по формулам (9) и (10), для корректно сгенерированной подписи получаем:

$$\mathbf{R}' = \left[\left(\mathbf{AHA}^{-1}\right)^{y}\left(\mathbf{AH}^{u}\mathbf{P}^{wx}\mathbf{D}^{-1}\right)^{x}\left(\mathbf{DPD}^{-1}\right)^{w}\left(\mathbf{DP}^{b}\mathbf{H}^{n}\mathbf{F}^{-1}\right)^{e}\left(\mathbf{FH}^{x}\mathbf{A}^{-1}\right)^{s}\right]^{e_1}\mathbf{AH}^{w}\mathbf{P}^{x}\mathbf{D}^{-1}\left(\mathbf{DPD}^{-1}\right)^{\rho}\times$$
$$\times\left(\mathbf{DP}^{b}\mathbf{H}^{n}\mathbf{F}^{-1}\right)\left(\mathbf{F}^{-1}\mathbf{H}^{x}\mathbf{P}^{u}\mathbf{B}\right)\left(\mathbf{B}^{-1}\mathbf{PB}\right)^{\sigma} =$$
$$= \left(\mathbf{AH}^{s+u}\mathbf{P}^{wx+e+b}\mathbf{H}^{n+x}\mathbf{A}^{-1}\right)^{e_1}\mathbf{AH}^{w}\mathbf{P}^{x+\rho+b}\mathbf{H}^{n+we_2+x}\mathbf{P}^{u+\sigma}\mathbf{B} =$$
$$= \left(\mathbf{AH}^{s+u}\mathbf{P}^{0}\mathbf{H}^{n+x}\mathbf{A}^{-1}\right)^{e_1}\mathbf{AH}^{w}\mathbf{P}^{x+\rho+b}\mathbf{H}^{0}\mathbf{P}^{u+\sigma}\mathbf{B} = \left(\mathbf{AH}^{s+u+n+x}\mathbf{A}^{-1}\right)^{e_1}\mathbf{AH}^{w}\mathbf{P}^{x+\rho+b+u+\sigma}\mathbf{B} =$$
$$= \mathbf{AH}^{(s+u+n+x)e_1+w}\mathbf{P}^{x+\rho+b+we_2+t-x-\rho-b-we_2}\mathbf{B} = \mathbf{AH}^{\left(e_1^{-1}k-u-n-x-se_1^{-1}w+u+n+x\right)e_1^{-1}+w}\mathbf{P}'\mathbf{B} =$$
$$= \mathbf{AH}^{k}\mathbf{P}'\mathbf{B} = \mathbf{R}.$$

С учетом равенства $\mathbf{R} = \mathbf{R}'$ имеем $\varepsilon_1 || \varepsilon_2 = \Phi(M, \mathbf{R}') = \Phi(M, \mathbf{R}) = e_1 || e_2$, т.е. корректно сгенерированная подпись проходит процедуру верификации как подлинная подпись, что означает корректность разработанного алгоритма ЭЦП.

4. Обсуждение

В основе стойкости разработанного алгоритма ЭЦП является вычислительная сложность нахождения решений большой системы степенных уравнений, определяемых формулами (9) и (10) для вычисления элементов открытого ключа по элементам секретного ключа (являющихся неизвестными). Соответствующую систему векторных уравнений можно представить в виде следующего набора уравнений второй, третьей и четвертой степеней, решаемых совместно:

$$\mathbf{YA} = \mathbf{AH};\ \mathbf{N} = \mathbf{AH}^{u}\mathbf{P}^{wx}\mathbf{D}^{-1};\ \mathbf{ZD} = \mathbf{DP};\ \mathbf{FTA} = \mathbf{H}^{x}; \quad (12)$$

$$\mathbf{V} = \mathbf{AH}^{w}\mathbf{P}^{x}\mathbf{D};\ \mathbf{FX} = \mathbf{H}^{u}\mathbf{F};\ \mathbf{K} = \mathbf{FH}^{x}\mathbf{P}^{u}\mathbf{B};\ \mathbf{BU} = \mathbf{PB}. \quad (13)$$

При решении такой системы векторных степенных уравнений следует определиться с неизвестными 128-битными значениями x, u и w. Если считать их неизвестными, то наша система уже будет системой экспоненциальных уравнений, сложность решения которой представляется существенно большей, чем сложность системы степенных векторных уравнений, в которых векторы $\mathbf{H}_u = \mathbf{H}^u$, $\mathbf{H}_x = \mathbf{H}^x$ и $\mathbf{H}_w = \mathbf{H}^w$ рассматриваются как неизвестные значения, коммутативные с неизвестной $\mathbf{H}$, а векторы $\mathbf{P}_{wx} = \mathbf{P}^{wx}$, $\mathbf{P}_x = \mathbf{P}^x$ и $\mathbf{P}_u = \mathbf{P}^u$ – как неизвестные значения, коммутативные с неизвестной $\mathbf{P}$. Учет условия коммутативности приводит к добавлению в систему следующих 6 уравнений, описывающих коммутативность неизвестных векторов, выбираемых из одной и той же скрытой коммутативной группы (уравнения проверки коммутативности):

$$\mathbf{HH}_u = \mathbf{H}_u\mathbf{H};\ \mathbf{HH}_x = \mathbf{H}_x\mathbf{H};\ \mathbf{HH}_w = \mathbf{H}_w\mathbf{H}; \quad (14)$$

$$\mathbf{PP}_{wx} = \mathbf{P}_{wx}\mathbf{P};\ \mathbf{PP}_x = \mathbf{P}_x\mathbf{P};\ \mathbf{PP}_u = \mathbf{P}_u\mathbf{P}. \quad (15)$$

С учетом возможности представления каждого из неизвестных векторов $\mathbf{H}_u$, $\mathbf{H}_x$ и $\mathbf{H}_w$ через координаты вектора $\mathbf{H}$ и пару скалярных неизвестных (d_u, h_u), (d_x, h_x) и (d_w, h_w) соответственно, а неизвестных $\mathbf{P}_{wx}$, $\mathbf{P}_x$ и $\mathbf{P}_u$ через координаты вектора $\mathbf{P}$ и пару скалярных неизвестных (d'_w, h'_w), (d'_x, h'_x) и (d'_u, h'_u) соответственно (см. формулу (6)) при сведении решения рассматриваемой системы векторных уравнений к решению системы степенных скалярных уравнений уравнения (14) и (15) автоматически учитываются при использовании указанного представления. В результате получим 32 скалярных степенных уравнения с 36 скалярными неизвестными. При этом степень некоторых уравнений увеличивается, но это несущественно изменяет вычислительную сложность решения больших систем степенных уравнений, которая наиболее сильно зависит от числа уравнений (по сравнению со степенью уравнений и порядком поля, в котором задается система) [11, 25].

С учетом 128-битного порядка поля $GF(p)$, в котором задаются степенные уравнения, сложность решения системы из 32 уравнений, т.е. стойкость разработанного алгоритма ЭЦП к прямой атаке, можно оценить как $>2^{100}$ (см. табл. 1 в [25]). Поскольку сложность атаки на основе известных подписей существенно превышает последнее значение, можно сделать вывод о достаточности рандомизации подписи в описанном алгоритме.

Для повышения стойкости разработанного алгоритма ЭЦП в качестве его алгебраического носителя следует использовать КНАА размерности $m > 4$. Для выполнения оценки стойкости реализаций описанного алгоритма ЭЦП на КНАА с размерностями $m = 6, 8, 10, 12$ к атакам на основе известных подписей требуется предварительно изучить их декомпозицию как некоммутативных конечных колец на коммутативные подкольца. Оценка стойкости к прямым атакам может быть дана в предположении, что для всех указанных значений размерности при прямой атаке можно устранить уравнения проверки коммутативности (14) и (15) из решаемой системы степенных уравнений, число которых в этом случае становится равным $8m$. Обоснование этого предположения связано с тем, что элементы коммутативных подколец вычисляются из векторного уравнения вида $\mathbf{AX} = \mathbf{XA}$ при фиксированном векторе $\mathbf{A}$, решение которого сводится к решению системы из m линейных уравнений. Ранг γ главного определителя последней меньше значения m и коммутативное подкольцо, включающее вектор $\mathbf{A}$, описывается как линейное пространство решений размерности $m - \gamma$, а выбор неизвестного вектора из заданной скрытой коммутативной группы может быть описан через $m - \gamma$ скалярные неизвестные. В табл. 2

Таблица 2.

Ожидаемый уровень стойкости к прямой атаке для различных значений размерности

Размерность	4	6	8	10	12
Число степенных уравнений в системе	32	48	64	80	96
Уровень стойкости к прямой атаке	$\approx 2^{100}$	$>2^{128}$	$\approx 2^{192}$	$>2^{192}$	$\approx 2^{256}$

приведены ожидаемые оценки стойкости к прямой атаке для случая использования в качестве алгебраического носителя КНАА различных размерностей m (получение оценок стойкости к атаке на основе известных подписей требует знания конкретных значений ранга γ).

Другим аспектом, связанным с использованием КНАА с размерностями $m = 6, 8, 10, 12$, является существенное увеличение числа операций умножения в поле $GF(p)$, выполняемых в ходе процедур генерации и верификации ЭЦП. Снижение производительности алгоритма можно уменьшить, выбирая меньшие размеры порядка поля $GF(p)$ при переходе к большим значениям размерности m. Однако, такой способ также должен учитывать строение используемых КНАА (с точки зрения декомпозиции на коммутативные подкольца).

Выводы

Предложен способ усиления рандомизации подписи в алгебраических алгоритмах ЭЦП, отличающийся от известных аналогов вычислением вектора-фиксатора в зависимости от взаимно некоммутативных генераторов двух скрытых коммутативных групп и обеспечивающий возможность построения алгоритмов с одним уравнением верификации. Благодаря последнему обеспечивается повышение производительности алгоритма. Разработанный на основе способа алгоритм использует в качестве алгебраического носителя четырехмерную КНАА, заданную по прореженной ТУБВ, и обладает уровнем стойкости $>2^{100}$.

Показана потенциальная возможность повышения стойкости до уровня 2^{256} при реализации разработанного алгоритма на КНАА размерности $m = 6, 8, 10, 12$. Однако конкретная реализация таких вариантов алгоритма связана с задачей детального изучения строения КНАА указанных размерностей, что составляет задачу дальнейших исследований, направленных на разработку алгебраического алгоритма, представляющего интерес в качестве основы практичного постквантового стандарта ЭЦП.

Исследование выполнено за счет гранта Российского научного фонда № 24-41-04006, https://rscf.ru/project/24-41-04006/

Литература

1. *Post-Quantum Cryptography. 15th International Conference, PQCrypto 2024, Oxford, UK, June 12–14, 2024, Proceedings // Lecture Notes in Computer Science. 2024. V. 14771–14772. Springer, Cham.*
2. *Post-Quantum Cryptography. 14th International Conference, PQCrypto 2023, College Park, MD, USA, August 16–18, 2023, Proceedings // Lecture Notes in Computer Science. 2023. V. 14154. Springer, Cham.*
3. *Battarbee C., Kahrobaei D., Perret L., Shahandashti S.F. SPDH-Sign: Towards Efficient, Post-quantum Group-Based Signatures // In: Johansson, T., Smith-Tone, D. (eds) Post-Quantum Cryptography. PQCrypto 2023 / Lecture Notes in Computer Science, 2023. V. 14154. P. 113–138. Springer, Cham. https://doi.org/10.1007/978-3-031-40003-2_5*
4. *Gärtner J. NTWE: A Natural Combination of NTRU and LWE // In: Johansson, T., Smith-Tone, D. (eds) Post-Quantum Cryptography. PQCrypto 2023 / Lecture Notes in Computer Science, 2023, vol 14154, pp. 321–353. Springer, Cham. https://doi.org/10.1007/978-3-031-40003-2_12*
5. *Alamelou Q., Blazy O., Cauchie S., Gaborit Ph. A code-based group signature scheme // Designs, Codes and Cryptography. 2017. V. 82. N. 1–2. P. 469–493.*
6. *Hamlin B., Song F. Quantum Security of Hash Functions and Property-Preservation of Iterated Hashing // In: Ding, J., Steinwandt, R. (eds) Post-Quantum Cryptography. PQCrypto 2019 / Lecture Notes in Computer Science. 2019. V. 11505. P. 329–349. Springer, Cham. https://doi.org/10.1007/978-3-030-25510-7_18*
7. *Ding J., Petzoldt A., Schmidt D. S. Multivariate Cryptography // In: Multivariate Public Key Cryptosystems. Advances in Information Security. 2020. V. 80. Springer, New York, NY. https://doi.org/10.1007/978-1-0716-0987-3_2*
8. *Ding J., Petzoldt A., Schmidt D. S. The Matsumoto-Imai Cryptosystem // In: Multivariate Public Key Cryptosystems. Advances in Information Security. 2020. V. 80. P. 25–60. Springer, New York, NY. https://doi.org/10.1007/978-1-0716-0987-3_3*
9. *Moldovyan D. N. A practical digital signature scheme based on the hidden logarithm problem // Computer Science Journal of Moldova. 2021. Vol. 29. N.2(86). P. 206–226.*
10. *Moldovyan D. N. New Form of the Hidden Logarithm Problem and Its Algebraic Support // Bulletin of Academy of Sciences of Moldova. Mathematics. 2020. No. 2 (93). P. 3–10.*

11. *Ikematsu Y., Nakamura S., Takagi T. Recent progress in the security evaluation of multivariate public-key cryptography // IET Information Security. 2022. P. 1–17. DOI: 10.1049/ise2.12092*
12. *Ding J., Petzoldt A., Schmidt D. S. Solving Polynomial Systems // In: Multivariate Public Key Cryptosystems. Advances in Information Security. Springer. New York. 2020. V. 80. P. 185–248. https://doi.org/10.1007/978-1-0716-0987-3_8*
13. *Cartor R., Cartor M., Lewis M., Smith-Tone D. IPRainbow // In: Cheon, J. H., Johansson, T. (eds) Post-Quantum Cryptography // Lecture Notes in Computer Science. 2022. V. 13512. P. 170–184. Springer, Cham. https://doi.org/10.1007/978-3-031-17234-2_9*
14. *Ding, J., Petzoldt, A., Schmidt, D. S. Oil and Vinegar // In: Multivariate Public Key Cryptosystems. Advances in Information Security. 2020. V. 80. P. 89–151. Springer, New York, NY. https://doi.org/10.1007/978-1-0716-0987-3_5*
15. *Moldovyan N. A. Finite algebras in the design of multivariate cryptography algorithms // Bulletin of Academy of Sciences of Moldova. Mathematics. 2023. No. 3 (103). P. 80–89. DOI: https://doi.org/10.56415/basm.y2023.i3.p80*
16. *Moldovyan A. A., Moldovyan N. A. Vector finite fields of characteristic two as algebraic support of multivariate cryptography // Computer Science Journal of Moldova. 2024. V.32. N.1(94). P. 46–60. DOI: 10.56415/csjm.v32.04*
17. *Moldovyan A. A., Moldovyan D. N. A New Method for Developing Signature Algorithms // Bulletin of Academy of Sciences of Moldova. Mathematics, 2022. No. 1(98), pp. 56–65. DOI: https://doi.org/10.56415/basm.y2022.i1.p56.*
18. *Молдовян Д. Н., Молдовян А. А. Алгебраические алгоритмы ЭЦП, основанные на трудности решения систем уравнений // Вопросы кибербезопасности. 2022. № 2(48). С. 7–17. DOI: 10.21681/2311-3456-2022-2-7-17.*
19. *Молдовян А. А., Молдовян Н. А. Алгоритмы ЭЦП на конечных некоммутативных алгебрах над полями характеристики два // Вопросы кибербезопасности. 2022. № 3(49). С. 58–68. DOI: 10.21681/2311-3456-2022-3-58-68.*
20. *Moldovyan D. N. A new type of digital signature algorithms with a hidden group // Computer Science Journal of Moldova. 2023, vol. .31, No.1(91), pp. 111–124. doi:10.56415/csjm.v31.06.*
21. *Молдовян А. А., Молдовян Д. Н., Костина А. А. Алгебраические алгоритмы ЭЦП с полной рандомизацией подписи // Вопросы кибербезопасности. 2024. № 2(60). С. 95–102. DOI: 10.21681/2311-3456-2024-2-95-102.*
22. *Молдовян Д. Н., Костина А. А. Способ усиления рандомизации подписи в алгоритмах ЭЦП на некоммутативных алгебрах // Вопросы кибербезопасности. 2024. № 4(62). С. 71–81. DOI: 10.21681/2311-3456-2024-4-71-81.*
23. *Moldovyan N. A., Moldovyan A. A. Structure of a 4-dimensional algebra and generating parameters of the hidden logarithm problem // Вестник Санкт-Петербургского университета. Прикладная математика. Информатика. Процессы управления. 2022. Т. 18. Вып. 2. С. 209–217. https//doi.org/10.21638/11701/spbu10.2022.202*
24. *Moldovyan D. N., Moldovyan A. A., Moldovyan N. A. Structure of a finite non-commutative algebra set by a sparse multiplication table // Quasigroups and Related Systems. 2022, vol. 30, no. 1, pp. 133–140. https://doi.org/10.56415/qrs.v30.11*
25. *J. Ding, A. Petzoldt Current State of Multivariate Cryptography // IEEE Security and Privacy Magazine. 2017, vol. 15, no. 4, pp. 28–36.*
26. *Moldovyan A. A., Moldovyan N. A. Post-quantum signature algorithms with a hidden group and doubled verification equation // Информационно-управляющие системы. 2023. № 3. С. 59–69. doi:10.31799/1684-8853-2023-3-59-69.*

ALGEBRAIC SIGNATURE ALGORITHMS WITH TWO HIDDEN GROUPS

Moldovyan N. A.[4], Petrenko A. S.[5]

Purpose of work *is increasing the performance of algebraic digital signature algorithms with enhanced signature randomization.*

Research methods: *application of two hidden commutative groups to enhance signature randomization in algebraic digital signature algorithms on finite non-commutative associative algebras (FNAA). Known results on the study of the decomposition of four-dimensional FNAA as finite rings into a set of commutative subrings are used to calculate the parameters of the digital signature algorithm with two hidden commutative groups. Application of a verification equation with two entries of the tuning signature element, which is a vector* S*, calculated by two commutative elements from different hidden groups. The presence of the exponentiation operation to a power, calculated as the value of the hash function of* S*. FNAAs specified by sparse multiplication tables of basis vectors are used as an algebraic support of the digital signature algorithm.*

Results of the study: *for the first time, the randomization enhancement mechanism is implemented in the algebraic digital signature algorithm without using the doubling of the verification equation. The developed digital signature algorithm is distinguished by the use of two hidden groups for calculating a random latch vector, by which the randomizing element of the generated signature is calculated. The latter ensures increased randomization not only for the signature values, but also for the value of the fixator vector. Due to this, the potentially achievable level of security is significantly increased. The sufficiency of performing the signature verification using only one verification equation is ensured by the following two techniques: 1) multiple entries of the tuning signature element* S *in the products that exponentiated to a large power, which appear in the right-hand side of the verification equation and 2) using the value of the hash function, depending on the vector* S*, as the value of the degree of one of the exponentiation operations performed during the signature authenticity verification*

4 Nikolay A. Moldovyan, Ph.D. (in Tech.) chief researcher of laboratory of computer security problems, St. Petersburg Federal Research Center of the Russian Academy of Sciences, St. Petersburg, Russia. ORCID: https://orcid.org/0000-0002-4483-5048. Scopus Author ID: 6603837461. E-mail: mdn.spectr@mail.ru

5 Alexey S. Petrenko, junior research fellow of laboratory of computer security problems, St. Petersburg Federal Research Center of the Russian Academy of Sciences, St. Petersburg, Russia. ORCID: https://orcid.org/0000-0002-9954-4643. Scopus Author ID: 57200260915. E-mail: a.petrenko1999@rambler.ru

procedure. An analysis of security to a direct attack and to signature forgery on the on base of many known signatures is performed.

Practical relevance: *The scientific and practical significance of the results of the article consists in increasing the performance of algebraic digital signature algorithms with two hidden commutative groups, which, due to the small sizes of the signature and public key, are of interest for the development of practical post-quantum signature standards.*

Keywords: *finite non-commutative algebra; associative algebra; computationally difficult problem; hidden group; digital signature; signature randomization; post-quantum cryptography.*

References

1. *Post-Quantum Cryptography. 15th International Conference, PQCrypto 2024, Oxford, UK, June 12–14, 2024, Proceedings. Lecture Notes in Computer Science. 2024. V. 14771–14772. Springer, Cham.*
2. *Post-Quantum Cryptography. 14th International Conference, PQCrypto 2023, College Park, MD, USA, August 16–18, 2023, Proceedings. Lecture Notes in Computer Science. 2023. V. 14154. Springer, Cham.*
3. *Battarbee C., Kahrobaei D., Perret L., Shahandashti S.F. SPDH-Sign: Towards Efficient, Post-quantum Group-Based Signatures. In: Johansson, T., Smith-Tone, D. (eds) Post-Quantum Cryptography. PQCrypto 2023. Lecture Notes in Computer Science, 2023. V. 14154. P. 113–138. Springer, Cham. https://doi.org/10.1007/978-3-031-40003-2_5*
4. *Gärtner J. NTWE: A Natural Combination of NTRU and LWE. In: Johansson, T., Smith-Tone, D. (eds) Post-Quantum Cryptography. PQCrypto 2023. Lecture Notes in Computer Science, 2023, vol 14154, pp. 321–353. Springer, Cham. https://doi.org/10.1007/978-3-031-40003-2_12*
5. *Alamelou Q., Blazy O., Cauchie S., Gaborit Ph. A code-based group signature scheme. Designs, Codes and Cryptography. 2017. V. 82. N. 1–2. P. 469–493.*
7. *Ding J., Petzoldt A., Schmidt D. S. Multivariate Cryptography. In: Multivariate Public Key Cryptosystems. Advances in Information Security. 2020. V. 80. Springer, New York, NY. https://doi.org/10.1007/978-1-0716-0987-3_2*
8. *Ding J., Petzoldt A., Schmidt D. S. The Matsumoto-Imai Cryptosystem. In: Multivariate Public Key Cryptosystems. Advances in Information Security. 2020. V. 80. P. 25–60. Springer, New York, NY. https://doi.org/10.1007/978-1-0716-0987-3_3*
9. *Moldovyan D. N. A practical digital signature scheme based on the hidden logarithm problem. Computer Science Journal of Moldova. 2021. Vol. 29. N.2(86). P. 206–226.*
10. *Moldovyan D. N. New Form of the Hidden Logarithm Problem and Its Algebraic Support. Bulletin of Academy of Sciences of Moldova. Mathematics. 2020. No. 2 (93). P. 3–10.*
11. *Ikematsu Y., Nakamura S., Takagi T. Recent progress in the security evaluation of multivariate public-key cryptography. IET Information Security. 2022. P. 1–17. DOI: 10.1049/ise2.12092*
12. *Ding J., Petzoldt A., Schmidt D. S. Solving Polynomial Systems. In: Multivariate Public Key Cryptosystems. Advances in Information Security. Springer. New York. 2020. V. 80. P. 185–248. https://doi.org/10.1007/978-1-0716-0987-3_8*
13. *Cartor R., Cartor M., Lewis M., Smith-Tone D. IPRainbow. In: Cheon, J.H., Johansson, T. (eds) Post-Quantum Cryptography. Lecture Notes in Computer Science. 2022. V. 13512. P. 170–184. Springer, Cham. https://doi.org/10.1007/978-3-031-17234-2_9*
14. *Ding, J., Petzoldt, A., Schmidt, D. S. Oil and Vinegar. In: Multivariate Public Key Cryptosystems. Advances in Information Security. 2020. V. 80. P. 89–151. Springer, New York, NY. https://doi.org/10.1007/978-1-0716-0987-3_5*
15. *Moldovyan N. A. Finite algebras in the design of multivariate cryptography algorithms. Bulletin of Academy of Sciences of Moldova. Mathematics. 2023. No. 3 (103). P. 80–89. DOI: https://doi.org/10.56415/basm.y2023.i3.p80*
16. *Moldovyan A. A., Moldovyan N. A. Vector finite fields of characteristic two as algebraic support of multivariate cryptography. Computer Science Journal of Moldova. 2024. V.32. N.1(94). P. 46–60. DOI: 10.56415/csjm.v32.04*
17. *Moldovyan A. A., Moldovyan D. N. A New Method for Developing Signature Algorithms. Bulletin of Academy of Sciences of Moldova. Mathematics, 2022. No. 1(98), pp. 56–65. DOI: https://doi.org/10.56415/basm.y2022.i1.p56.*
18. *Moldovyan D. N. Moldovyan A. A. Algebraic signature algorithms based on difficulty of solving systems of equations. Voprosy kiberbezopasnosti [Cibersecurity questtions]. 2022. N. 2(48). P. 7–17. DOI: 10.21681/2311-3456-2022-2-7-17.*
19. *Moldovyan A. A., Moldovyan N. A. Signature algorithms on finite non-commutative algebras over fields of characteristic two. Voprosy kiberbezopasnosti [Cibersecurity questtions]. 2022. № 3(49). C. 58–68. DOI: 10.21681/2311-3456-2022-3-58-68.*
20. *Moldovyan D. N. A new type of digital signature algorithms with a hidden group. Computer Science Journal of Moldova. 2023, vol. .31, No.1(91), pp. 111–124. doi:10.56415/csjm.v31.06.*
21. *Moldovyan A. A., Moldovyan D. N., Kostina A. A. Algebraic signature algorithms with complete signature randomization. Voprosy kiberbezopasnosti [Cibersecurity questtions]. 2024. № 2(60). C. 95–102. DOI: 10.21681/2311-3456-2024-2-95-102.*
22. *Moldovyan D. N., Kostina A. A. A method for strengthening signature randomization in algebraic signature algorithms on non-commutative algebras. Voprosy kiberbezopasnosti [Cibersecurity questtions]. 2024. № 4(62). C. 71-81. DOI: 10.21681/2311-3456-2024-4-71-81.*
23. *Moldovyan N. A., Moldovyan A. A. Structure of a 4-dimensional algebra and generating parameters of the hidden logarithm problem. Vestnik of Saint Petersburg University. Applied Mathematics. Computer Science. Control Processes. 2022. T. 18. Вып. 2. C. 209–217. https//doi.org/10.21638/11701/spbu10.2022.202*
24. *Moldovyan D. N., Moldovyan A. A., Moldovyan N. A. Structure of a finite non-commutative algebra set by a sparse multiplication table. Quasigroups and Related Systems. 2022, vol. 30, no. 1, pp. 133–140. https://doi.org/10.56415/qrs.v30.11*
25. *J. Ding, A. Petzoldt. Current State of Multivariate Cryptography. IEEE Security and Privacy Magazine. 2017, vol. 15, no. 4, pp. 28–36.*
26. *Moldovyan A. A., Moldovyan N. A. Post-quantum signature algorithms with a hidden group and doubled verification equation. Informatsionno-upravliaiushchie sistemy [Information and Control Systems], 2023, no. 3, pp. 59–69. doi:10.31799/1684-8853-2023-3-59-69.*

ПОВЫШЕНИЕ «УСТОЙЧИВОСТИ» РЕГЛАМЕНТОВ ДЕЯТЕЛЬНОСТИ КАК СПОСОБ ПРОТИВОДЕЙСТВИЯ НЕУМЫШЛЕННОМУ ИНСАЙДИНГУ

Буйневич М. В.[1], Моисеенко Г. Ю.[2]

DOI: 10.21681/2311-3456-2024-6-108-116

Цель исследования: *обеспечение безопасности информационных ресурсов организации от угрозы неумышленного инсайдинга за счет повышения «устойчивости» регламентов деятельности сотрудников.*

Методы исследования: *системный анализ, аналитическое моделирование, синтез, гипотетический эксперимент, программная инженерия.*

Полученные результаты: *получена графоаналитическая модель предметной области – неумышленного инсайдинга, разработан пошаговый метод синтеза устойчивых регламентов деятельности и архитектура программного комплекса моделирования инструкций; предполагается, что эти научные результаты на данный момент не имеют релевантных аналогов. Теоретическая значимость работы состоит в переводе деятельности, традиционно описываемой на естественном языке, в аналитическую плоскость. Практическая же значимость определяется применением каждого из результатов для повышения безопасности защищаемых информационных ресурсов в практически любой организации, связанной с информационными технологиями.*

Научная новизна *состоит в том, что впервые в качестве уязвимости организации рассматривается «неустойчивость» регламентов деятельности сотрудников (инструкций), а в качестве источника угрозы безопасности информационных ресурсов – девиация поведения сотрудников, вследствие чего происходит отклонение от шагов инструкции.*

Ключевые слова: *информационные ресурсы, регламент деятельности, неумышленный инсайдинг, угроза безопасности, способ противодействия, моделирование.*

Введение

Обеспечение сохранности информационных ресурсов (далее – ИР) является важнейшим аспектом функционирования любой организации; особенно это критично для организаций, обеспечивающих устойчивое функционирование и безопасность государства. ИР таких организаций могут подвергаться целому пулу деструктивных воздействий различной природы – программной, технической, иной. Однако особое место среди них занимают те, которые исходят изнутри самого защищаемого периметра, поскольку источник угрозы в этом случае уже формально преодолел ряд защитных мер. Характерным направлением такого рода угроз является инсайдинг, суть которого заключается в неправомерной деятельности сотрудников организации против своих же защищаемых ИР; как правило, с целью незаконного овладения информацией, которая является собственностью организации.

При этом инсайдеров можно поделить на два типа: заведомо заинтересованного в неправомерных действиях (например, если он был внедрен или завербован враждебным государством или конкурирующей организацией) [1], и легального сотрудника – несознательно (неосознанно) нарушившего инструкции, что привело к возникновению инцидента с ИР (например, если он по халатности раскрыл конфиденциальную информацию) [2]. Второй тип инсайдерства – неумышленного – в некоторой степени даже опаснее первого, поскольку такой сотрудник соответствует практически всем критериям отбора (не замечен в подозрительных связях, лоялен, ответственен, не склонен к… и проч.) и не может быть выявлен при приеме на работу или при регулярных контрольных проверках.

Ситуация еще более усложняется, если легальный сотрудник в принципе выполняет все действия строго по инструкциям и допускает на первый взгляд незначительную «оплошность», которая, однако, приводит к катастрофическим последствиям; такая ситуация описывается в теории катастроф, как прохождение точки бифуркации. Однако, нисколько не оправдывая сотрудника, для недопущения подобных ситуаций необходимо учитывать человеческий фактор, которому подвержены абсолютно все субъекты, обладающие разумом, эмоциями, психикой и другими аспектами, отсутствующими у автомата. И если устранение подобных субъективных уязвимостей в самом человеке находится в зоне ответственности психологических

1 Буйневич Михаил Викторович, доктор технических наук, профессор, профессор кафедры прикладной математики и информационных технологий Санкт-Петербургского университета государственной противопожарной службы МЧС России, Санкт-Петербург. ORCHID: https://orcid.org/0000-0001-8146-0022. Scopus Author ID: 56122749800. E-mail: bmv1958@yandex.ru

2 Моисеенко Григорий Юрьевич, руководитель направления Министерства обороны РФ, Москва, Россия. E-mail: mogreq@mail.ru

и иных социальных наук, то альтернативное решение стоит искать в более организационно-технической составляющей предметной области.

Причины неумышленного инсайдерства

Базовый анализ предметной области [3–6] показывает, что причина возникновения неумышленного инсайдерства со стороны благонадежных сотрудников может лежать в плоскости формирования «небезопасных» должностных и/или иных регламентирующих деятельность инструкций, по сути, представляющих собой совокупность (в более строгом смысле – граф) переходов субъекта и окружающей системы (на которую он влияет) между некоторыми состояниями. При этом, ошибочный переход в иное состояние, не указанное в инструкции, в результате может привести к иному развитию всего «пути» субъекта и, как результат, к попаданию в состояние, характеризуемое нарушениями конфиденциальности, целостности и доступности ИР. Одной из основных причин такой ситуации является близость состояний, позволяющая субъекту (в т.ч. под влиянием внешних воздействий, изменяющих его внутренние факторы) выполнить ошибочное действие, перейдя тем самым в незапланированное состояние.

Поясним данную идею на следующем примере, который будет являться «сквозным» для статьи. Предположим, что высококвалифицированный, опытный и благонадёжный сотрудник в процессе выполнения должностных обязанностей получает документ с конфиденциальными сведениями в пункте «A» и переносит его в пункт «B». При этом он проходит около помещения «C». В случае строгого и автоматического выполнения инструкций, сотрудник походит мимо помещения «C», перенеся документ из начальной в конечную точку.

Однако, исходя из того, что сотрудник является человеком, то он обладает собственной целенаправленной активностью, связанной с мыслительной деятельностью, потребностями, ощущениями и пр. Как результат, он может «зайти» в помещение «C» (например, если это столовая или место для курения – в зависимости от повышенных потребностей сотрудника), по небрежности оставить там документ, выйти из помещения и продолжить выполнение инструкции вплоть до пункта «B». Естественно, пропажа документа в момент посещения помещения «C» может быть достаточно оперативно обнаружена (например, если его сдача контролируется другими сотрудниками или техническими средствами), однако документ будет находиться в «небезопасном» состоянии время, которое сотрудник затратит на движение из «C в «B». При этом необходимо отметить, что уже, как только документ покинул пункт «A» и до тех пор, пока он оказался в пункте «B», его безопасность (даже в условиях нахождения в руках человека) была снижена – например, вследствие потенциально возможного физического или иного воздействия на сотрудника.

Примечание. Следует говорить о снижении/повышении именно *уровня* безопасности, так как применительно к безопасности более корректно «обеспечена – не обеспечена» или нарушена. Именно так следует трактовать понятие «безопасность» применительно к ИР и, в частности, конфиденциальным документам.

Таким образом, часть ответственности за нарушение безопасности (в данном случае – возможности раскрытия конфиденциальной информации) лежит на непродуманной инструкции сотрудника [7] в рамках конкретной организации (поскольку, если между пунктами «A» и «B» проходит полностью изолированный коридор, то шанс попадания документа третьим лицам будет минимальным).

Анализ нарушения для приведенного примера позволяет выдвинуть две следующие его «глубинные» причины. Во-первых, очевидную – физическое расположение «C», близкое к месту, через которое проходит маршрут сотрудника из «A» в «B», создает потенциальную возможность отклонения им от регламентированной траектории. А, во-вторых, скрытую – возможности, предоставляемые помещением «C», оказываются близкими к потенциальным потребностям, присущим сотруднику; такой сотрудник может объяснить свои действия фразами «захотелось перекусить» или «покурить».

Исходя из приведенного (и достаточно показательного) примера можно выделить проблему предметной области в форме противопоставления потребностей и возможностей следующим образом. С одной стороны, от сотрудников организации требуется выполнения инструкций, безопасное по отношению к ИР. С другой стороны, наличие человеческого фактора приводит к девиации поведения сотрудников (т.е. отхождение от пользовательских паттернов [8]), вследствие чего происходит отклонение от шагов инструкции, что может приводить к угрозе безопасности ИР [9]. Исходя из того, что эффективность противодействия самой сути человека является в некотором смысле спорной, то возможным разрешением противоречия может являться снижение вероятности угрозы путем повышения «устойчивости» инструкций к их нарушению сотрудником под воздействием внутренних (т.е. человеческих) факторов (вызванных, в том числе и внешними).

Говоря простым языком, требуется создание таких инструкций, которые бы заданная группа сотрудников в данной организации не смогла бы нарушить, а в идеале – даже нарушив, но незначительно, не смогла бы создать угрозу безопасности ИР. Для примера выше, инструкция может быть модифицирована отдалением пути перемещения документа

с конфиденциальными сведениями от потенциально-опасных помещений.

Альтернативные способы в примере, такие, как перенесение помещения «С» или разнесение времени работы с документами и режима открытия помещения «С», хотя и «имеют место быть», но в статье не рассматриваются, т.к. являются в разы более ресурсозатратными и имеющими в целом негативное влияние на общее функционирование организации. Повышения же устойчивости инструкций к неумышленному инсайдингу можно достичь, предоставив для начала аппарат оценки их безопасности. Как результат, руководитель (ответственное за безопасность должностное лицо) сможет варьировать содержание инструкций, их параметры и порядок действий сотрудников для достижения сценариев поведения, наиболее безопасных с позиции ИР (даже в условиях «нулевого доверия» [10]).

В интересах разрешения противоречия указанным способом предложим достаточно каноническую методологическую трехэтапную схему исследования с получением соответствующих научных результатов (в кавычках):

1. Анализ предметной области с получением «Аналитической модели неумышленного инсайдинга» (далее – Модель), взаимоувязывающей в формализованном виде элементы организации, инструкции, параметры (т.е. внутренние факторы) сотрудников и их характеристики (т.е. влияние на выполняемые регламентированные действия), ИР и их безопасность;
2. Создание «Метода структурно-параметрического синтеза и оценки устойчивости инструкций» (далее – Метод), позволяющей построить Модель, задать ее параметры, а также произвести оценки инструкций с позиции безопасности ИР;
3. Разработка «Архитектуры программного комплекса моделирования устойчивости инструкций», представляющей собой многослойное описание (с позиции логических модулей, информационных объектов, алгоритмов и т.п.) реализации Метода, функционирующего на базе Модели; за этим следует разработка соответствующего прототипа (далее – Прототип), базовое тестирование которого покажет работоспособность Метода и адекватность Модели.

По завершении указанных этапов исследования потребуется произвести оценку полученных результатов, например, путем моделирования реально произошедших инцидентов по причине нарушения инструкций сотрудниками организации и установления возможности и точности выявления (идентификации и локализации) потенциально возможных нарушений.

Опишем далее гипотетическое (на данный момент исследования) представление ожидаемых результатов на каждом этапе, указав способы их реализации.

Этап 1. Аналитическая модель

На первом этапе требуется провести анализ предметной области на предмет определения основных ее онтологических сущностей и их связей. Так, очевидно, что основными сущностями будут следующие: структура конкретной организации, защищаемые ИР, инструкции, сотрудники, безопасность. К вторичным сущностям можно отнести уточняющие основные, а именно следующие: внутренние факторы сотрудников и выполняемые в рамках инструкций действия, состояния сотрудников (с позиции выполнения инструкций), состояния защищаемых ИР, переходы между состояниями и влияние на безопасность.

Анализ онтологической модели позволит формализовать все ее сущности и их связи в виде соответствующей аналитической модели, что будет первым научным результатом. Идея такой модели может строиться на следующих предпосылках.

Во-первых, инструкции, хотя и описаны, как правило, в достаточно общем виде, однако их выполнение происходит на конкретной структуре организации (например, если в организации принципиально отсутствует работа с конфиденциальными документами в бумажном виде, то их перенос из пункта «А» в «В» невозможен в принципе).

Во-вторых, выполнение сотрудником действий, указанных в инструкции, приводит к его переходу в другое состояние (например, выполнив действие согласно инструкции по переходу из «А» в «В», сотрудник поменяет пространственную характеристику с координаты «А» на координату «В»).

В-третьих, последовательность перемещения сотрудника между состояниями влияет также и на ИР, и в особенности те, к которым применяется инструкция (например, выход сотрудникам из пункта «А» с документом приводит к некоторому снижению безопасности последнего, а посещение сотрудником помещения «С», в котором, по внутреннему регламенту, все документы должны быть временно оставлены без присмотра и вовсе приводит к реальной предпосылке утечки конфиденциальной информации).

Следуя указанным предпосылкам, можно переложить действия из примера на Модель в графическом представлении. Предположим, что должностная инструкция звучит, как «Сотрудник должен получить документ в пункте «А», перенести его в пункт «В» и сдать там на хранение». Также учтем специфику структуры организации, заключающуюся в расположении помещения «С» на пути следования из «А в «В».

Таким образом, можно ввести следующие состояния сотрудника:

«W» (*аббр. от англ.* Workplace, *перев. на русс.* Рабочее Место) – штатное расположение сотрудника в организации (например, рабочий кабинет);
«A» – нахождение сотрудника в пункте «A» до получения документа;
«A'» – нахождение сотрудника в пункте «A» после получения документа;
«T'» (*аббр. от англ.* Temporary, *перев. на русс.* Временное Место) – нахождение сотрудника, имеющего с собой документы, около помещения «C»;
«B'» – нахождение сотрудника в пункте «B» до сдачи документа;
«B» – нахождение сотрудника в пункте «B» после сдачи документа.

Аналитически, состояния могут быть записаны следующим образом:

$$P \in \{W, A, A', T', B', B\},$$

где P (*аббр. от англ.* Position, *перев. на русс*. Место, локация) – некоторое (пространственное) состояние сотрудника.

Наличие показателей у каждого из состояний, позволяющих описывать их в едином пространстве, может быть записано следующим образом:

$$\begin{cases} i \in I \\ N = |I| \\ P \equiv \langle P^1 \ldots P^i \ldots P^N \rangle \end{cases}$$

где i – показатель, I – множество всех показателей, N – количество всех показателей (как мощность множества I), P^i – значение i-го показателя состояния.

«Безопасное» моделирование выполнения инструкции представлено на Рисунке 1 – сценарий 1; обозначение «[ ]» означает состояние документа: «+» – документ у сотрудника, «–» – документ не у сотрудника.

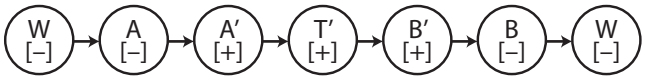


Рис. 1. Пример моделирования «безопасного» выполнения инструкции

Так, следуя Рисунку 1, в некоторый момент времени (состояние «A' [+]») документ оказывается у сотрудника, а в последующий (состояние «B' [–]») он переходит в хранилище.

Аналитически, такие переходы между состояниями могут быть записаны следующим образом:

Сценарий 1: $W \rightarrow A \rightarrow A' \rightarrow T' \rightarrow B' \rightarrow W$.

Также необходимо учесть, что у сотрудника есть право на обед или «перекур», моделирование чего представлено на Рисунке 2 – сценарий 2.

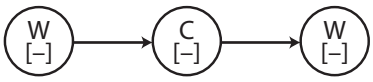


Рис. 2. Пример моделирования не директивных, но правомерных действий сотрудника (не регламентированных инструкцией по работе с документами)

Аналитически такие переходы между состояниями могут быть записаны следующим образом:

Сценарий 2: $W \rightarrow C \rightarrow W$,

где C – нахождение сотрудника в помещении «C».

Так, согласно Рисунку 2, сотрудник в некоторый момент времени может посетить помещение «C», а затем вернуться обратно на рабочее место; при нормальном стечении обстоятельств, документ в эти моменты у него отсутствует.

Для корректного моделирования обоих сценариев, представленных на Рисунках 1 и 2 (путем их отображения в едином пространстве), учтем близость состояний «T'» к «C» (как метрику в этом пространстве) следующих показателей состояния: физическое расположение (т.к. помещение «C» находиться около траектории от «A» к «B») и удовлетворенность потребностей (т.к. помещении «C» может обеспечить сотрудника едой или реализацией иных естественных потребностей и привычек). Аналитическая запись такой близости может быть записана следующим образом:

$$|P_1 - P_2| < \delta,$$

где P_1 и P_2 – некоторые состояния, «$|\ldots|$» – метрика или расстояние между состояниями в пространстве их показателей, δ – параметр близости состояний. Суть данной записи означает, что состояния называются близкими, если расстояние между ними меньше некоторого значения.

Моделирование обоих сценариев поведения сотрудника с учетом близости состояний приведено на Рисунке 3 (т.е. сценарии 1 и 2); пунктирными линиями показаны переходы между состояниями согласно предыдущим сценариям, а линиями с красными цифрами – возможное развитие событий.

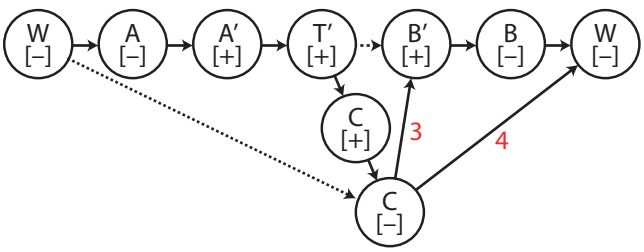


Рис. 3. Пример моделирования «мультисценарных» действий сотрудника

Так, следуя Рисунку 3, до состояния «T'» происходит корректное выполнение инструкции, за которым по причине близости состояний «T'» и «C» происходит «халатный переход» сотрудника во второе состояние, при этом при наличии на руках документа – «$C[+]$»; таким образом возникает первая предпосылка к неумышленному инсайдингу. Затем, по мере нахождения в помещении «С» и уже следуя его «правилам», сотрудник оказывается в состоянии «$C[-]$» – например, оставив документ на столе или полке. После этого можно предположить развитие действий сотрудника по двум вариациями совместного сценария: сценарий 3 – продолжающему выполнение текущего сценария 1, при котором сотрудник продолжит перемещение в пункт «В», перейдя для этого в состояние «B'» (модель которого приведена на Рисунке 1); сценарий 4 – срабатывание привычки сотрудника идти после помещения «С» на рабочее место, т.е. в состояние «W» (модель которого приведена на Рисунке 1). Такие сценарные развития показаны на Рисунке 3 красными цифрами.

Аналитически такие переходы между состояниями могут быть записаны следующим образом:

$$\begin{cases} \text{Сценарий 3: } W \to A \to A' \to T' \to C \to C \to B' \to W \\ \text{Сценарий 4: } W \to A \to A' \to T' \to C \to C \to W) \end{cases}$$

Достаточно показательным будет оценка изменения безопасности документа для всех вариантов развития действий сотрудника – двух сценариев и вариаций их объединения. Для этого условно можно считать, что при хранении документа в пунктах «А» и «В» его безопасность максимальна, при нахождении в руках сотрудника – средняя, а при оставлении документа сотрудником в общественном месте – минимальная. График такого изменения безопасности документа приведен на Рисунке 4.

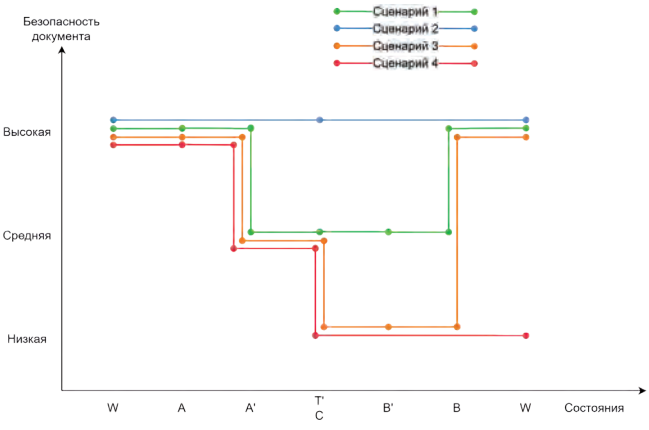


Рис.4. Безопасность документа при выполнении сотрудником действий по различным сценариям

Дадим интерпретацию предложенной динамике изменения безопасности документа, формализовав ее следующем образом:

$$S \in \{High = 3, Middle = 2, Low = 1\},$$

где S – (*аббр. от англ.* Security или Safety, *перев. на русс.* Безопасность) безопасность документа в некоторый момент времени, которая может следующего уровня (с числовыми значениями в скобках): High (*перев. с англ. на русс.* Высокая), Middle (Средняя) и Low (Низкая).

Исходя из того, что между состояниями сотрудника безопасность документа практически не меняется, то средняя оценка может быть произведена как суммирование ее значения на время перехода между состояниями, поделенное на общее время сценария, т.е.:

$$S = \frac{\sum_{i=1 \dots M-1} S_i \times T_i}{\sum_{i=1 \dots M-1} T_i},$$

где i – индекс состояния, M – количество всех состояний, S_i – безопасность документа в состоянии P_i, T_i – время нахождения сотрудника в i-м состоянии.

Для упрощения дальнейших расчетов (но без потери смысла) примем, что время перехода между состояниями одинаковое и равно t; в этом случае, суммарное время всех сценариев (даже с учетом 2-го, в котором не участвует документы) будем считать равным времени 1-го сценария, т.е. 6t.

В сценарии 1 до состояния «A'» и после состояния «B'» документ находится в хранилище и имеет высокую безопасность. В момент перенесения его сотрудником из пункта «А» в «В» безопасность снижается до средней, поскольку документ все также находится под контролем. В этом случае, безопасность такого сценария (с учетом ее числовых значений) равняется:

$$S\,(\text{Сценарий 1}) = \frac{3+3+2+2+2+3}{6} = 2{,}5.$$

В сценарии 2 документ продолжает находиться в хранилище все время и, следовательно, его безопасность не снижается, а безопасность равняется:

$$S\,(\text{Сценарий 2}) = \frac{3+3+3+3+3+3}{6} = 3.$$

В сценарии 3 вначале безопасность изменяется так же, как и в сценарии 1, однако в момент оставления сотрудником помещения «С» без документа, безопасность последнего снижается до низкой (поскольку он становится доступным третьим лицам) и продолжает такой оставаться на всем действии сценария. Безопасность документа в данном случае равняется:

$$S\,(\text{Сценарий 3}) = \frac{3+3+2+1+1+3}{6} \approx 2{,}33.$$

В сценарии 4, в отличие от 3, сотрудник после помещения «С» приходит в пункт «В», где выявляется факт потери документа, за которым, очевидно, следует введение нештатной ситуации с «изъятием» документа из помещения «С» в хранилище, где безопасность документа снова становится высокой. Таким образом, безопасность равняется:

$$S(\text{Сценарий } 4) = \frac{3+3+2+1+1+1}{6} \approx 1{,}83.$$

Как показало моделирование сценариев (и, в частности, неумышленного инсайдинга в сценариях 3 и 4), безопасность защищаемых документов имеет сложный (нелинейный) характер зависимости от динамики изменения состояний.

Этап 2. Метод синтеза

Несмотря на то, что предложенная Модель в принципе содержит всю необходимую информацию о поведении сотрудника в отношении защищаемых ИР согласно инструкции и иных действий в условиях организации, тем не менее, ее адекватное построение считается отдельно стоящей наукоемкой задачей структурно-параметрического синтеза. Также, проведение оценок безопасности ИР в процессе моделирования поведения сотрудников (и в особенности, при неумышленном инсайдинге) нуждается в создании соответствующего математического аппарата (ввиду отсутствия такового). В интересах этого необходимо создание специального Метода, гипотетическими фазами и шагами которого могут стать следующие.

Фаза 1. *Построение Модели*. Данная фаза предназначена для синтеза модели в аналитическом виде.

Шаг 1.1. *Сбор информации об организации*. На этом шаге необходимо собрать информацию о состояниях, которые может «посещать» сотрудник (конкретных или абстрактных в виде их группы), исходя из специфики организации (помещения, ИР, средства защиты, точки доступа и пр.).

Шаг 1.2. *Сбор информации о сотрудниках*. На этом шаге необходимо выделить основные факторы, влияющие на поведение (а точнее, на его девиацию) сотрудников, в том числе, с учетом специфики организации (например, потребность в отдыхе из-за сверхнапряженной эмоциональной работы).

Шаг 1.3. *Сбор информации об инструкциях*. На этом шаге необходимо из инструкций, которые требуется проверить на устойчивость, выделить состояния для сотрудника (например, пункты перемещение, хранилища документов, рабочие места, средства защиты и пр.).

Шаг 1.4. *Сбор информации о нерегламентированных действиях*. На этом шаге необходимо определить действия, которые может выполнять сотрудник, но которые не регламентируются инструкциями. Анализ таких сценариев позволит выделить состояния, возможные при девиации поведения сотрудников, поскольку они отражают не заранее заданные требования к поведению, а особенности человека.

Шаг 1.5. *Сбор информации о защищаемых ИР*. На этом шаге необходимо выделить защищаемые в организации ИР, а также влияние на них действий и состояний сотрудника (например, «грифованные» документы, ключи доступа и пр.)

Шаг 1.6. *Систематизация собранной информации*. На этом шаге необходимо систематизировать (а также и гармонизировать) всю собранную информацию, установив все необходимые связи (например, определить влияние внутренних факторов сотрудников на выполнение действий).

Шаг 1.7. *Формализация модели*. На этом шаге необходимо перевести всю собранную и систематизированную информацию в формализованный вид с использованием структуры и параметров Модели.

Фаза 2. *Моделирование поведения сотрудника*. Данная фаза предназначена для имитации поведения сотрудников в процессе выполнения инструкций с учетом близости состояний и возможности возникновения неумышленного инсайдинга; она состоит из следующих шагов. Также, в результате будет дана оценка устойчивости инструкции в контексте безопасности ИР.

Шаг 2.1. *Выбор должностной инструкции*. На этом шаге необходимо выбрать инструкцию, устойчивость которой требуется проверить, при этом, настроив необходимые для нее параметры с учетом специфики информации, собранной в Фазе 1 (например, как в примере, перемещение конфиденциального документа между двумя пунктами с указанием возможного маршрута и времени доставки).

Шаг 2.2. *Имитация действий сотрудника*. На этом шаге осуществляется непосредственное моделирование действий сотрудников в процессе выполнения инструкции путем имитации его передвижения по состояниями. При этом должны учитываться внутренние факторы сотрудника, пытающиеся «сбить» его с заданного маршрута (например, переход в близкое состояние из-за потребности в отдыхе). Сами факторы могут как задаваться постоянными величинами (например, сотрудники, как правило, с вероятностью 1% могут «заглянуть» в помещение для отдыха), так и учитывать специфику конкретного сотрудника, полученную на основании личностных и иных тестов (например, данный сотрудник, имея низкую потребность в отдыхе, тем не менее, обладает вредными привычками, которые соответствующим образом вносят девиацию в его поведение).

Шаг 2.3. *Оценка устойчивости инструкции*. На этом шаге на основании имитации действий сотрудника следует оценить устойчивость инструкции от изменения своего сценария через близкие состояния (например, как в примере, вероятность оставления документа в помещении С).

Шаг 2.4. *Оценка безопасности ИР при выполнении инструкции*. На этом шаге на основании имитации действий сотрудника следует оценить безопасность защищаемых ИР (усредненное значение или в динамике), исходя из возможных переходов сотрудника на близкие состояния и изменения сценария действий (например, как в примере, оценка средней S).

DOI: 10.21681/2311-3456-2024-6-108-116

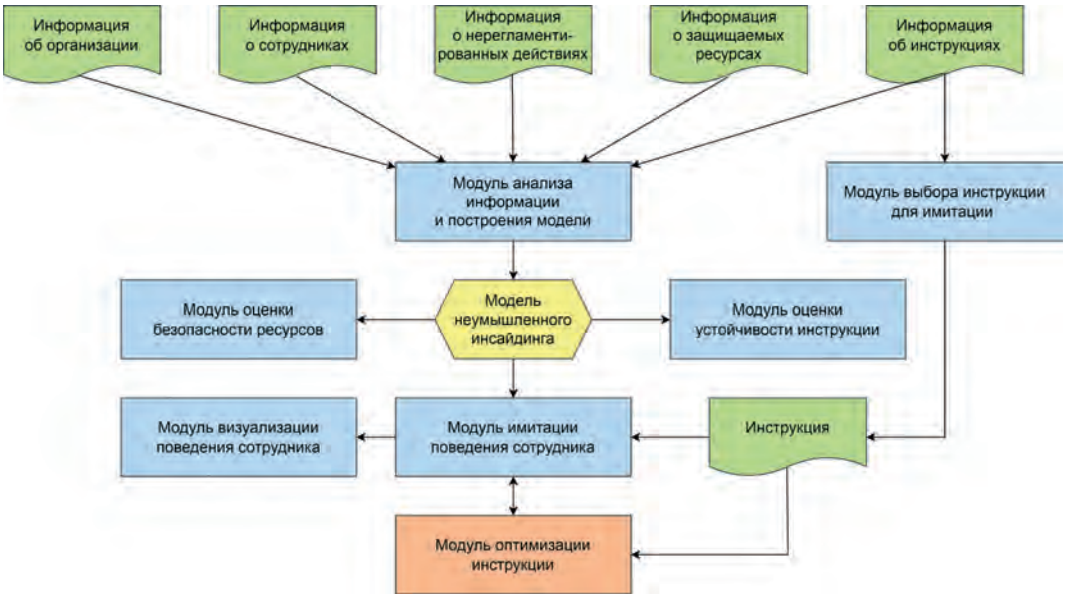


Рис. 5. Базовый структурный слой архитектуры Комплекса

Фаза 3. *Исправление инструкций*. Данная фаза предназначена для непосредственной корректировки инструкций, имеющих недостаточную устойчивость, что приводит к угрозе безопасности ИР.

Шаг 1. *Проверка результатов оценки инструкции*. На этом шаге специалист проверяет полученные оценки инструкции и, в случае удовлетворительного результата, оканчивает метод (например, если инструкция оказалась устойчивой). В противном случае – переход к Шагу 1'.

Шаг 1'. *Выявление «слабых» мест в инструкции*. На этом шаге специалист экспертно выявляет места в инструкции (состояния, переходы между ними, действия сотрудников и т.п.), приводящие к снижению ее устойчивости и потенциальному нарушению безопасности (собственно, слабые места будут являться, как правило, состояниями, из которых может происходить переход в смежные).

Шаг 2. *Корректировка «слабых» мест в инструкции*. На этом шаге специалист также экспертно вносит изменения в инструкцию, предполагая, что это приведет к ее улучшению – повышению устойчивости (например, меняет порядок действий, траекторию движения сотрудника или добавляет переходы через дополнительные состояния).

Шаг 3. *Повторная проверка инструкций*. Шаг является формальным и приводит к повторному переходу на Фазу 2 Метода, но уже с исправленной инструкцией. Таким образом, специалист получает возможность ручной «оптимизации» инструкции.

Следует отметить, что Фаза 3 может быть частично автоматизирована применением определенных техник оптимизации, поскольку в данном случае как устойчивость инструкций, так и безопасность ИР организации можно считать целевой функцией, требующей максимизации[3].

Этап 3. Программный комплекс

Для автоматизации Метода может быть применена область программной инженерии, так как потребуется создать программный комплекс для моделирования инструкций на основании собранной специалистом информации (далее – Комплекс). Такое программное решение позволит, как визуально имитировать деятельность сотрудников, так и автоматически производить необходимые вычисления (устойчивости инструкций и безопасности ИР). Интерактивность работы с Комплексом даст возможность специалисту корректировать инструкции и оценивать получаемые результаты.

Базовый структурный слой архитектуры Комплекса (как совокупности логических модулей и их связей) может иметь вид, представленный на Рисунке 5.

Структурный слой архитектуры (см. Рисунок 5) является интуитивно понятным и полностью отражает работу Метода, функционирующего на основании Модели. Архитектурными элементами Комплекса являются следующие:

1. «Информация об организации» – вводимая информация, собираемая на Шаге 1.1 Метода (далее – на Шаге);
2. «Информация о сотрудниках» – вводимая информация, собираемая на Шаге 1.2;
3. «Информация о нерегламентированных действиях» – вводимая информация, собираемая на Шаге 1.4;

3 Смоленцева Т. Е. Методы определения целевой функции организационных систем // Моделирование, оптимизация и информационные технологии. 2018. Т. 6. № 3 (22). С. 143–152.

4. «Информация о защищаемых ресурсах» – вводимая информация, собираемая на Шаге 1.5;
5. «Информация об инструкциях» – вводимая информация, собираемая на Шаге 1.3;
6. «Модуль анализа информации и построения модели» – модуль для формализации вводимой информации с целью построения Модели;
7. «Модель неумышленного инсайдинга» – формализованное представление Модели, готовой для имитационного моделирования и проведения оценок;
8. «Модуль выбора инструкции для имитации» – модуль для взаимодействия с оператором в интересах выбора инструкции, необходимой для моделирования;
9. «Инструкция» – формализованное представление инструкции, подходящее для проведения моделирования (в т.ч. в процессе оптимизации);
10. «Модуль имитации поведения сотрудника» – основной модуль Комплекса, позволяющий проводить моделирование действий сотрудника с учетом возникновения инцидентов неумышленного инсайдинга (т.е. переходов на близкие состояния);
11. «Модуль визуализации поведения сотрудника» – модуль для отображения оператору процесса деятельности сотрудника согласно заданной инструкции в графическом или текстовом виде;
12. «Модуль оценки безопасности ресурсов» – модуль для вычислений различных метрик, связанных с безопасностью ИР в соответствии с текущей моделируемой инструкцией;
13. «Модуль оценки устойчивости инструкции» – модуль для вычислений различных метрик, связанных с устойчивостью текущей моделируемой инструкции;
14. «Модуль оптимизации инструкции» – гипотетический модуль, упомянутый при описании метода, проводящий оптимизацию инструкции путем ее структурно-параметрических изменений и оценки влияния этого на целевые функции (вычисляемые модулями под номерами 12 и 13).

Данный комплекс, в случае успешной реализации, предоставит достаточно мощный инструмент как для моделирования, так и оценки выполнения сотрудником инструкций с позиции их устойчивости, а также безопасности ИР организации.

Заключение

Работа посвящена противодействию неумышленному инсайдингу, ведущему к нарушениям информационной безопасности в организациях и качественно отличного от других видов угроз тем, что он сложно выявляем на ранних этапах (например, тестирование сотрудников) и имеет вполне легальный источник (поскольку угроза может исходить от добропорядочных сотрудников). Поскольку данная «застаревшая» проблема в принципе является достаточно «свежей» для науки (то есть, слабо освещенной в научных публикациях), то предлагается проведение отдельного, целостного научного исследования по канонической схеме с созданием таких научных результатов, как аналитическая модель предметной области, метод синтеза устойчивых регламентов деятельности и архитектура программного комплекса моделирования инструкций (естественно, с последующей его реализацией и оценкой). Гипотетически это позволит получить новые научные результаты, на данный момент не имеющие релевантных аналогов.

Новизна работы состоит в том, что впервые в качестве уязвимости организации рассматривается «неустойчивость» регламентов деятельности сотрудников (инструкции), а в качестве источника угрозы безопасности ИР – девиация поведения сотрудников, вследствие чего происходит отклонение от шагов инструкции.

Теоретическая значимость работы состоит в переводе деятельности, традиционно описываемой на естественном языке, в аналитическую плоскость. Практическая же значимость определяется применением каждого из результатов для повышения безопасности защищаемых ИР в практически любой организации, связанной с ИТ.

В контексте последней «связки», продолжением исследования должна стать глубокая научная проработка вопросов проецирования соответствующих результатов на сферу ИТ, то есть на виртуальную (цифровую) среду деятельности неумышленного инсайдера по отношению к ИР организации.

Литература

1. *Buinevich M., Izrailov K., Kotenko I., Ushakov I., Vlasov D. Approach to combining different methods for detecting insiders // The proceedings of 4th International Conference on Future Networks and Distributed Systems (New York, USA, 2020). Iss. 26. PP. 1–6. DOI: 10.1145/3440749.3442619.*
2. *Власов Д. С. К вопросу о мотивации инсайдера организации и способах его классификации // Электронный сетевой политематический журнал «Научные труды КубГТУ». 2022. № 1. С. 128–147.*
3. *Власов Д. С. Мультикритериальная модель систематизации способов обнаружения инсайдера // Вопросы кибербезопасности. 2024. № 2 (60). С. 66–73. DOI: 10.21681/2311-3456-2024-2-66-73.*
4. *Буйневич М. В., Власов Д. С., Моисеенко Г. Ю. Комбинирование способов выявления инсайдеров больших информационных систем // Вопросы кибербезопасности. 2024. № 3 (61). С. 2–13. DOI: 10.21681/2311-3456-2024-3-2-13.*
5. *Анализ и систематизация инсайдерских угроз в информационных системах // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2021): сборник научных статей (Санкт-Петербург, 24–25 февраля 2021 года). Т. 4. 2021. С. 399–403*

6. Буйневич М. В., Власов Д. С. Сравнительный обзор способов выявления инсайдеров в информационных системах // Информатизация и связь. 2019. № 2. С. 83–91. DOI: 10.34219/2078-8320-2019-10-2-83-91.
7. Васильев М. В., Федорова А. В. Несоответствие должностных инструкций сотрудников банковской сферы новым угрозам информационной безопасности // Поколение будущего: Взгляд молодых ученых- 2019: сборник научных статей 8-й Международной молодежной научной конференции (Курск, 13–14 ноября 2019 года). 2019. С. 253–255.
8. Нашивочников Н. В. Выявление отклонений в поведенческих паттернах пользователей корпоративных информационных ресурсов с использованием топологических признаков // Вопросы кибербезопасности. 2023. № 4 (56). С. 12–22. DOI: 10.21681/2311-3456-2023-4-12-22.
9. Поляничко М. А. Методика обнаружения аномального взаимодействия пользователей с информационными активами для выявления инсайдерской деятельности // Труды учебных заведений связи. 2020. Т. 6. № 1. С. 94-98. DOI: 10.31854/1813-324X-2020-6-1-94-98.
10. Астахова Л. В. Модель нулевого доверия как фактор влияния на информационное поведение сотрудников организации // Научно-техническая информация. Серия 1: Организация и методика информационной работы. 2022. № 3. С. 13–17. DOI: 10.36535/0548-0019-2022-03-2.

THE INSTRUCTIONS «RESISTANT» INCREASING AS A WAY TO COUNTER UNINTENTIONAL INSIDING

Buinevich M. V.[4], Moiseenko G. Yu.[5]

The goal of the investigation: *ensuring the organization's information resources security from threat of unintentional including by increasing instructions «resistant».*

Research methods: *systems analysis, analytical modeling, synthesis, hypothetical experiment, software engineering.*

Results: *a graphoanalytical model of the unintentional insiding are obtained, a step-by-step method for synthesizing resistant instructions and the architecture of a software package for they modeling are developed. It is assumed that these scientific results currently have no relevant analogues. The theoretical significance of the work consists in translating the activities traditionally described in natural language into an analytical plane. The practical significance is determined by the application of each of the results to improve the protected information resources security in almost any organization related to information technology.*

The scientific novelty *lies in the fact that for the first time, the «instability» of employee regulations is considered as an organization's vulnerability; the employee behaviors deviation is considered as to the security of information resources threat source, as a result of which there is a deviation from the instructions steps.*

Keywords: *information resources, instructions, unintentional insiding, security threat, counteraction method, modeling.*

References

1. Buinevich M., Izrailov K., Kotenko I., Ushakov I., Vlasov D. Approach to combining different methods for detecting insiders // The proceedings of 4th International Conference on Future Networks and Distributed Sys-tems (New York, USA, 2020). Iss. 26. PP. 1–6. DOI: 10.1145/3440749.3442619.
2. Vlasov D. S. K voprosu o motivatsii insaydera organizatsii i sposobakh yego klassifikatsii // Elek-tronnyy setevoy politematicheskiy zhurnal «Nauchnyye trudy KubGTU». 2022. № 1. S. 128–147.
3. Vlasov D. S. Mul'tikriterial'naya model' sistematizatsii sposobov obnaruzheniya insaydera // Vo-prosy kiberbezopasnosti. 2024. № 2 (60). S. 66–73. DOI: 10.21681/2311-3456-2024-2-66-73.
4. Buynevich M. V., Vlasov D. S., Moiseyenko G. YU. Kombinirovaniye sposobov vyyavleniya insayderov bol'shikh informatsionnykh sistem // Voprosy kiberbezopasnosti. 2024. № 3 (61). S. 2–13. DOI: 10.21681/2311-3456-2024-3-2-13.
5. Analiz i sistematizatsiya insayderskikh ugroz v informatsionnykh sistemakh // Aktual'nyye problemy infotelekommunikatsiy v nauke i obrazovanii (APINO 2021): sbornik nauchnykh statey (Sankt-Peterburg, 24–25 fevralya 2021 goda). T. 4. 2021. S. 399–403.
6. Buynevich M. V., Vlasov D. S. Sravnitel'nyy obzor sposobov vyyavleniya insayderov v informatsi-onnykh sistemakh // Informatizatsiya i svyaz'. 2019. № 2. S. 83–91. DOI: 10.34219/2078-8320-2019-10-2-83-91.
7. Vasil'yev M. V., Fedorova A. V. Nesootvetstviye dolzhnostnykh instruktsiy sotrudnikov bankovskoy sfery novym ugrozam informatsionnoy bezopasnosti // Pokoleniye budushchego: Vzglyad molodykh uchenykh- 2019: sbornik nauchnykh statey 8-y Mezhdunarodnoy molodezhnoy nauchnoy konferentsii (Kursk, 13–14 noyabrya 2019 goda). 2019. S. 253–255.
8. Nashivochnikov N. V. Vyyavleniye otkloneniy v povedencheskikh patternakh pol'zovateley korporativ-nykh informatsionnykh resursov s ispol'zovaniyem topologicheskikh priznakov // Voprosy kiberbezopasnosti. 2023. № 4 (56). S. 12-22. DOI: 10.21681/2311-3456-2023-4-12-22.
9. Polyanichko M. A. Metodika obnaruzheniya anomal'nogo vzaimodeystviya pol'zovateley s informatsi-onnymi aktivami dlya vyyavleniya insayderskoy deyatel'nosti // Trudy uchebnykh zavedeniy svyazi. 2020. T. 6. № 1. S. 94-98. DOI: 10.31854/1813-324X-2020-6-1-94-98.
10. Astakhova L. V. Model' nulevogo doveriya kak faktor vliyaniya na informatsionnoye povedeniye sotrud-nikov organizatsii // Nauchno-tekhnicheskaya informatsiya. Seriya 1: Organizatsiya i metodika informatsionnoy raboty. 2022. № 3. S. 13-17. DOI: 10.36535/0548-0019-2022-03-2

4 Mikhail V. Buinevich, Dr.Sc., Professor, Professor of Dep. Applied Mathematics and Information Technologies of Saint-Petersburg University of State Fire Service of EMERCOM of Russia, Saint-Petersburg, Russia. ORCHID: https://orcid.org/0000-0001-8146-0022. Scopus Author ID: 56122749800. E-mail: bmv1958@yandex.ru
5 Grigory Yu. Moiseenko, Head of direction, Ministry of Defense of the Russian Federation, Moscow, Russia. E-mail: mogreq@mail.ru

РАСПРЕДЕЛЕННАЯ СИСТЕМА ОБНАРУЖЕНИЯ СЕТЕВЫХ АТАК НА ОСНОВЕ ФЕДЕРАТИВНОГО ТРАНСФЕРНОГО ОБУЧЕНИЯ

Васильев В. И.[1], Вульфин А. М.[2], Картак В. М.[3], Башмаков Н. М.[4], Кириллова А. Д.[5]

DOI: 10.21681/2311-3456-2024-6-117-129

Цель исследования: *повышение эффективности обнаружения сетевых атак ботнетов за счет применения федеративного трансферного обучения, что позволит аккумулировать в рамках гибридной нейросетевой модели знания о сетевых атаках на различные клиентские корпоративные информационные инфраструктуры, обеспечивая конфиденциальность клиентского сетевого трафика.*

Метод исследования: *для оперативной обработки и анализа сетевого трафика использованы методы машинного обучения. Применены методы построения моделей вложений и автоэнкодеров для извлечения признаков, методы построения бинарных классификаторов на основе глубоких нейронных сетей, включая сверточные нейронные сети и полносвязные сети прямого распространения. Использованы методы федеративного трансферного обучения.*

Полученные результаты: *разработан прототип интеллектуальной системы обнаружения сетевых атак и вторжений на основе федеративного трансферного обучения. Предложена архитектура системы в составе центра мониторинга информационной безопасности, приведена структурная схема серверной и клиентской компонент системы, позволяющих решать задачи сбора и предобработки данных сетевых сессий и управлять жизненным циклом моделей анализа. Приводятся результаты сравнительной оценки эффективности обнаружения специализированных сетевых атак на примере управляющего трафика ботнетов. Сравниваются бинарные классификаторы на основе полносвязных глубоких нейронных сетей прямого распространения, сверточных нейронных сетей с одномерным входным слоем, ансамблевых моделей на основе деревьев решений, гибридных автоэнкодеров со слоем вложений и сверточным классификатором в сценариях централизованного и федеративного обучения. Гибридная нейросетевая модель в режиме федеративного обучения демонстрирует наилучшие показатели (F1-мера = 0,91) благодаря эффективной схеме представления признаков, но время ее обучения существенно возрастает (в 1,5–2 раза).*

Научная новизна: *предложена гибридная нейросетевая модель классификации сетевых сессий, основанная на нейросетевых моделях вложений и моделях нейросетевых сверточных автоэнкодеров, отличающаяся алгоритмом кодирования разреженных категориальных и непрерывных признаков без использования размеченной обучающей выборки и применением федеративного трансферного обучения, что позволит обеспечить конфиденциальности данных локальных клиентов и возможность переноса обучения, а также повысить оперативность и достоверность обнаружения вредоносного сетевого трафика специалистами центров мониторинга информационной безопасности.*

Вклад авторов: *Васильев В. И. – планирование исследований в области построения систем обнаружения атак с применением методов машинного обучения, проведение сравнительного анализа результатов моделирования, подготовка аналитического обзора. Вульфин А. М. – проведение экспериментального исследования на основе разработанного программного обеспечения. Картак В. М. – подготовка аналитического обзора, планирование эксперимента, проектирование программного обеспечения. Башмаков Н. М. – подготовка данных для моделирования, интерпретация результатов исследования; обобщение результатов исследования; формулировка выводов. Кириллова А. Д. – разработка программного обеспечения, оформление рукописи статьи; работа с графическим материалом.*

Ключевые слова: *глубокое обучение, трафик управления ботнетами, сверточные нейросетевые классификаторы, автоэнкодеры, нейросетевые модели вложений.*

1 Васильев Владимир Иванович, доктор технических наук, профессор, Уфимский университет науки и технологий, г. Уфа, Россия. E-mail: vas0015@yandex.ru
2 Вульфин Алексей Михайлович, доктор технических наук, профессор, Уфимский университет науки и технологий, г. Уфа, Омский государственный технический университет, г. Омск, Россия. E-mail: vulfin.am@ugatu.su
3 Картак Вадим Михайлович, доктор физико-математических наук, профессор, Уфимский университет науки и технологий, г. Уфа, Россия. E-mail: kartak.vm@ugatu.su
4 Башмаков Наиль Маратович, аспирант, Уфимский университет науки и технологий, г. Уфа, Россия. E-mail: nail.bashmakov@gmail.com
5 Кириллова Анастасия Дмитриевна, кандидат технических наук, старший преподаватель, Уфимский университет науки и технологий, г. Уфа, Россия. E-mail kirillova.andm@gmail.com

Введение

Одной из ключевых проблем использования методов искусственного интеллекта и машинного обучения (МО) при построении систем обнаружения атак (СОА) и обнаружения вторжений (СОВ) для распределенных автоматизированных объектов и систем является проблема формирования качественных наборов обучающих данных [1]. Данные могут иметь такие осложняющие особенности, как неполнота (т.е. отсутствие конкретных данных определенного вида), дисбаланс (неравномерность распределения данных для различных классов), недостаток размеченных данных. Попытки собрать и объединить разрозненные (локальные) обучающие данные, принадлежащие разным организациям, в единую, централизованную обучающую выборку, которая будет храниться на центральном сервере (центре обработки данных), входят в противоречие с нормативными требованиями обеспечения конфиденциальности этих данных.

Одним из эффективных путей решения этой проблемы является идея федеративного обучения, впервые предложенная в работах [2]. Федеративное обучение (ФО) – это новое направление в МО, когда несколько участников (клиентов) совместно обучают свои локальные модели МО под управлением центрального сервера, при этом не сообщая ему свои обучающие данные, т.е. сохраняя их конфиденциальность. Участники только информируют центральный сервер о своих промежуточных результатах обучения (настройках моделей), а центральный сервер, в свою очередь, обрабатывает эту информацию с целью обновления собственной (глобальной) модели МО и предоставляет эти обновления всем участникам для изменения настроек их моделей. В литературе представлен ряд подробных обзоров, посвященных рассмотрению методов ФО [3, 4] и применению этих методов в задачах построения СОА [5,6].

Отличительной особенностью многих работ, посвященных построению СОА на основе ФО, является использование открытых наборов обучающих данных (датасетов), таких как NSL-KDD, CICIDS 2017 и 2018, UNSW-NB15, N-BaIoT, BoT-IT, Edge-IoT dataset и др. В зависимости от признаков проявления различных классов атак с учетом специфики конкретной предметной области и способа их использования в процессе обучения различают следующие группы методов ФО: горизонтальное ФО, вертикальное ФО и федеративное трансферное обучение. Наибольший интерес из перечисленных методов представляют методы федеративного трансферного обучения (ФТО), предложенного в 2018 г. в работе [7], в основе которого заложено объединение двух подходов: федеративного обучения (ФО) и трансферного обучения (ТО). Основная идея ТО – перенос знаний (Knowledge Transfer) с некоторой предварительно обученной модели МО на другие модели (задачи), что позволяет дообучать (fine-tuning) другие проблемно-ориентированные модели на малом наборе данных, одновременно повышая точность их обучения [8]. Использование ТО позволяет в данном случае более полно воспользоваться располагаемыми гетерогенными данными и знаниями, имеющимися в арсенале участников, для восполнения возможного недостатка данных или меток.

Настоящая статья построена следующим образом. В первой главе приведены основные положения ФТО и анализ релевантных работ по рассматриваемой тематике. Во второй главе представлена архитектура предложенной СОА на основе ФТО, рассмотрены ее основные компоненты. Третья глава содержит изложение полученных результатов моделирования СОА и сравнительную оценку ее эффективности. В заключении приведены выводы по результатам исследований и направления будущих работ.

1. Федеративное трансферное обучение. Анализ релевантных работ

Рассмотрим формальную постановку задачи федеративного обучения (ФО) [4]. Пусть имеется N клиентов (узлов, участников ФО), каждый из которых обладает собственным набором обучающих данных (датасетов) D_i. Каждый из этих наборов включает в себя определенное число объектов (образцов, samples) I_i, каждый из которых, в свою очередь, характеризуется некоторым множеством признаков (features) $\{X_i\}$ и метками (labels) $\{Y_i\}$, обозначающими принадлежность объекта I_i тому или иному классу, т.е. $Di = <I_i, X_i, Y_i>$. Обозначим через $I = \{I_i\}$ множество объектов, $X = \{X_i\}$ – множество признаков, $Y = \{Y_i\}$ – множество меток объектов, а через $\mathrm{D} = \{D_i\}$ набор обучающих данных.

Применительно к задаче построения распределенной СОА под множеством объектов обычно понимается множество сетевых потоков (сегментов сетевого трафика, подлежащего распознаванию и классификации); множество признаков X включает такие признаки, как длительность сетевого соединения, число передаваемых байтов или пакетов, используемый протокол, целевую службу и т.п.; множество Y – это множество меток типа «Норма» или с указанием конкретного типа атак (в случае аномального сетевого трафика).

При использовании традиционного машинного обучения все наборы обучающих данных объединяются в полный набор $D = D_1 \cup \ldots \cup D_N$, на котором обучается модель MD с некоторой точностью $A(MD)$. В случае ФО не происходит объединения наборов данных D_i, а глобальная модель M_{FL} обучается

на основе локально обученных моделей M_{D_i}, использующих локальные наборы D_i, причем точность полученной глобальной модели $A(M_{ML})$ должна удовлетворять следующему требованию:

$$| A(M_D) - A(M_{ML}) | \le \delta, \quad (1)$$

где δ – малая положительная величина.

В зависимости от того, как обучающие данные распределяются между N клиентами, участвующими в ФО, различают следующие категории ФО:

а) горизонтальное ФО (Horizontal Federated Learning, HFL) – наборы обучающих данных используют одно и то же множество признаков, но разные множества объектов:

$$X_i = X_j, \quad Y_i = Y_j, \quad I_i \ne I_j, \quad \forall D_i, D_j, \quad i \ne j; \quad (2)$$

б) вертикальное ФО (Vertical Federated Learning, VFL) – наборы обучающих данных используют одно и то же множество объектов, но различные множества признаков:

$$X_i \ne X_j, \quad Y_i \ne Y_j, \quad I_i = I_j, \quad \forall D_i, D_j, \quad i \ne j; \quad (3)$$

в) федеративное трансферное обучение (Federated Transfer Learning, FTL) – наборы данных отличаются как по объектам, так и по множеству признаков:

$$X_i \ne X_j, \quad Y_i \ne Y_j, \quad I_i \ne I_j, \quad \forall D_i, D_j, \quad i \ne j. \quad (4)$$

Особенность ФТО заключается в сочетании федеративного обучения, гарантирующего конфиденциальность обучающих данных клиентов, с трансферным обучением, с помощью которого осуществляется перенос знаний от одних клиентов, располагающих более богатой информацией, другим клиентам, имеющим недостаточно признаков или меток. Данная ситуация иллюстрируется рис. 1, где обучающие данные 2-х клиентов (A и B) перекрываются лишь в малой зоне как по множеству объектов (samples), так и по множеству признаков (features), но с помощью трансферного обучения происходит передача части признаков и меток от клиента B клиенту A. Таким образом, обучаемая модель распространяется на непересекающуюся область данных в A (т. е. на рис. 1 фактически заполняется правый верхний угол).

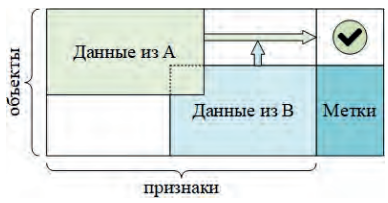


Рис. 1. Схема федеративного трансферного обучения [4]

Рассмотрению различных подходов к реализации ФТО посвящены обзоры [9]. На рис. 2 представлена типовая архитектура системы ФТО с N клиентами.

Рис. 2. Архитектура системы ФТО

Согласно рис. 2, процесс ФТО содержит следующие основные этапы:

1. **Инициализация** – на центральном сервере создается глобальная модель МО, которая предварительно обучается на открытом датасете, содержащем большое количество обучающих данных. Эта модель затем распространяется в качестве базовой модели для построения локальных моделей всех N клиентов.
2. **Локальное обучение** – каждый клиент дообучает полученную им предварительно обученную модель (fine-tuning) на своих собственных обучающих данных (локальном датасете) для решения своей конкретной задачи.
3. **Передача параметров (настроек) серверу** – результаты обучения, т.е. параметры настроек локальных моделей (веса, градиенты изменения весов) пересылаются в зашифрованном виде центральному серверу.
4. **Агрегирование** – центральный сервер усредняет полученные значения параметров локальных моделей и вносит соответствующие изменения в свою глобальную модель, обновляя ее таким образом с учетом полученной информации.
5. **Передача обновлений клиентам** – улучшенная глобальная модель возвращается клиентам, которые, в свою очередь, дообучают ее на своих обучающих данных (далее шаги 2-5 повторяются до тех пор, пока не будут достигнуты заданные показатели точности модели ФТО (см. условие (1)).

Использование ФТО при этом обеспечивает такие преимущества, как:

– сохранение конфиденциальности данных клиентов (поскольку обучающие данные клиентов остаются у них, они не передаются на центральный сервер, а передаются только параметры локальных моделей);

– эффективное использование данных (нет необходимости использования клиентами больших датасетов, публичный датасет с большим набором данных используется только на этапе предварительного обучения (pre-training) глобальной модели);
– трансфер знаний (фактически имеет место для совместного обучения локальных моделей клиентов, с переносом знаний от одних клиентов к другим);
– снижение нагрузки на сервер (распределяя вычисления по отдельным клиентам, ФТО позволяет уменьшить вычислительную нагрузку на центральный сервер).

На сегодня предложено значительное количество алгоритмов агрегирования параметров локальных моделей (настроек ФТО) [6]. Наиболее популярным из них является алгоритм федеративного усреднения FedAvg (Federated Averaging), согласно которому обновленные значения векторов параметров вычисляются по формуле

$$W_{t+1}^k = \sum_{k=1}^{N} \frac{n_k}{N} W_t^k, \quad (5)$$

где W_t^k и W_{t+1}^k – векторы настраиваемых параметров (весов, градиентов) локальной модели k-го клиента соответственно в моменты времени t и $(t + 1)$; t – дискретное время (итерация обучения); n_k – размерность используемого k-м клиентом набора обучающих данных; N – общее количество клиентов. Другие известные алгоритмы агрегирования параметров локальных моделей МО – FedSGD, FedProx, Fed+, FedCM, DWFed, FedMA [6].

Для разработки и реализации прикладных систем ФТО обычно используются специализированные проблемно-ориентированные фреймворки (frameworks). я кодом на основе языка Python, такие как Tensor Flow Federated (TFF), FATE, PFL, PySyft, FL&DP [10], а также Java-ориентированные программные продукты, например, Federated Learning for Java (FL4J) [11].

Исследованию особенностей применения ФТО для построения распределенных СОА в последние годы посвящено достаточно много публикаций. Значительное внимание, в частности, уделяется вопросам обеспечения защищенности IoT. Так, в [12] представлены результаты разработки фреймворка для построения СОА на основе принципов ФТО на примере 3-х узлов (клиентов) IoT. Произведена оценка эффективности СОА с использованием специально собранного экспериментального стенда и датасета CSE-CICIDS 2018 (с разделением этого датасета на непересекающиеся подмножества, имеющие различные признаки и метки, для отдельных клиентов). В качестве вариантов построения глобальной и локальных моделей рассмотрены полносвязная глубокая нейронная сеть (НС) и сверточная НС.

В [13] аналогичные базовые модели МО (полносвязная НС и сверточная МО) использовались при построении СОА для медицинской IoT-системы (Internet of Medical Things) с 3-мя клиентами. При обучении этих моделей с помощью ФТО предполагалось использование сервером и клиентами 4-х различных подмножеств датасета Edge-IIoT set.

В [14] представлена разработка федеративной системы IoT Defender, представляющей собой фреймворк для построения распределенной СОА на основе ФТО применительно к IoT с использованием телекоммуникаций нового поколения 5G. В качестве базовой модели (сервер и 4 клиента) рассматривается сверточная НС. Всего были использованы 5 различных датасетов: 2 публичных датасета CICIDS 2017 и NSL-KDD, а также 3 частных (специально подобранных) датасета для различных групп «умных» устройств IoT. Алгоритм агрегирования – FedAvg.

В [15] задача построения СОА на основе ФТО решалась с использованием в качестве базовой модели гибридной НС (полносвязная НС + сверточная НС). Исходный датасет – Edge-IIoT set, с разделением по различным клиентам; алгоритм агрегирования – FedSGD (Federated Stochastic Gradient Descent).

Работа [16] посвящена разработке СОА на основе ФТО для IIoT с использованием перспективных беспроводных сетей 6-го поколения (6G) – Mobile Edge Computing. На стороне сервера (глобальная модель) и клиентской стороне (локальные модели, 10 клиентов) используется сверточная НС; 2 датасета NSL-KDD и UNSW-NB15 разбиты на отдельные непересекающиеся подмножества; алгоритм агрегирования – FedSGD.

Другая группа работ, в отличие от перечисленных, использует в качестве базовых достаточно редкие классы моделей МО, пока не столь характерные для рассматриваемой предметной области. Так, в [17] при построении СОА на основе ФТО базовая модель (на серверной и клиентской стороне) выбрана в классе НС специального вида – машин экстремального обучения (Extreme Learning Machine, ELM). При обучении моделей использовались датасеты NSL-KDD, KDD99, ISCX 2012; алгоритм агрегирования – FedAvg.

В [18] глобальная и локальные модели МО строятся в классе генеративно-состязательных сетей (Generative Adversarial Networks, GAN). Клиенты представляют собой информационные системы промышленных предприятий; датасеты сформированы на основе собранных клиентами реальных данных об атаках; алгоритм агрегирования – FedAvg.

В [19] для защиты IoT от атак ботнетов предложена СОА, в которой в качестве базовой, предварительно обучаемой модели МО используется нейросетевая модель трансформера. Исходный датасет – N-BaIoT,

собран клиентами с 9 коммерческих IoT-устройств, атакуемых ботнетами Mirai и BASHLITE. Алгоритм агрегирования – FedAvg.

Все рассмотренные СОА, построенные с применением ФТО, показали высокую точность обнаружения атак по сравнению с традиционными децентрализованными СОА, в том числе при обнаружении неизвестных ранее для клиентов атак (немаркированных, отсутствующих в локальных датасетах), при сохранении конфиденциальности обучающих данных клиентов, что является главным преимуществом ФТО.

2. Архитектура системы обнаружения атак на основе федеративного трансферного обучения

Современным этапом развития комплексного подхода к обеспечению безопасности информационной инфраструктуры является создание центров мониторинга информационной безопасности (ЦМИБ, Security Operation Center, SOC). Организационно-технические процедуры ЦМИБ направлены на обнаружение и предотвращение киберугроз с учетом ключевых принципов проактивной защиты как сочетания тактической и стратегической аналитики на основе инженерии знаний и интеллектуальной обработки гетерогенных слабоструктурированных данных, получаемых из внутренних и внешних источников.

Архитектура СОА/СОВ. Гибридная архитектура распределенной СОА/СОВ, предназначенной для использования в составе ЦМИБ, представлена на рис. 3. Здесь выделены клиентские компоненты СОА/СОВ, роль которых заключается в оперативном мониторинге сетевого трафика в пределах клиентской инфраструктуры, а также серверная компонента, предназначенная для агрегации накапливаемых клиентскими компонентами знаний о реализации сетевых атак и их ключевых признаках и их интеграции с внешними базами знаний. Подобное разделение позволяет снизить как объемы передаваемых в ЦМИБ для анализа данных (минимизируя передачу чувствительных данных), так и необходимые вычислительные ресурсы.

Рассмотрим более подробно структурную организацию предлагаемой распределенной СОА/СОВ (рис. 4).

В состав клиентской компоненты системы входят следующие подсистемы:

- подсистема (IK) сбора и предобработки данных сетевых сессий – выполняет непрерывный мониторинг сетевой активности в пределах клиентской информационной инфраструктуры, размещая данные сетевых сессий в формате Netflow в хранилище на основе локальной колоночной базы данных;
- подсистема (IIK) – обеспечивает взаимодействие с локальными компонентами SIEM ЦМИБ для создания контекста сетевого взаимодействия и разметки сетевой активности на основе совокупности событий и инцидентов ИБ;
- подсистема (IIIK) подготовки данных для обучения локальных моделей анализа сетевого трафика – формирует обучающий набор данных для контролируемого обучения локальной модели в процессе ФТО;

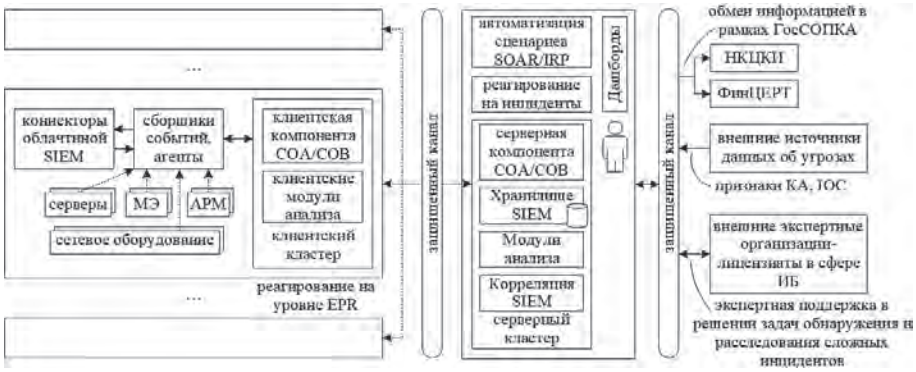


Рис. 3. Гибридная архитектура СОА/СОВ в составе ЦМИБ (МЭ – межсетевой экран, КА – компьютерные атака, IOC – индикаторы компрометации)

Рис. 4. Структурная схема клиентской и серверной компонент СОА/СОВ

– подсистема (IVK) предназначена для управления жизненным циклом цепочки моделей в процессе ФТО: «глобальная модель»-«локальная модель»-«обновления для глобальной модели».

В состав подсистемы IVK входят следующие модули:

– модуль (1K) подготовки локальной модели анализа;
– модуль (2K) оперативного анализа клиентского сетевого окружения;
– модуль (3K) управления жизненным циклом локальной модели анализа;
– модуль (4K) для обновления локальной модели на основе серверной глобальной модели, а также пересылки на сервер обновлений для глобальной модели в рамках процесса ФТО.

Серверная компонента распределенной СОА/СОВ включает в себя следующие подсистемы:

– подсистема (IC) и подсистема (VC) обеспечивают сбор, предобработку и систематизацию данных сетевого взаимодействия, полученных из открытых источников (наборы данных PCAP с разметкой) и доступных клиентских сессий в серверном хранилище;
– подсистема (IIIC) обеспечивает верификацию и валидацию собираемых данных с целью обнаружения возмож ных атак на модель ФТО;
– подсистема (IIC) в ходе взаимодействия с базой знаний ЦМИБ позволяет создавать и обогащать контекст собираемых сетевых сессий;
– подсистема (IVC) реализует серверную часть процесса ФТО.

В состав подсистемы IVC серверной компоненты входят: модуль (1C) подготовки и обновления глобальной модели, модуль двустороннего взаимодействия с локальными моделями (2C) и модуль (3C) для управления жизненным циклом моделей ФТО.

Гибридная нейросетевая модель классификации сетевых сессий на основе ФТО. Основные трудности классификации сетевых сессий заключаются в следующем:

– существенный дисбаланс количества доступных сетевых сессий обычного взаимодействия конечных систем и подтвержденной вредоносной сетевой активности [20];
– неидентичность и зависимость в распределении данных (nonIDD) [1, 21], собираемых с различных клиентских инфраструктур;
– разнообразие способов выделения и кодирования ключевых признаков;
– необходимость учитывать как «дрейф данных», так и «дрейф концепции» для оценки горизонта пригодности обучаемых моделей [22];
– проверка эффективности работы системы при использовании нескольких наборов данных с разными способами реализации атак одного класса;
– как правило, основные результаты в известных работах получены с использованием различных типов широко распространенных сетевых атак, вопросы обнаружения узкоспециализированных атак анализируются очень редко.

Исходя из вышеперечисленного, предлагается использовать гибридную нейросетевую модель классификации сетевых сессий (рис. 5).

Блок (1) в составе модели обобщает предложенный в [23–25] способ представления разреженных категориальных и непрерывных численных переменных в виде компактного векторного представления. Отличительной особенностью является возможность эффективного кодирования признаков разного типа

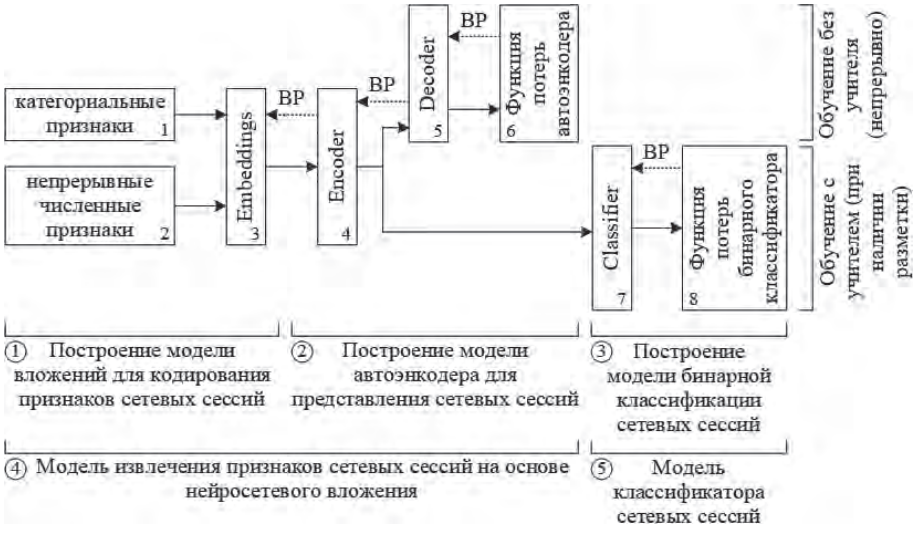


Рис. 5. Гибридная нейросетевая модель классификации сетевых сессий

(особенно важно при использовании глубоких нейросетевых моделей) с помощью модели вложений, настройка параметров которой осуществляется с помощью автоэнкодера (2), для обучения которого не требуется размеченной выборки – схема построения «автоэмбеддера» [26]. Обучение слоев вложений (Embedder) и симметричных слоев автоэнкодера (Encoder-Decoder) позволяет использовать все доступные сетевые сессии – нормальной работы, различных типов атак и т.д., игнорируя дисбаланс распределения примеров по классам. Блок (3) представляет собой классификатор, использующий в качестве вектора признаков подготовленные входными слоями высокоуровневые признаки. «Заморозка» параметров входных слоев позволяет дообучить слои классификатора на имеющихся, ограниченных по количеству, размеченных сетевых сессиях. Таким образом, гибридная нейросетевая модель классификации сетевых сессий включает две составляющие: модель извлечения признаков сетевых сессий (4) и модель классификатора (5).

Федеративное трансферное обучение гибридной нейросетевой модели классификации сетевых сессий. Рассмотрим схему обучения предлагаемой гибридной нейросетевой модели (рис. 6) в схеме ФТО:

Шаг 0. На основе доступных данных с частичной разметкой на сервере создаются и обучаются последовательно два блока модели: автоэнкодер (Embedder и Encoder-Decoder – используются все доступные верифицированные данные без разметки) и классификатор (на ограниченном размеченном наборе обучается блок Classifier).

Шаг 1. Обученная глобальная модель передается по защищенным каналам на клиентские компоненты.

Шаг 2. Клиентские модели на локальных данных поэтапно продолжают обучение блока автоэнкодера (повышая эффективность извлечения признаков) и, при наличии размеченных данных, – блока классификатора. По истечении заданного количества итераций локального обучения выполняется передача оценок градиентов на сервер.

Шаг 3. Полученные оценки градиентов агрегируются на сервере и используются для трансферного обучения блоков глобальной модели. Весовые коэффициенты глобальной модели передаются клиентским моделям.

Шаг 4. Процедура продолжается либо до достижения сходимости оценок (FedAVG), либо по достижении заданных критериев (FedAVG+), обеспечивая устойчивость к неидентичности и зависимости в распределении данных (non-IID) на клиентских подсистемах.

3. Оценка эффективности СОА

Для оценки эффективности работы прототипа распределенной СОА/СОВ на основе ФТО обратимся к задаче обнаружения сетевого трафика командных центров ботнетов на ранних стадиях проникновения в корпоративные информационные инфраструктуры. Сетевой трафик инфраструктур управления и контроля ботнетов (Command & Control, C&C) характеризует взаимодействие специализированных серверов злоумышленника со скомпрометированными устройствами и является узкоспециализированной сетевой атакой, обнаружение которой сопряжено с рядом трудностей [14].

Для серии экспериментов были выбраны наборы данных NF-UNSW-NB15 и NF-CSE-CIC-IDS2018, преобразованные к единому формату представления признаков NetFlow, а также специализированные наборы

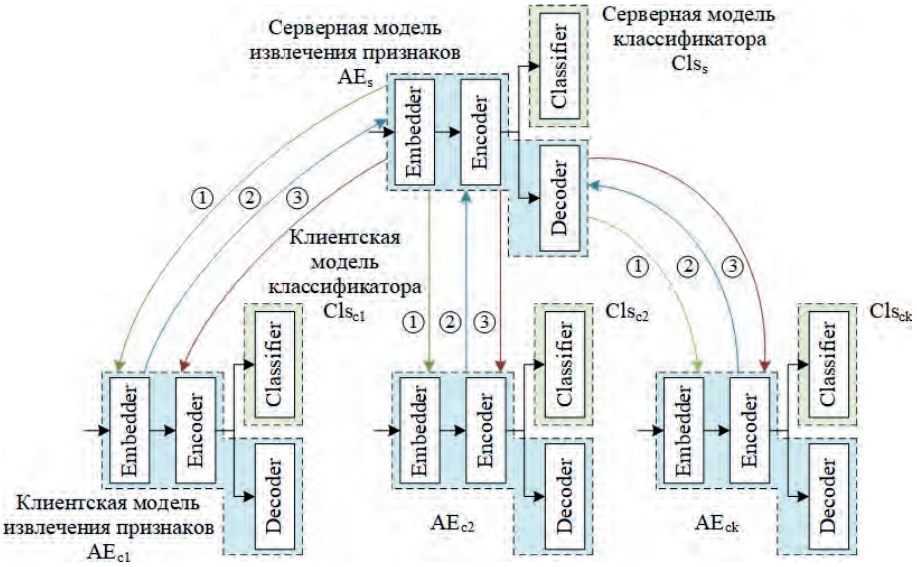


Рис. 6. Схема федеративного трансферного обучения гибридной нейросетевой модели

Таблица 1.

Характеристики наборов данных C&C

Название набора данных	Источник	Количество признаков	Тип разметки	Количество сетевых сессий	Роль в схеме обучения моделей
NF-UNSW-NB15	Университет Нового Южного Уэльса (UNSW), Австралия	43	Netflow	Benign – 1550712 Backdoor – 1782	Тестовое множество для имитации новых реализаций атаки
BH-KSU23	Университет имени Короля Сауда (KSU), Саудовская Аравия	79	CICFlowmeter	Benign – 257691 Malicious – 209539	Обучение базовой серверной модели
NF-CSE-CIC-IDS2017	Канадский институт кибербезопасности (CIC), Канада	43	Netflow	Benign – 7373198 Bot – 15683	Имитация данных «Клиент 1»
Trojan Detection	Университет Дрексела (Drexel), США	79	CICFlowmeter	Benign – 86799 Trojan – 90683	Имитация данных «Клиент 2»
MTA-KDD19	Университет Л'Акуила (L'Aquila), Италия	50	Netflow + модификации	Benign – 31926 Malware – 39544	Имитация данных «Клиент 3»

данных BH-KSU23, Trojan Detection и MTA-KDD19, содержащие сессии нормальной работы и размеченный трафик для 15 схем реализации C&C взаимодействия (табл. 1).

Проекция исходного признакового пространства с помощью алгоритма UMAP (Uniform Manifold Approximation and Projection) для сетевых сессий нормальной работы и C&C трафика из подмножеств NF-UNSW-NB15 и NF-CSE-CIC-IDS2017 приведена на рис. 7.

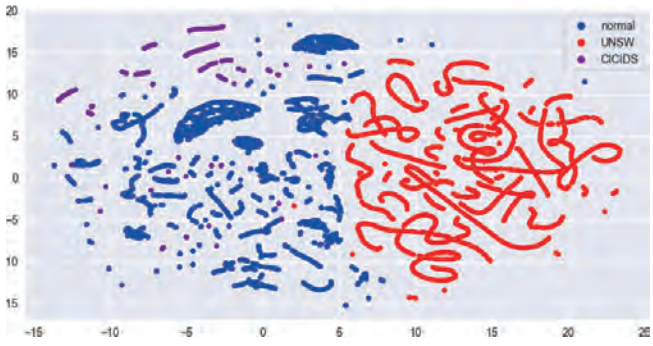


Рис. 7. Проекция исходного признакового пространства для сетевых сессий нормальной работы и C&C трафика двух наборов

Как видно, сетевые сессии, помеченные как «нормальный трафик», располагаются компактно, выраженными группами. Данные C&C из набора данных UNSW достаточно хорошо отделимы от обычного трафика. Данные C&C из набора CICIDS, напротив, отделимы хуже и перемешаны с примерами нормальных сетевых сессий.

Схема проведения серии экспериментов включает построение четырех типов моделей бинарных классификаторов (табл. 2).

Модели DNN, CNN1D построены в данном случае как с помощью ФТО (эксперименты 1, 3), так и в качестве одиночных глобальных моделей (эксперименты 2, 4), обучаемых на объединенном наборе данных. Модель XGBoost строится только как глобальная модель (эксперимент 7), а модель AE-CNN1D – только с помощью ФТО (эксперименты 5 и 6).

Из всех имеющихся наборов данных существенный дисбаланс по числу примеров в классах наблюдается только для NF-UNSW-NB15 и NF-CSE-CIC-IDS2017 – для них применяется схема случайного удаления примеров мажоритарного класса. Обучающая выборка после семплирования в каждом из наборов включает 75 % примеров, тестовая – 25 %. Кодировщики категориальных и непрерывных признаков для всех моделей, кроме AE-CNN1D: RS – Robust Scaler, OE – Ordinal Encoder, OHE – One hot encoder. С помощью фреймворка Optuna выполнена предварительная оптимизация гиперпараметров бинарных классификаторов на основе алгоритма TPE.

В качестве алгоритма передачи обновлений градиентов в ходе обучения локальных моделей

Таблица 2.

Схема проведения серии экспериментов

Модель	Характеристика модели	Параметры модели		Схема построения	Особенность набора данных	№ эксперимента
DNN	Полносвязная глубокая нейронная сеть прямого распространения	Количество слоев	6	ФТО	Сбалансированный	1
		Количество нейронов по слоям	98, 128, 64, 32, 4, 1			
		Исключение (dropout)	3, 4 слои	Глобальная	Сбалансированный	2
		Функция активации	ReLU, последний – sigmoid			
		Функция потерь	Binary Cross-Entropy With Logits			
		Коэффициент скорости обучения	0,085			
		Количество эпох обучения	64			
CNN1D	Сверточная нейронная сеть с входным слоем в виде одномерного кортежа	Слои: conv1d, dropout, conv1d, dropout, flatten, dropout, batch_normalization, fully_connected, fully_connected, fully_connected		ФТО	Сбалансированный	3
		Размер ядер свертки	3			
		Количество фильтров	5	Глобальная	Сбалансированный	4
		Параметр исключения (dropout) по слоям	0,1, 0,1			
		Количество нейронов в полносвязных слоях	128, 64, 1			
		Функция активации	ReLU, последний – sigmoid			
		Функция потерь	Binary Cross-Entropy With Logits			
		Количество эпох обучения	32			
AE-CNN1D	Гибридный автоэнкодер со слоем вложений в сочетании с классификатором на основе полносвязной сети с пакетной нормализацией	Слои Encoder/Decoder (симметрично): (embeddings1 + embeddings2), concat_embeddings, conv1d, dropout, conv1d, dropout, flatten		ФТО	Сбалансированный	5
		Слои Classifier: flatten, dropout, batch_normalization, fully_connected, fully_connected, fully_connected				
		Размер ядер свертки	3		Не сбалансированный + все доступные сессии прочих атак – для автоэнкодера	6
		Количество фильтров	5			
		Параметр исключения (dropout) по слоям	0,1, 0,1			
		Количество нейронов в полносвязных слоях	128, 64, 1			
		Функция активации	ReLU, последний слой Classifier – sigmoid			
		Количество эпох обучения автоэнкодера	32			
		Количество эпох обучения классификатора	16			
XGBoost	Ансамблевый метод объединяет слабые модели на основе деревьев решений	Максимальная глубина дерева	16	Глобальная	Сбалансированный	7
		Скорость обучения	0,029			
		Количество слабых классификаторов в ансамбле	316			
		Соотношение подвыборок обучающих экземпляров	0,997			
		Коэффициенты регуляризации L1, L2	1,391, 2,840			

Таблица 3.

Параметры сервера для запуска моделей

Параметр	Характеристика
GPU	4 GPU Tesla V100
Объем видеопамяти GPU	128 ГБ
Процессор	Intel Xeon E5-2698 v4 2,2 ГГц (20-ядерный)
Объем оперативной памяти	256 ГБ RDIMM DDR4

использована модификация алгоритма FedAvg+ в версии [8] фреймворка FATE. Размер пакета при обучении серверной и клиентских моделей составляет 64, агрегация локальных градиентов осуществлялась через каждые 2 эпохи обучения моделей.

Для обучения использовался высокопроизводительный сервер, параметры которого представлены в табл. 3. Каждая из моделей в изолированном окружении использовала выделенные GPU, обмен данными между моделями осуществлялся через каналы в RAM.

При оценке качества бинарной классификации были использованы следующие метрики:

– Precision (Точность) – доля правильно предсказанных положительных случаев среди всех предсказанных положительных случаев;
– Recall (Полнота) – доля правильно предсказанных положительных случаев среди всех реальных положительных случаев;
– F1-мера – является гармоническим средним точности и полноты.

Результаты оценки качества моделей приведены в табл. 4.

Схема с централизованным обучением моделей на всех имеющихся данных (2, 4, 7) продемонстрировала ожидаемые высокие показатели F1-меры. Причем, сверточная модель по показателям F1-меры несущественно опережает классическую полносвязную DNN; модель XGBoost в задаче опережает на отдельных тестовых наборах данных нейросетевые модели, но лишена преимуществ обучения распределенных моделей.

В целом, применение схемы федеративного обучения оказалось весьма успешным: модель способна классифицировать трафик исходного набора данных (BH-KSU23), клиентских наборов данных и «новых» сетевых сессий (NF-UNSW-NB15) командного трафика ботнетов.

С точки зрения повышения эффективности классификации оказалось целесообразным использовать гибридную нейросетевую модель AE-CNN1D – модель демонстрирует наилучшие показатели благодаря эффективной схеме представления признаков, но время

Таблица 4.

Основные результаты серии экспериментов

Набора данных	Метрики	ФТО				Глобальная модель		
		DNN	CNN1D	AE-CNN1D		DNN	CNN1D	XGB
	Эксперимент	1	3	5	6	2	4	7
	Время обучения, мин	98	173	247	362	24	51	16
BH-KSU23 (тестовая)	Precision	0,9605	0,9674	0,9708	0,9963	0,9612	0,9771	0,972
	Recall	0,9689	0,9721	0,972	0,9936	0,9703	0,9859	0,9723
	F1-мера	0,9647	0,9698	0,9714	0,995	0,9654	0,9815	0,9722
NF-CSE-CIC-IDS2017 (тестовая)	Precision	0,9737	0,9755	0,9811	0,9962	0,9755	0,9771	0,9788
	Recall	0,9629	0,9723	0,9867	0,9934	0,9652	0,9778	0,9773
	F1-мера	0,9683	0,9739	0,9839	0,9948	0,9703	0,9774	0,9781
Trojan Detection (тестовая)	Precision	0,8558	0,8623	0,892	0,9086	0,8569	0,8673	0,8854
	Recall	0,8512	0,86	0,9067	0,9175	0,8534	0,863	0,9103
	F1-мера	0,8535	0,8612	0,8993	0,913	0,8551	0,8652	0,8977
MTA-KDD19 (тестовая)	Precision	0,987	0,9883	0,9941	0,9956	0,9876	0,9894	0,9921
	Recall	0,9893	0,994	0,9969	0,9986	0,99	0,9953	0,9961
	F1-мера	0,9881	0,9911	0,9955	0,9971	0,9888	0,9923	0,9941
NF-UNSW-NB15 (весь набора)	Precision	0,8547	0,8872	0,9416	0,9888	0,8636	0,9108	0,8872
	Recall	0,7484	0,7889	0,9149	0,988	0,7589	0,8059	0,8907
	F1-мера	0,798	0,8352	0,9281	0,9884	0,8079	0,8551	0,8889

ее обучения существенно возрастает. Однако можно прогнозировать стабильную работу модели при увеличении количества клиентских моделей и объемов доступных сетевых сессий, т.к. основным затруднением является именно обработка неидентичных (non-IID) данных. Построение эффективных векторов вложений позволит избежать дальнейшего повышения сложности классификатора.

Заключение

На основании проведенного анализа источников литературы для повышения эффективности систем обнаружения сетевых атак и вторжений в корпоративных информационных системах предлагается использовать модели и алгоритмы федеративного трансферного обучения.

Разработан прототип интеллектуальной системы обнаружения сетевых атак и вторжений. Предложена архитектура СОА/СОВ в составе ЦМИБ, приведена структурная схема серверной и клиентской компонент системы, позволяющих решать задачи сбора и предобработки данных сетевых сессий, обеспечивать взаимодействие с ЦМИБ и управлять жизненным циклом моделей.

Предложена гибридная нейросетевая модель классификации сетевых сессий, включающая автоэнкодер, блоки построения вложений и классификатор. Отличительной особенностью является возможность эффективного кодирования разреженных категориальных и непрерывных признаков без использования размеченной обучающей выборки.

Результаты проведенных вычислительных экспериментов позволяют сделать выводы о высокой эффективности обнаружения специализированных сетевых атак на примере C&C трафика с помощью предложенного прототипа СОА/СОВ. Применение федеративного трансферного обучения обеспечивает при этом как сохранение конфиденциальности данных локальных клиентов, так и возможность переноса обучения – аккумулирования знаний о проводимых атаках на различные информационные инфраструктуры в рамках единой гибридной нейросетевой модели, что позволяет повысить оперативность и достоверность обнаружения вредоносного сетевого трафика, и тем самым, повысить защищенность клиентских корпоративных информационных систем.

Благодарности.
Работа выполнена в ОмГТУ в рамках государственного задания Минобрнауки России на 2023–2025 годы № FSGF-2023-0004.

Литература

1. *Wagle S. et al. Embedding alignment for unsupervised federated learning via smart data exchange // GLOBECOM 2022-2022 IEEE Global Communications Conference. IEEE, 2022, pp. 492–497.*
2. *McMaham H. B. et al. Communication-Efficient Learning of Deep Networks from Decentralized Data // arXiv preprint arXiv: 1602.05629 [cs.LG]. 2023. DOI: 10.48550/arXiv.1602.05629.*
3. *Wen J. et al. A Survey on Federated Learning: challenges and applications // International Journal of Machine Learning and Cybernetics. 2023, vol. 14, pp. 513–535.*
4. *Yang Q. et al. Federated Machine Learning: concept and applications // ACM Transactions on Intelligent Systems and Technology (TIST). 2019, vol. 10, no. 2, pp. 1–19. DOI: 10.1145/3298981.*
5. *Новикова Е. С. и др. Обнаружение вторжений на основе федеративного обучения: архитектура системы и эксперименты // Вопросы кибербезопасности. 2023, №6 (58). С. 50–66. DOI: 10.21681/2311-3456-2023-6-50-66.*
6. *Новикова Е. С. и др. Аналитический обзор подходов к обнаружению вторжений, основанных на федеративном обучении: преимущества использования и открытые задачи // Информатика и автоматизация. 2023. Т. 22, № 5. С. 1034–1082. DOI: 10.15622/ia.22.5.4.*
7. *Hernandez-Ramos J. L. et al. Intrusion Detection based on Federated Learning: a systematic review // arXiv preprint arXiv:2308.09522. 2023. DOI: 10.48550/arXiv.2308.09522.*
8. *Liu Y. et al. A secure federated transfer learning framework // IEEE Intelligent Systems. 2020 vol. 35, no. 4, pp. 70–82. DOI: 10.1109/MIS.2020.2988525.*
9. *Guo W. et al. A Comprehensive Survey of Federated Thansfer Learning: Challenges, Methods and Applications // arXiv preprint arXiv:2403.01387. 2024. DOI: 10.48550/arXiv.2403.01387.*
10. *Kholod I. et al. Open-Source Federated Learning Frameworks for IoT: A Comparative Review and Analysis // Sensors. 2020, no. 21, pp. 167. DOI: 10.3390/521010167.*
11. *Ефремов М. А., Холод И. И. Разработка архитектуры универсального фреймворка федеративного обучения // Программные продукты и системы. 2022. Т. 35, № 2. С. 263–273. DOI: 10.15827/0236-235X.138.263-272.*
12. *Otoum K., Yadlappali S. K., Nayk A. FTLIoT: A Federated Thansfer Learning Framework for Securing IoT // GLOBECOM 2022-2022 IEEE Global Communications Conference. IEEE, 2022, pp. 1146–1151. DOI: 10.1109/GLOBECOM48099.2022.10001461.*
13. *Otoum K., Chamola V., Nayak A. Federated and Transfer Learning – Empowered Intrusion Detection for IoT Applications // IEEE Internet of Things Magazine. 2022, vol. 5, no. 3, pp. 50–54. DOI: 10.1109/IOTM.001.2200048.*

14. Fan Y. et al. IoTDefender: A Federated Transfer Learning Intrusion Detection Framework for 5G IoT // 2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE). IEEE, 2020, pp. 88–95. DOI:10.1109/BigDataSE50710.2020.00020.
15. Rajesh L. T. et al. Give and Take: Federated Transfer Learning for Industrial IoT Network Intrusion Detection // 2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE, 2023, pp. 2365–2371. DOI: 10.1109/TrustCom60117.2023.00333.
16. Cheng Y. et al. Federated Transfer Learning with Client Selection for Intrusion Detection in Mobile Edge Computing // IEEE Communications Letters. 2022, vol. 26, no. 3, pp. 552–556. DOI:10.1109/LCOMM.2022.3140273.
17. Wang K., Li J., Wu W. An efficient intrusion detection method based on federated transfer learning and an extreme learning machine with privacy preservation // Security and Communication Networks. 2022, vol. 2022, no. 1, pp. 2913293. DOI:10.1155/2022/291329.
18. Guo W. et al. Federated transfer learning for auxiliary classifier generative adversarial networks: framework and industrial application // Journal of intelligent manufacturing. 2024, vol. 35, no. 4, pp. 1439–1454.
19. Metwaly A. A., Elhenawy I. Protecting IoT Devices from BotNet threats: a federated machine learning solution // Sustainable Machine Intelligence Journal. 2023, vol. 2, pp. 1–12. DOI:10.61185/SMIJ.2023.22105.
20. Azizjon M., Jumabek A., Kim W. 1D CNN based network intrusion detection with normalization on imbalanced data // 2020 international conference on artificial intelligence in information and communication (ICAIIC). IEEE, 2020, pp. 218-224. DOI:10.1109/ICAIIC48513.2020.9064976.
21. Новикова Е. С., Чен Я., Мелешко А. В. Методы оценки уровня разнородности данных в федеративном обучении // XXVII Международная конференция по мягким вычислениям и измерениям (SCM'2024) (Санкт-Петербург, 22-24 мая 2024). 2024. С. 446–450.
22. Yang Z. et al. A systematic literature review of methods and datasets for anomaly-based network intrusion detection // Computers & Security. 2022, vol. 116, pp. 102675. DOI:10.1016/j.cose.2022.102675.
23. Lee G. et al. Network Intrusion Detection with Improved Feature Representation // 2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). IEEE, 2021, pp. 2049–2054.
24. He Y., Yan D., Chen F. Hierarchical federated learning with local model embedding // Engineering Applications of Artificial Intelligence. 2023, vol. 123, pp. 106148. DOI:10.1016/j.engappai.2023.106148.
25. Sivasubramanian A., Devisetty M., Bhavukam P. Feature Extraction and Anomaly Detection Using Different Autoencoders for Modeling Intrusion Detection Systems // Arabian Journal for Science and Engineering. 2024, pp. 1–13.
26. Wang Z. X. et al. Network traffic classification based on federated semi-supervised learning // Journal of Systems Architecture. 2024, vol. 149, pp. 103091. DOI: 10.1016/j.sysarc.2024.103091.

DISTRIBUTED NETWORK ATTACK DETECTION SYSTEM BASED ON FEDERATE TRANSFER LEARNING

Vasilyev V. I.[6], Vulfin A. M.[7], Kartak V. M.[8], Bashmakov N. M.[9], Kirillova A. D.[10]

Purpose: *Improving the efficiency of detecting botnet network attacks through the use of federated transfer learning. This makes it possible to accumulate knowledge about network attacks on various client corporate information infrastructures within the framework of a hybrid neural network model, ensuring the confidentiality of client network traffic.*

Methods: *Machine learning methods were used for operational processing and analysis of network traffic. Methods for constructing embedding models and autoencoders for feature extraction, methods for constructing binary classifiers based on deep neural networks, including convolutional neural networks and fully connected feedforward networks, are applied. Federated transfer learning methods were used.*

Research results: *A prototype of an intelligent system for detecting network attacks and intrusions based on federated transfer learning was developed. The architecture of the system as part of the information security monitoring center is proposed. The structural diagram of the server and client components of the system is given. The components allow solving the problems of collecting and preprocessing network session data and managing the life cycle of analysis models. The results of a comparative assessment of the effectiveness of detecting specialized network attacks are presented using the example of botnet control traffic. Binary classifiers based on fully connected deep feedforward neural networks, convolutional neural networks with a one-dimensional input layer, ensemble models based on decision trees, hybrid autoencoders with an embedding layer and a convolutional classifier are compared in centralized and federated learning scenarios. The hybrid neural network model in the federated learning mode demonstrates the best performance (F1-measure = 0.91) due to the effective feature representation scheme, but its training time increases significantly (by 1.5–2 times).*

The scientific novelty: *A hybrid neural network model for classifying network sessions is proposed, based on neural network embedding models and neural network convolutional autoencoder models. The neural network model is distinguished by an algorithm for encoding sparse categorical and continuous features without using a labeled training sample and by the use of federated transfer learning. This ensures the confidentiality of local client data and the ability to transfer training, as well as increases the speed and reliability of detecting malicious network traffic by specialists at information security monitoring centers.*

6 Vladimir I. Vasilyev, Dr.Sc. (of Tech.), Professor, Ufa University of Science and Technology, Ufa, Russia. E-mail: vas0015@yandex.ru
7 Alexey M. Vulfin, Dr.Sc. (of Tech.), Professor, Ufa University of Science and Technology, Ufa, Omsk State Technical University, Omsk, Russia. E-mail: vulfin.am@ugatu.su
8 Vadim M. Kartak, Dr.Sc. (in Physics and Math.), Professor, Ufa University of Science and Technology, Ufa, Russia. E-mail: kartak.vm@ugatu.su
9 Nail M. Bashmakov, Post-Graduate Student, Ufa University of Science and Technology, Ufa, Russia. E-mail: nail.bashmakov@gmail.com
10 Anastasia D. Kirillova, Ph.D. (of Tech.), Senior Lecturer, Ufa University of Science and Technology, Ufa, Russia. E-mail: kirillova.andm@gmail.com

Authors' contributions: *Vasilyev V. I. – planning research in the field of building attack detection systems using machine learning methods, conducting a comparative analysis of modeling results, preparing an analytical review. Vulfin A. M. – conducting an experimental study based on the developed software. Kartak V. M. – preparation of analytical review, experimental planning, software design. Bashmakov N. M. – preparation of data for modeling, interpretation of research results; generalization of research results; formulation of conclusions. Kirillova A. D. – software development, article manuscript design; work with graphic material.*

Keywords: *deep learning, botnet control traffic, convolutional neural network classifiers, autoencoders, neural network models of embeddings.*

References

1. *Wagle S. et al. Embedding alignment for unsupervised federated learning via smart data exchange // GLOBECOM 2022-2022 IEEE Global Communications Conference. IEEE, 2022, pp. 492–497.*
2. *McMaham H. B. et al. Communication-Efficient Learning of Deep Networks from Decentralized Data // arXiv preprint arXiv: 1602.05629 [cs.LG]. 2023. DOI: 10.48550/arXiv.1602.05629.*
3. *Wen J. et al. A Survey on Federated Learning: challenges and applications // International Journal of Machine Learning and Cybernetics. 2023, vol. 14, pp. 513–535.*
4. *Yang Q. et al. Federated Machine Learning: concept and applications // ACM Transactions on Intelligent Systems and Technology (TIST). 2019, vol. 10, no. 2, pp. 1–19. DOI: 10.1145/3298981.*
5. *Novikova E. S. et al. Federated Learning Based Intrusion Detection: System Architecture and Experiments // Voprosy kiberbezopasnosti. 2023, no. 6 (58), pp. 50–66. DOI: 10.21681/2311-3456-2023-6-50-66.*
6. *Novikova E. S. et al. Analytical review of intelligent intrusion detection systems based on federated learning: advantages and open challenges // Informatics and Automation. 2023, vol. 22, no. 5, pp. 1034–1082. DOI: 10.15622/ia.22.5.4.*
7. *Hernandez-Ramos J. L. et al. Intrusion Detection based on Federated Learning: a systematic review // arXiv preprint arXiv:2308.09522. 2023. DOI: 10.48550/arXiv.2308.09522.*
8. *Liu Y. et al. A secure federated transfer learning framework // IEEE Intelligent Systems. 2020 vol. 35, no. 4, pp. 70–82. DOI: 10.1109/MIS.2020.2988525.*
9. *Guo W. et al. A Comprehensive Survey of Federated Thansfer Learning: Challenges, Methods and Applications // arXiv preprint arXiv:2403.01387. 2024. DOI: 10.48550/arXiv.2403.01387.*
10. *Kholod I. et al. Open-Source Federated Learning Frameworks for IoT: A Comparative Review and Analysis // Sensors. 2020, no. 21, pp. 167. DOI: 10.3390/521010167.*
11. *Efremov M. A., Kholod I. I., Developing universal framework design for federated learning // Software & systems. 2022, vol. 35, no. 2, pp. 263–273. DOI: 10.15827/0236-235X.138.263-272.*
12. *Otoum K., Yadlappali S. K., Nayk A. FTLIoT: A Federated Thansfer Learning Framework for Securing IoT // GLOBECOM 2022-2022 IEEE Global Communications Conference. IEEE, 2022, pp. 1146–1151. DOI: 10.1109/GLOBECOM48099.2022.10001461.*
13. *Otoum K., Chamola V., Nayak A. Federated and Transfer Learning – Empowered Intrusion Detection for IoT Applications // IEEE Internet of Things Magazine. 2022, vol. 5, no. 3, pp. 50–54. DOI: 10.1109/IOTM.001.2200048.*
14. *Fan Y. et al. IoTDefender: A Federated Transfer Learning Intrusion Detection Framework for 5G IoT // 2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE). IEEE, 2020, pp. 88–95. DOI:10.1109/BigDataSE50710.2020.00020.*
15. *Rajesh L. T. et al. Give and Take: Federated Transfer Learning for Industrial IoT Network Intrusion Detection // 2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE, 2023, pp. 2365–2371. DOI: 10.1109/TrustCom60117.2023.00333.*
16. *Cheng Y. et al. Federated Transfer Learning with Client Selection for Intrusion Detection in Mobile Edge Computing // IEEE Communications Letters. 2022, vol. 26, no. 3, pp. 552–556. DOI: 10.1109/LCOMM.2022.3140273.*
17. *Wang K., Li J., Wu W. An efficient intrusion detection method based on federated transfer learning and an extreme learning machine with privacy preservation // Security and Communication Networks. 2022, vol. 2022, no. 1, pp. 2913293. DOI: 10.1155/2022/291329.*
18. *Guo W. et al. Federated transfer learning for auxiliary classifier generative adversarial networks: framework and industrial application // Journal of intelligent manufacturing. 2024, vol. 35, no. 4, pp. 1439–1454.*
19. *Metwaly A. A., Elhenawy I. Protecting IoT Devices from BotNet threats: a federated machine learning solution // Sustainable Machine Intelligence Journal. 2023, vol. 2, pp. 1–12. DOI: 10.61185/SMIJ.2023.22105.*
20. *Azizjon M., Jumabek A., Kim W. 1D CNN based network intrusion detection with normalization on imbalanced data // 2020 international conference on artificial intelligence in information and communication (ICAIIC). IEEE, 2020, pp. 218–224. DOI: 10.1109/ICAIIC48513.2020.9064976.*
21. *Novikova E. S., Chen Ya., Meleshko A. V. Methods for Assessing the Level of Data Heterogeneity in Federated Learning // XXVII International Conference on Soft Computing and Measurements (SCM'2024) (Saint Petersburg, May 22–24, 2024). 2024, pp. 446–450.*
22. *Yang Z. et al. A systematic literature review of methods and datasets for anomaly-based network intrusion detection // Computers & Security. 2022, vol. 116, pp. 102675. DOI: 10.1016/j.cose.2022.102675.*
23. *Lee G. et al. Network Intrusion Detection with Improved Feature Representation // 2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). IEEE, 2021, pp. 2049–2054.*
24. *He Y., Yan D., Chen F. Hierarchical federated learning with local model embedding // Engineering Applications of Artificial Intelligence. 2023, vol. 123, pp. 106148. DOI: 10.1016/j.engappai.2023.106148.*
25. *Sivasubramanian A., Devisetty M., Bhavukam P. Feature Extraction and Anomaly Detection Using Different Autoencoders for Modeling Intrusion Detection Systems // Arabian Journal for Science and Engineering. 2024, pp. 1–13.*
26. *Wang Z. X. et al. Network traffic classification based on federated semi-supervised learning // Journal of Systems Architecture. 2024, vol. 149, pp. 103091. DOI: 10.1016/j.sysarc.2024.103091.*

МАСКИРОВАНИЕ ТОПОЛОГИЧЕСКИХ СВОЙСТВ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ В УСЛОВИЯХ СЕТЕВОЙ РАЗВЕДКИ. Часть 1

Горбачев А. А.[1]

DOI: 10.21681/2311-3456-2024-6-130-139

Цель исследования: *исследование моделей случайных графов и генетических алгоритмов для решения задачи синтеза ложной структуры для маскирования топологических свойств вычислительных сетей при генерации ложного сетевого трафика и применении ложных сетевых информационных объектов, с учетом степени сходства топологических свойств реальных вычислительных сетей с ложными, а также с учетом показателя защищенности вычислительных сетей.*

Используемые методы: *генетический алгоритм оптимизации, метод линейной свертки, модель Эрдеша-Реньи, Барбаши, Харари.*

Результат исследования: *синтез ложной структуры вычислительной сети на основе моделей случайных графов и эволюционных алгоритмов оптимизации позволяет повысить результативность защиты вычислительной сети за счет снижения возможностей злоумышленника по идентификации ее критических узлов посредством анализа сетевого трафика. В качестве показателя близости топологических характеристик вычислительных сетей выступает коэффициент Жаккара между множествами ребер истинной и ложной вычислительных сетей, а в качестве аппроксимации дистанции между истинными и ложными критическими узлами выступает среднее кратчайшее расстояние. Генетические алгоритмы позволяют решить задачу оптимальной параметризации моделей случайных графов с точки зрения выбранной функции приспособленности, а также при явной комбинаторной оптимизации ложной топологии. Экспоненциальный рост переборного пространства не позволяет решать задачу комбинаторной оптимизации матрицы смежности графа, характеризующего топологию вычислительной сети большого размера, что приводит к необходимости использования методов снижения размерности и параметрических моделей при маскировании топологических свойств составных вычислительных сетей.*

Научная новизна: *заключается решении задачи синтеза топологических свойств ложной вычислительной сети с использованием генетических алгоритмов и моделей случайных графов, параметризованных с учетом скалярной целевой функции приспособленности, включающей показатель близости ложной и истинной топологической структуры вычислительной сети, а также аппроксимацию расстояния между истинными и ложными критическими узлами вычислительной сети.*

Ключевые слова: *анализ сетевого трафика, проактивная защита, ложные сетевые информационные объекты, эволюционные алгоритмы оптимизации, критические узлы.*

Введение

Критически важным этапом реализации кибератак является сетевая (компьютерная) разведка (*network reconnaissance*). Определение структурно-функциональных характеристик узлов вычислительной сети (*IP*-адресов, *MAC*-адресов, состояний *TCP*, *UDP*-портов), версий программного обеспечения, служб и сервисов, версий используемых сетевых протоколов, настроек межсетевых экранов и средств антивирусной защиты, а также других сведений, позволяет выявить уязвимости или подобрать соответствующие алгоритмы для реализации сетевой (компьютерной) атаки [1, 2].

Большинство вычислительных сетей, функционирующих в интересах различных организаций имеют статичную структуру (топологию), которая может быть идентифицирована с использованием различных методов сетевой разведки. Наиболее распространенным методом построения топологии вычислительной сети является анализ сетевого трафика (сниффинг, прослушивание), проходящего через интерфейсы коммутационного оборудования, прокси-серверов и другие контролируемые злоумышленниками узлы вычислительной сети, а также активное сканирование вычислительной сети с использованием таких инструментов как *Nmap*, *Nessus*, *Wireshark*, *ZENMap*, *Sparta*, *OpenVAS* (соответствующие тактики и техники реализации описаны в матрице *MITRE ATT&CK*: *TA*0043, *T*1595, *T*1590[2], и в методике оценки угроз информационной безопасности ФСТЭК России: Т1.3, Т1.4). В связи с тем, что различные типы активного сканирования сетевых узлов вычислительной сети содержатся в сигнатурах

1 Горбачев Александр Александрович, кандидат технических наук, Краснодарское высшее военное орденов Жукова и Октябрьской Революции Краснознаменное училище имени генерала армии С. М. Штеменко, г. Краснодар, Россия. E-mail: infosec23.00@mail.ru

2 MITRE ATT&CK. Enterprice: Reconnaissance. https://attack.mitre.org/tactics/TA0043/ (дата обращения 01.07.24 г.)

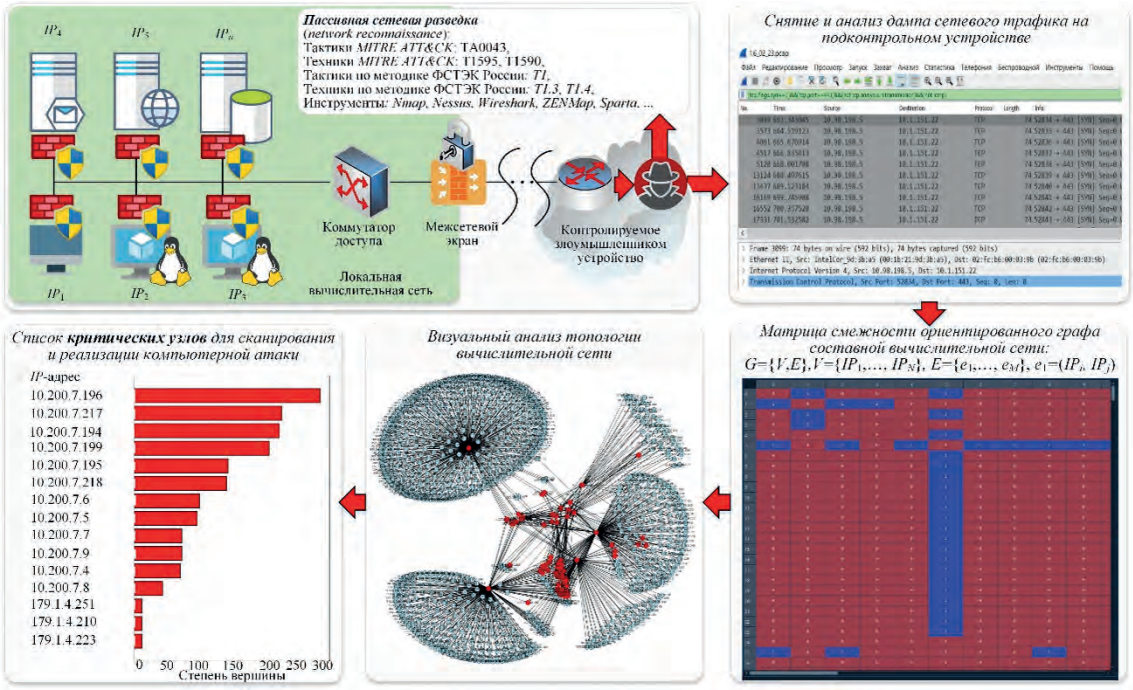


Рис. 1. Процесс пассивной сетевой разведки злоумышленника, направленной на вскрытие топологических свойств вычислительной сети

и правилах *SIEM*-систем (*Security Information and Event Management*), средств обнаружения вторжений и межсетевых экранов, то наиболее безопасным и скрытным способом сетевой разведки для злоумышленников является пассивный анализ сетевого трафика [3].

После реконструкции топологии вычислительной сети злоумышленником может быть составлен перечень наиболее важных для дальнейшего исследования и информационно-технического воздействия узлов вычислительной сети с точки зрения их топологических свойств (рисунок 1).

Работа посвящена методам генерации ложного сетевого трафика с заданной ложной топологией защищаемой вычислительной сети с целью снижения эффективности действий злоумышленников, реализующих сетевую разведку посредством пассивного анализа сетевого трафика. В зарубежной литературе аналогичный метод введения злоумышленника в заблуждение называется *киберобманом* посредством *обфускации топологии* вычислительной сети, при этом синтез ложной топологии осуществляется посредством: случайного изменения параметров маршрутизации заголовков пакетов в соответствии с заданными критериями и ограничениями[3] [4], добавления логических связей между узлами вычислительной сети с учетом распределения степеней вершин [5], использования виртуальных маршрутизаторов[4] и сетевых ловушек [6, 7]. В отечественной литературе маскирование истинной топологии вычислительной сети рассматривалось посредством трансляции сетевых адресов, расширения адресного пространства, введения ложных объектов[5] [8, 9], обеспечения работоспособности сложных систем в условиях деструктивных информационных воздействий [10, 11].

Тем не менее, анализ качества различных моделей, алгоритмов и постановок задач синтеза ложной топологии вычислительной сети с учетом размерности в полной мере не был реализован. В приведенном материале синтез ложной топологии вычислительной сети осуществляется посредством решения задачи комбинаторной оптимизации матрицы смежности ориентированного графа с использованием генетического алгоритма для вычислительных сетей низкой размерности, а также с использованием классических моделей случайных графов, реализующих синтез ложной структуры вычислительной сети с заданными свойствами близости к топологии реальной сети и защищенности от деструктивных воздействий на критические узлы, качество ложной структуры также оценивается как степень пересечения множества ложных и истинных критических узлов вычислительной сети.

3 Roland Meier, Petar Tsankov, Vincent Lenders, Laurent Vanbever, Martin Vechev. NetHide: Secure and Practical Network Topology Obfuscation. Proceedings of the 27-th USENIX Security Symposium. 2018. pp. 693–709.

4 Stefan Achleitner, Thomas F. La Porta, Patrick McDaniel, Shridatt Sugrim, Srikanth V. Krishnamurthy, Ritu Chadha. Deceiving Network Reconnaissance Using SDN-Based Virtual Topologies // IEEE Transactions On Network And Service Management, Vol. 14, No. 4, 2017. pp. 1098–1112.

5 Шерстобитов Р. С., Шарифуллин Р. С., Максимов Р. В. Маскирование интегрированных сетей связи ведомственного назначения // Системы управления, связи и безопасности. 2018. № 4. С. 136–175.

Анализ объекта исследования

Вычислительная сеть (компьютерная сеть, сеть передачи данных) представляет собой совокупность узлов (компьютеров) и аппаратуры передачи данных, ветви которых являются линиями передачи данных. Под топологией вычислительной сети в контексте данной работы будет пониматься ориентированный и невзвешенный граф (орграф).

Ориентированный граф, характеризующий топологию вычислительной сети с N узлами может быть однозначно определен компонентами (выражение 1):

$$G = \{V,E\}, \quad (1)$$

где $V = \{v_1, \ldots, v_N\}$ – вершины графа, применительно к рассматриваемой области $V = \{IP_1, \ldots, IP_N\}$ вершинами графа являются *IP*-адреса сетевых узлов; $E = \{e_1, \ldots, e_N\}$ – ребра графа, причем $e_1 = (v_i, v_j)$ – направленное ребро от узла v_i до ребра v_j или для вычислительной сети $e_1 = (IP_i, IP_j)$.

Для математического моделирования ориентированных графов как правило используют матричное представление графа в виде *матрицы смежности* или *матрицы инциденций*. Матрица смежности представляет собой квадратную матрицу размерностью N с бинарными значениями $A \in \{0,1\}^{N\times N}$, причем для орграфов матрица смежности в общем случае не является симметричной.

Под топологическими свойствами в работе понимаются характеристики ориентированного графа, которые с точки зрения формы их представления можно разделить на *скалярные* и *векторные* (*распределения*). Скалярные характеристики представляют собой функционал от компонентов орграфа G.

С целью создания ***правдоподобной ложной топологии*** вычислительной сети могут быть использованы различные характеристики *близости* топологических свойств орграфов, характеризующих реальную и ложную вычислительные сети. С одной стороны, указанная близость может быть оценена как отклонение между функционалами (*норма Фробениуса*) или как *мера схожести* (*коэффициент Жаккара*, *d-мера*, *δ-мера*) от матриц смежности реальной сети A_{real} и ложной сети A_{synt}. С другой стороны, характеристиками близости могут выступать любые скалярные или векторные характеристики орграфа, к примеру, количество ребер $M = |E|$, вершин $N = |V|$, спектральный радиус, коэффициент кластеризации, распределение входящих и исходящих степеней вершин. Для оценки качества аппроксимации векторных характеристик графов могут быть использованы различные метрики дистанций распределений (дивергенция Кульбака-Лейблера, статистические критерии проверки гипотез об однородности распределений). Вопрос использования тех или иных показателей близости структур является дискуссионным, но с точки зрения вычислительной сложности и физической интерпретируемости в качестве показателей качества аппроксимации топологических свойств реальной вычислительной сети посредством ложной в работе рассматриваетс я коэффициент Жаккара $J(A_{real}, A_{synt})$ в соответствии с выражением (2). Для удобства решения задачи оптимизации за счет минимизации целевой функции, характеризующей степень близости ложной и реальной вычислительной сети, форма коэффициента Жаккара имеет вид:

$$K_{sym1} = J(A_{real}, A_{synt}) = 1 - \frac{|E_{real} \cap E_{synt}|}{|E_{real} \cup E_{synt}|}, \quad (2)$$

где $E_{real} = \{e^{real}_1,\ldots,e^{real}_k\}$ – множество ребер орграфа реальной вычислительной сети, $E_{synt} = \{e^{synt}_1,\ldots,e^{synt}_n\}$ – множество ребер орграфа ложной вычислительной сети.

В контексте оценки ***защищенности*** вычислительных сетей наиболее важными топологическими характеристиками ориентированных графов являются: *характеристики устойчивости структуры в целом к деструктивным воздействиям и характеристики важности отдельных узлов*.

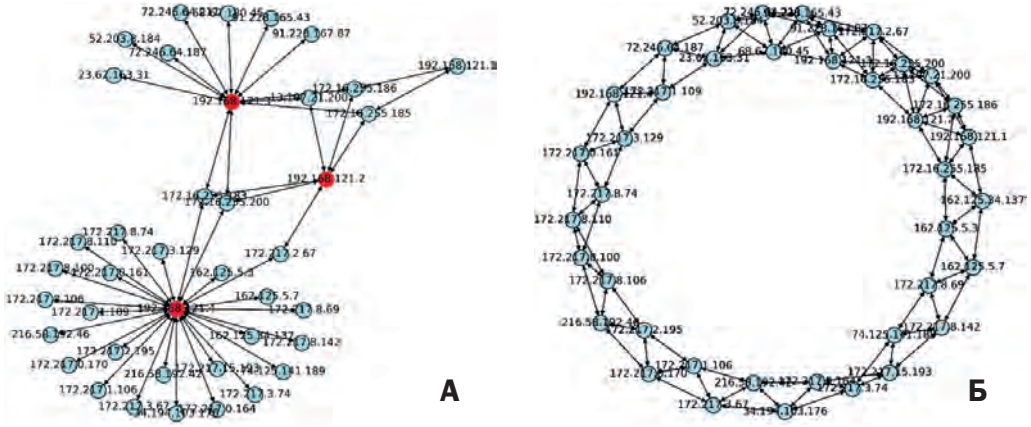


Рис. 2. Граф Харари (б) с количеством вершин N = 40 и количеством ребер M = 92, соответствующий реальной вычислительной сети (а). Средний коэффициент кластеризации исходного графа (а) – 0,0; графа Харари (б) – 0,5375.

Коэффициент кластеризации позволяет оценить общую устойчивость топологии вычислительной сети к деструктивным воздействиям. Например, при высоком коэффициенте кластеризации орграфа удаление случайного ребра или вершины приведет к незначительным негативным последствиям. Граф *Харари* (рисунок 2, б) характеризуется высоким коэффициентом кластеризации и соответственно высокой устойчивостью к деструктивным воздействиям, направленным на нарушение связности сети при фиксированном количестве вершин и ребер графа, поэтому данную модель топологии также целесообразно использовать при построении ложной топологии вычислительной сети, которая с точки зрения злоумышленника будет иметь меньше уязвимых узлов.

Несмотря на то, что адекватную модель злоумышленника в общем случае построить не представляется возможным, существует возможность выделить несколько предпочтений, которые могут быть сделаны на основе анализа нарушителями топологических свойств вычислительной сети. В связи с ограниченностью ресурсов планирование и реализация атаки злоумышленниками ведется на подмножество наиболее важных или в некотором смысле ***критических узлов*** в вычислительной сети. Атака на критические узлы позволяет нарушить работоспособность сетевых сервисов для целых кластеров (подмножеств) вычислительной сети. Критичность узла может быть определена по нескольким топологическим признакам:

- критические узлы с наибольшим значением *степени вершины* или значением степени, значительно превосходящим степени других узлов (вершин). Интуитивные соображения подсказывают, что важные узлы вычислительной сети реализуют информационный обмен с большим числом узлов в сети;
- критические узлы с наибольшим значением *коэффициента связности*, значительно превосходящим другие узлы;
- критические узлы как *артикуляционные узлы* графа. Вершина графа $v_k = \{IP_k\}$ называется артикуляционной тогда и только тогда, когда некоторое наименьшее вершинное покрытие графа содержит эту вершину[6]. Наименьшее вершинное покрытие графа — это минимальный набор вершин, таких, что каждое ребро графа инцидентно хотя бы одной вершине из этого набора. Это означает, что каждая вершина в наименьшем вершинном покрытии является важной для обеспечения связности между элементами графа. Удаление артикуляционных узлов приведет к нарушению связи у целых подсетей или подмножеств подсетей. Одним из распространенных и эффективных способов нахождения артикуляционных узлов является алгоритм *Тарьяна* с линейной временной сложностью $O(|V|+|E|)$[7].

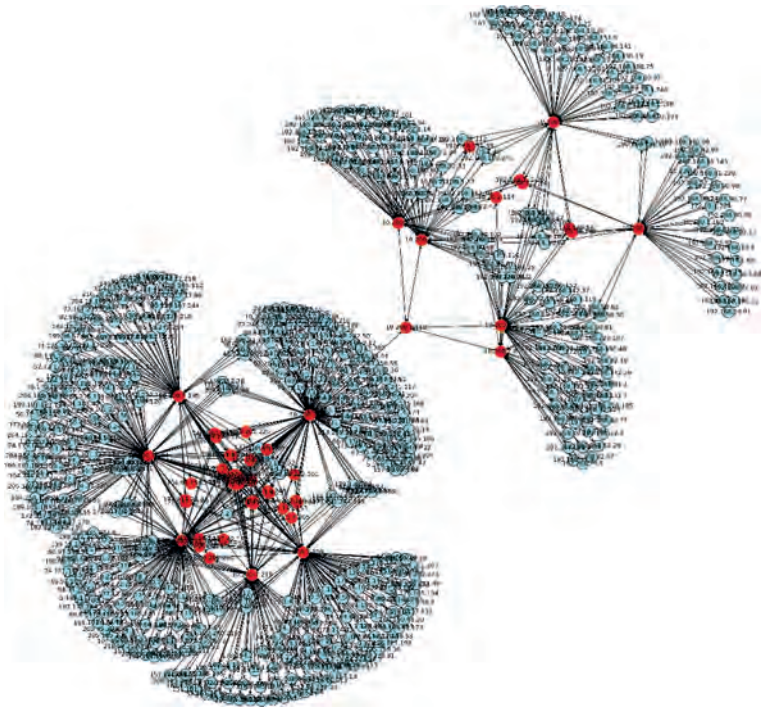


Рис. 3. Реконструкция топологии составной вычислительной сети из общедоступного дампа трафика[8]: N = 643 шт.; E = 1710 шт.; красным цветом обозначены критические узлы со степенями выше, чем 99-й процентиль степеней графа

Рассмотрим визуализацию топологии вычислительной сети, полученную посредством анализа дампа трафика из общедоступного репозитория (рисунок 3). Анализ сетевого трафика за длительный промежуток времени позволяет восстановить топологию составной вычислительной сети с учетом идентификации критических узлов (на рисунке 3 критические узлы определены, исходя из степеней вершин выше, чем 99-й процентиль всех степеней графа, то есть критическими являются узлы, степени которых больше, чем у 99 % остальных узлов сети). Как видно из рисунка, простейшее отсечение узлов с высокими степенями вершин позволяет идентифицировать наиболее важные с точки зрения связности и обеспечения доступности информационных ресурсов узлы вычислительной сети.

6 Харари Ф. Теория графов / Ф. Харари. М: 1973, 300 с.
7 Farima G. A linear time algorithm to compute the impact of all the articulation points // arXiv:1504.00341v3 [cs.DS]. – 2015 (дата обращения 01.07.24 г.)
8 Kaggle: IP Network Traffic Flows Labeled with 75 Apps. https://www.kaggle.com/datasets/jsrojas/ip-network-traffic-flows-labeled-with-87-apps (дата обращения 01.07.24 г.)

Решение задачи синтеза топологии ложной вычислительной сети

Для снижения возможностей злоумышленника по реализации информационно-технических воздействий на критические узлы исходя из топологических свойств вычислительной сети целесообразной является гипотеза о том, что синтезированная структура должна обладать критическими узлами, отличными от реальных критических узлов (по относительному расположению), более того необходимо расположить их на максимальной дистанции от реальных критических узлов, то есть максимизировать среднее кратчайшее расстояние от ложных до реальных критических узлов.

То есть вербальная постановка задачи на синтез ориентированного графа ложной вычислительной сети заключается в подборе такой модели и ее параметров, которая будет генерировать орграфы с максимальной степенью близости к исходному графу и при этом максимальным средним кратчайшим расстоянием от ложных до реальных узлов.

Решение указанной задачи синтеза ложной структуры вычислительной сети в общем виде можно определить в форме задачи многокритериальной оптимизации (выражение 3):

$$\begin{cases} K_{sim}(S, \theta, A_{real}) \rightarrow \underset{K_{sim},K_{def},S,\theta,A_{real} \in Q_1}{extr} \\ K_{def}(S, \theta, A_{real}) \rightarrow \underset{K_{sim},K_{def},S,\theta,A_{real} \in Q_1}{extr} \end{cases} \quad (3)$$

где S – форма модельного оператора, определяющая класс моделей, аппроксимирующих соответствующие свойства орграфа вычислительной сети; $\theta = (\theta_1, \ldots, \theta_m)$ – в общем случае вектор параметров соответствующего модельного оператора; Q_1 – допустимое множество значений целевых функций и аргументов; $K_{sim}(S, \theta, A_{real})$ – функция качества аппроксимации близости топологии реальной вычислительной сети; $K_{def}(S, \theta, A_{real})$ – функция, характеризующая количественную оценку защищенности реальной вычислительной сети при генерации заданной ложной топологии вычислительной сети.

Решение подобных задач сводится к нахождению Парето оптимального множества решений либо к выбору одного из методов *скаляризации*, к примеру, метода главного критерия (целевой функции), идеальной точки, линейной свертки целевой функции или использования других отношений предпочтения в критериальном пространстве. Причем, если $\theta = A$, то есть если в качестве параметров рассматривать элементы матрицы смежности, то синтез структуры осуществляется как решение задачи *комбинаторной оптимизации* матрицы смежности с количеством параметров N^2, а алгоритм нахождения экстремума целевой функции имеет нижнюю оценку вычислительной сложности $O(N^2)$. Размерность пространства решений задачи полного перебора значений элементов матрицы смежности составляет 2^{N^2}. Решение подобной задачи целесообразно лишь для графов малой размерности (при $N < 50$).

Рассмотрим синтез структуры вычислительной сети как задачу комбинаторной оптимизации скалярной целевой функции $f_1(\alpha_1, \alpha_2, S, \theta, A_{real})$, представляющей собой линейную свертку (взвешенную сумму) функций $K_{sim}(S, \theta, A_{real})$ и $K_{def}(S, \theta, A_{real})$. Постановка задачи с непосредственным нахождением элементов матрицы смежности удовлетворяет задаче (выражение 4):

$$f_1(\alpha_1, \alpha_2, S, \theta, A_{real}) = \alpha_1 \cdot K_{sim}(S, \theta, A_{real}) + \\ + \alpha_2 \cdot K_{def}(S, \theta, A_{real}) \rightarrow \underset{K_{sim},K_{def},S,\theta,A_{real},\alpha_1,\alpha_2 \in Q_1}{extr}, \quad (4)$$

где α_1, α_2 – коэффициенты значимости функций $K_{sim}(S, \theta, A_{real})$ и $K_{def}(S, \theta, A_{real})$ соответственно.

В работе в качестве характеристики близости структур $K_{sim}(S, \theta, A_{real})$ выступает коэффициент Жаккара $J(A_{real}, A_{synt})$ между генерируемой матрицей смежности A и матрицей смежности реального графа A_{real}. Характеристикой защищенности $K_{def}(S,\theta,A_{real})$ структуры является функция $D(A_{real}, A)$ как среднее кратчайшее расстояние между критическими узлами генерируемой матрицы смежности A и критическими узлами исходной матрицы A_{real} (выражение 5, 6):

$$f_1(\alpha_1, \alpha_2, S, \theta, A_{real}) = \alpha_1 \cdot J(A_{real}, A) + \\ + \alpha_2 \cdot (D(A_{real}, A) + \varepsilon)^{-1} \rightarrow \underset{A_{real},A,\alpha_1,\alpha_2 \in Q_3}{min} \quad (5)$$

$$Q_3 = \begin{cases} \alpha_1 \in [0,1], \alpha_2 \in [0,1], \\ J(A_{real}, A) \in [0,1], \\ D(A_{real}, A) \geq 0, \\ A \in \{0,1\}^{N \times N}, A_{real} \in \{0,1\}^{N \times N}, \\ N \leq 30, \varepsilon = 1{,}0. \end{cases} \quad (6)$$

где ε – коэффициент, предотвращающий деление на 0; $D(A_{real}, A)$ – функция, вычисляющая среднее кратчайшее расстояние между подмножеством критических узлов графа, восстановленного из матрицы смежности A_{real} реальной вычислительной сети и подмножеством критических узлов графа, восстановленного из матрицы смежности A ложной вычислительной сети.

Алгоритм вычисления функции $D(A_{real}, A)$ включает в себя:

- вычисление списка критических узлов реальной сети по матрице смежности A_{real}, для чего в зависимости от критерия важности узлов осуществляют сортировку списка критических узлов по степени важности (степени вершины или степени связности вершины);
- вычисление списка критических узлов ложной сети по матрице смежности A;

- вычисление кратчайшего расстояния между каждым критическим узлом реальной и ложной вычислительной сети с использованием алгоритма Дейкстры;
- вычисление среднего значения кратчайшего расстояния между критическими узлами реальной и ложной вычислительной сети.

Алгоритмическая реализация функции $D(A_{real}, A)$ может быть дополнена штрафными слагаемыми за пересечение списков критических узлов ложной и реальной вычислительной сети.

Для решения задач комбинаторной оптимизации с большим количеством переменных часто используют генетические алгоритмы оптимизации, демонстрирующие приближенное решение задач об N ферзях, о рюкзаке, коммивояжера и маршрутизации транспорта с допустимой точностью и за приемлемое время [12].

Использование простого генетического алгоритма для решения задач численной оптимизации включает в себя этапы:

- ***создание начальной популяции:*** начальная популяция представляет собой количество разглаженных векторов, содержащих значения 0 или 1, длиной N^2. В работе размер начальной популяции составил от 100 до 1000 индивидуумов (векторов) в зависимости от размерности матрицы смежности A_{real};
- ***вычисление приспособленности*** каждого индивидуума: в качестве функции приспособленности выступает скалярная целевая функция $f_1(\alpha_1, \alpha_2, A_{real}, A)$, весовые коэффициенты функций $\alpha_1 = \alpha_2 = 1$.
- ***отбор:*** включает в себя процедуру турнирного отбора индивидуумов-родителей для индивидуумов следующей итерации алгоритма. Размер турнира является гиперпараметром и составляет величину от 3 до 10;
- ***скрещивание:*** в работе реализовано одноточечным скрещиванием, при этом вероятность скрещивания каждой особи в поколении составляет величину 0,5;
- ***мутация:*** способ мутации – инвертирование бинарного значения элемента вектора (индивида), вероятность мутации составляет величину 0,2;
- ***проверка критерия остановки алгоритма:*** достижение заданного количества итераций расчета (от 100 до 1000 в зависимости от размерности N).
- ***выбор*** индивидуумов с максимальной приспособленностью (минимальным значением целевой функции f_1).

Для реализации вычислительного эксперимента по синтезу ложной структуры вычислительной сети с использованием генетического алгоритма и решения задачи комбинаторной оптимизации был использован дамп трафика из открытого источника[9], из которого была восстановлена топология вычислительной сети с заданным количеством узлов N.

Результаты вычислений, представленные на рисунке 4 показывают, что при использовании генетического алгоритма полученные структуры характеризуются более равномерной связностью (степенями вершин), а критические узлы, вычисленные по признаку степени вершины (значение степени выше 99-го процентиля степеней в графе) либо отсутствуют (рисунок 4, а), либо переместились на другие вершины (рисунок 4, б). Стоит отметить высокую вычислительную сложность процесса поиска оптимальной в указанном смысле структуры, которая в свою очередь ограничивает применимость генетических алгоритмов и решения задачи комбинаторной оптимизации элементов матрицы смежности A_{real} лишь для локальных вычислительных сетей с небольшим количеством узлов $N < 50$.

Оценка качества маскирования структуры вычислительной сети

Как предполагалось ранее, злоумышленник стремится к минимизации вероятности компрометации факта сетевой разведки узлов вычислительной сети, что приводит к тому, что в качестве мишеней для дальнейшего исследования (сканирования) и реализации компьютерных атак выбирается ограниченное подмножество узлов вычислительной сети, исходя из множества узлов, вскрытого посредством пассивного анализа сетевого трафика.

В качестве критерия отбора узлов по степени их важности рассмотрим степень вершины. В указанных условиях в качестве количественного показателя защищенности вычислительной сети может выступать *степень пересечения* множества критических узлов, отобранного злоумышленником на основе анализа ложной вычислительной сети с множеством критических узлов реальной вычислительной сети (выражение 7):

$$\Omega(S_{real}, S_{id}, N, \Delta) = \frac{|S_{real}(N, \Delta) \cap S_{id}(N, \Delta)|}{|S_{real}(N, \Delta)|} \quad (7)$$

где $S_{real} = \{IP^{real}{}_1,\ldots,IP^{real}{}_k\}$ – множество критических узлов реальной вычислительной сети, $S_{id} = \{IP^{id}{}_1,\ldots, IP^{id}{}_n\}$ – множество критических узлов, отобранных злоумышленником в качестве критических на основе анализа ложной топологии вычислительной сети, Δ – доля вершин графа, определяющая длину ранжированного по степени списка критических узлов графа, то есть, $\Delta = 0{,}1$, означает, что злоумышленником будет

9 Kaggle: Labeled Network Traffic flows – 141 Applications. https://www.kaggle.com/datasets/jsrojas/labeled-network-traffic-flows-114-applications/data (дата обращения 01.07.24 г.).

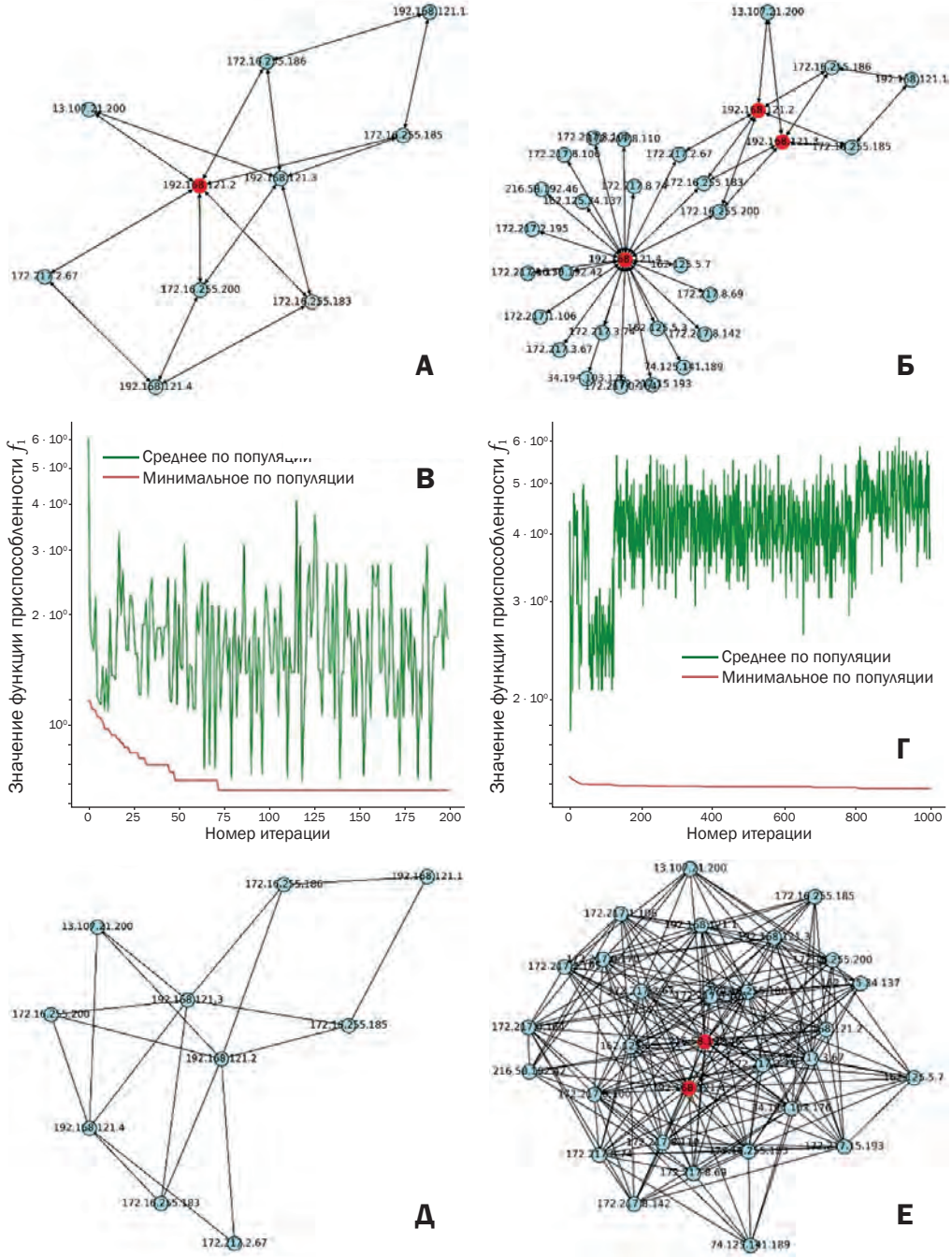


Рис. 4. Синтез топологии ложной вычислительной сети посредством решения задачи комбинаторной оптимизации генетическим алгоритмом в зависимости от количества вершин N: исходный граф для N = 10 шт. (а), для N = 30 шт. (б); процесс поиска минимума функции приспособленности для N = 10 шт. (в), для N = 30 шт. (г); результат синтеза ложной структуры для N = 10 шт. (д), время расчета – 6,481 с, для N = 30 шт. (е), время расчета – 839,432 с

составлен список критических узлов, ранжированный по степени и длина которого равна 10% от общего количества узлов подсети.

Для сравнительной характеристики эффективности максирования топологических свойств вычислительной сети были использованы классические модели случайных графов Эрдеша-Реньи, Барбаши и Харари. Параметры соответствующих моделей были оптимизированы с точки зрения показателя близости J структур, то есть, для модели Эрдеша-Реньи параметрическая идентификация имеет вид (выражение 8):

$$J_{ER}\left(p_{ER}, N_{ER}, A, A_{real}\right) \rightarrow \min_{N_{ER} = |N_{real}|,\ p_{ER} \in [0,1]}, \qquad (8)$$

где, p_{ER} – вероятность наличия ребра между вершинами графа Эрдеша-Реньи, N_{ER} – количество вершин графа Эрдеша-Реньи.

Для модели Барбаши параметрическая оптимизация имеет вид (выражение 9):

$$J_B(M_B, N_B, A, A_{real}) \rightarrow \min_{N_B = |N_{real}|,\ M_B \in Int,\ M_B \in [1, |N_{real}|-1]}, \quad (9)$$

где M_B – среднее количество ребер вершины графа Барбаши, N_B – количество вершин графа Барбаши.

Нахождение экстремумов функций J_{ER} и J_B является тривиальной задачи в связи с поиском оптимальных значений одного параметра в каждом случае, так как параметры $N_{ER} = N_B = N_{real}$ имеют фиксированное значение. Для модели Харари оптимальными условиями моделирования реальной вычислительной сети является равенство количества ребер $N_H = |N_{real}|$ и количества вершин $E_H = |E_{real}|$ графа Харари и реального графа соответственно.

Далее, используя указанные модели с оптимальными параметрами с точки зрения критерия близости, генерируется по 50 ложных вычислительных сетей с фиксированным количеством узлов вычислительной сети N и фиксированной долей Δ вершин вычислительной сети, относимой злоумышленником к критическим узлам (рисунок 5).

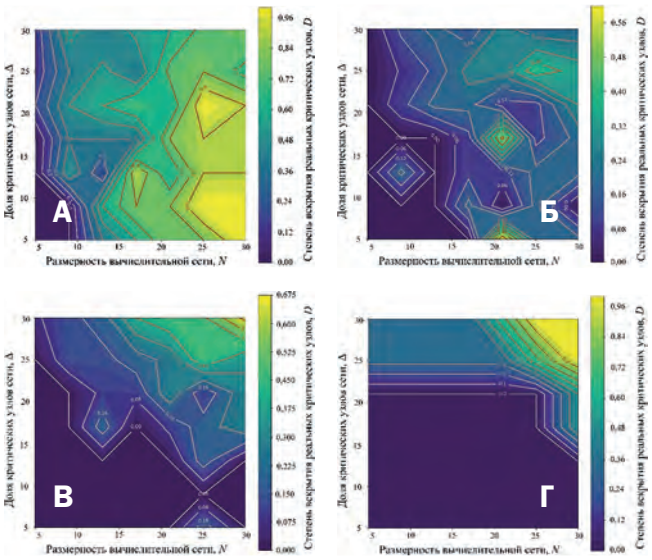


Рис. 5. Степень вскрытия злоумышленником реальных критических узлов вычислительной сети в зависимости от размерности сети N и доли Δ критических узлов в вычислительной сети: а) для генетического алгоритма; б) для модели Эрдеша-Реньи; в) для модели Барбаши; г) для модели Харари

Как видно из расчетов, при использовании генетического алгоритма (рисунок 5 а) степень вскрытия реальных критических узлов вычислительной сети D быстро растет с увеличением размерности сети N и доли Δ критических узлов в вычислительной сети, что, в свою очередь, может быть связано с тем, что поиск оптимальной структуры осуществляется исходя из двух показателей качества с одной стороны и приближенностью полученного решения с другой стороны. Использование моделей Эрдеша-Реньи и Барбаши (рисунок 5 б, в) показывают схожие результаты и лучшие по отношению к генетическому алгоритму с вышеуказанными диапазонами гиперпараметров. Также синтез ложной структуры по модели Харари позволяет синтезировать защищенную топологию и с точки зрения вскрытия критических узлов, и с точки зрения среднего коэффициента кластеризации. Также на результаты численного эксперимента оказывает влияние конкретный вид реальной вычислительной сети, по отношению к которой синтезируется ложная структура.

Полученные результаты позволяют сделать выводы:

- изолированное использование численных алгоритмов оптимизации, в частности, генетического алгоритма, при решении задачи комбинаторной оптимизации матрицы смежности нецелесообразно в связи с высокой вычислительной сложностью процесса поиска оптимальной структуры, обусловленной экспоненциальным ростом пространства возможных комбинаций;
- синтез структур с использованием классических моделей случайных графов, в частности моделей Эрдеша-Реньи, Барбаши, Харари, характеризуются относительно низкой вычислительной сложностью и при этом обеспечивают синтез структур вычислительных сетей, относительно более защищенных в рассмотренной метрике пересечения списка ложных и истинных критических узлов вычислительной сети;
- для решения задачи синтеза ложной топологии с большим количеством вершин ($N > 50$), что свойственно для структур, формируемых при анализе дампов трафика в информационно-телекоммуникационной сети общего пользования, целесообразно использовать либо относительно простые модели синтеза графов, рассмотренные в работе, либо методы снижения размерности поставленной задачи;
- дальнейшее направление исследований будет направлено на исследование эффективности совместного использования алгоритмов снижения размерности, машинного обучения, генетических алгоритмов и классических моделей случайных графов для синтеза ложных топологий вычислительных сетей большой размерности с заданными свойствами близости и защищенности.

Литература

1. Зегжда Д. П., Александрова Е. Б., Калинин М. О., Марков А. С. и др. Кибербезопасность цифровой индустрии. Теория и практика функциональной устойчивости к кибератакам / Под ред. профессора РАН, доктора техн. наук Д. П. Зегжды. – М.: Горячая линия – Телеком, 2019. – 560 с.
2. Stefan Marksteiner, Bernhard Jandl-Scherf and Harald Lernbeiß. Automatically Determining a Network Reconnaissance Scope Using Passive Scanning Techniques. Fourth International Congress on Information and Communication Technology. London. 2020. vol. 2. p. 117–127.
3. Дорофеев А. В., Марков А. С. Мониторинг событий информационной безопасности: технологии и методы контроля эффективности // Вестник военного инновационного технополиса «ЭРА». 2022. Т.3. № 4. С. 392–400.
4. Tao Hou, Tao Wang, Zhou Lu, Yao Liu. Combating Advarsarial Network Topology Inference by Proactive Topology Obfuscation. IEEE INFOCOM 2020. 2020. pp. 1–14.
5. Jinwoo Kim, Eduard Marin, Mauro Conti, Seungwon Shin. EqualNet: A Secure and Practical Defense for Long-term Network Topology Obfuscation. Network and Distributed Systems Security (NDSS) Symposium. 2022. pp. 1–18.
6. Rawski M. Network Topology Mutation as Moving Target Defense for Corporate Networks // INTL Journal Of Electronics And Telecommunications. 2019. Vol. 65, No. 4, pp. 571–577.
7. Hou T. et al. Proto: Proactive topology obfuscation against adversarial network topology inference // IEEE INFOCOM 2020-IEEE Conference on Computer Communications. – IEEE, 2020. Pp. 1598–1607.
8. Кучуров В. В., Максимов Р. В., Шерстобитов Р. С. Модель и методика маскирования адресации корреспондентов в киберпространстве // Вопросы кибербезопасности. 2020. № 6(40). С. 2–13. DOI:10.21681/2311-3456-2020-6-2-13
9. Теленьга А. П. Маскирование метаструктур информационных систем в киберпространстве // Вопросы кибербезопасности. 2024. № 5(57). С. 50–59. DOI:10.21681/2311-3456-2024-5-50-59
10. Зегжда Д. П. Интеллектуальные методы саморегуляции распределенных сетевых структур в условиях кибератак // XIV Всероссийская мультиконференция по проблемам управления МКПУ-2021. 2021. С. 16–19.
11. Лаврова Д. С., Зегжда Д. П., Зайцева Е. А. Моделирование сетевой инфраструктуры сложных объектов для решения задачи противодействия кибератакам // Вопросы кибербезопасности. 2019. №2(30). С. 13–20. DOI:10.21681/2311-3456-2019-2-13-20
12. Вирсански Э. Генетические алгоритмы на Python / пер. с англ. А. А. Слинкина. – М.: ДМК Пресс, 2020. – 286 с.: ил. ISBN 978-5-97060-857-9.

MASKING OF TOPOLOGICAL PROPERTIES OF COMPUTER NETWORKS IN NETWORK RECONNAISSANCE CONDITIONS. Part 1

Gorbachev A. A.[10]

The purpose of the study: *to study models of random graphs and genetic algorithms for solving the problem of synthesizing a false structure to mask the topological properties of computer networks when generating false network traffic and using false network information objects, taking into account the degree of similarity of the topological properties of real computer networks with false ones, as well as taking into account the security index of computer networks.*

Methods used: *genetic optimization algorithm, linear convolution method, Erdos-Renyi, Barbashi, Harari model.*

The result of the study: *the synthesis of a false structure of a computer network based on random graph models and evolutionary optimization algorithms makes it possible to increase the effectiveness of protecting a computer network by reducing the ability of an attacker to identify its critical nodes through network traffic analysis. The Jacquard coefficient between the sets of edges of true and false computer networks acts as an indicator of the proximity of the topological characteristics of computer networks, and the average shortest distance acts as an approximation of the distance between true and false critical nodes. Genetic algorithms make it possible to solve the problem of optimal parameterization of random graph models from the point of view of the selected fitness function, as well as with explicit combinatorial optimization of a false topology. The exponential growth of the bulkhead space does not allow solving the problem of combinatorial optimization of the adjacency matrix of a graph characterizing the topology of a large computer network, which leads to the need to use dimensionality reduction methods and parametric models when masking the topological properties of composite computer networks.*

Scientific novelty: *it consists in solving the problem of synthesizing the topological properties of a false computer network using genetic algorithms and random graph models parameterized taking into account the scalar fitness objective function, which includes an indicator of the proximity of the false and true topological structure of the computer network, as well as approximating the distance between true and false critical nodes of the computer network.*

Keywords: *network traffic analysis, proactive protection, honeypots, evolutionary optimization algorithms, critical nodes.*

10 Alexander A. Gorbachev , Ph.D. Krasnodar Higher Military Orders of Zhukov and the October Revolution Red Banner School named after General of the Army S. M. Shtemenko, Krasnodar, Russia. E-mail: infosec23.00@mail.ru

References

1. *Zegzhda D. P., Aleksandrova E. B., Kalinin M. O., Markov A. S. i dr. Kiberbezopasnost' cifrovoj industrii. Teoriya i praktika funkcional'noj ustojchivosti k kiberatakam / Pod red. professora RAN, doktora texn. nauk D. P. Zegzhdy'. – M.: Goryachaya liniya – Telekom, 2019. – 560 p.*
2. *Stefan Marksteiner, Bernhard Jandl-Scherf and Harald Lernbeiß. Automatically Determining a Network Reconnaissance Scope Using Passive Scanning Techniques. Fourth International Congress on Information and Communication Technology. London. 2020. vol. 2. p. 117–127.*
3. *Dorofeev A. V., Markov A. S. Monitoring sobytij informacionnoj bezopasnosti: tekhnologii i metody kontrolya effektivnosti // Vestnik voennogo innovacionnogo tekhnopolisa «ERA». 2022. T.3. № 4. pp. 392–400.*
4. *Tao Hou, Tao Wang, Zhou Lu, Yao Liu. Combating Advarsarial Network Topology Inference by Proactive Topology Obfuscation. IEEE INFOCOM 2020. 2020. pp. 1–14.*
5. *Jinwoo Kim, Eduard Marin, Mauro Conti, Seungwon Shin. EqualNet: A Secure and Practical Defense for Long-term Network Topology Obfuscation. Network and Distributed Systems Security (NDSS) Symposium. 2022. pp. 1–18.*
6. *Rawski M. Network Topology Mutation as Moving Target Defense for Corporate Networks // INTL Journal Of Electronics And Telecommunications. 2019. Vol. 65, No. 4, pp. 571–577.*
7. *Hou T. et al. Proto: Proactive topology obfuscation against adversarial network topology inference // IEEE INFOCOM 2020-IEEE Conference on Computer Communications. – IEEE, 2020. pp. 1598–1607.*
8. *Kuchurov V. V., Maksimov R. V., Sherstobitov R. S. Model' i metodika maskirovaniya adresacii korrespondentov v kiberprostranstve // Voprosy kiberbezopasnosti. 2020. № 6(40). pp. 2–13.*
9. *Telen'ga A. P. Maskirovanie metastruktur informacionnyh sistem v kiberprostranstve // Voprosy kiberbezopasnosti. 2024. № 5(57). pp. 50–59.*
10. *Zegzhda D. P. Intellektual'ny'e metody' samoregulyacii raspredelenny'x setevy'x struktur v usloviyax kiberatak // XIV Vserossijskaya mul'tikonferenciya po problemam upravleniya MKPU-2021. 2021. pp. 16–19.*
11. *Lavrova D. S., Zegzhda D. P., Zajceva E. A. Modelirovanie setevoj infrastruktury' slozhny'x ob'ektov dlya resheniya zadachi protivodejstviya kiberatakam // Voprosy` kiberbezopasnosti. 2019. №2(30). pp. 13–20.*
12. *Virsanski E. Geneticheskie algoritmy na Python / per. s angl. A. A. Slinkina. – M.: DMK Press, 2020. – 286 p.: ISBN 978-5-97060-857-9.*

МОДЕЛЬ СИСТЕМЫ АДАПТИВНОГО УПРАВЛЕНИЯ КИБЕРПОЛИГОНОМ МЧС РОССИИ НА ОСНОВЕ ОПЕРАТОРНОГО УРАВНЕНИЯ

Грызунов В. В.[1], Шестаков А. В.[2]

DOI: 10.21681/2311-3456-2024-6-140-149

Цель исследования: *формулировка условия существования киберполигона как организационно-технической системы, гарантированно решающей поставленные задачи.*

Методы исследования: *предложенная модель базируется на модели FIST, методах теории адаптивного управления.*

Полученные результаты: *1) показано, что киберполигон МЧС России имеет несколько треков согласно направлениям решаемых задач, является территориально распределённым, интегрируется с информационной инфраструктурой МЧС, оперирует пространственными данными, функционирует в среде всех возможных типов и проявляется в виде набора производительностей обеспечивающего уровня, уровня персонала, уровня аппаратного обеспечения и уровня программного обеспечения; 2) сформулированы ограничения, при которых возможен синтез киберполигона как системы адаптивного управления с изменяющейся архитектурой: на стабильность множества управляющих воздействий, на среднее время стабильного существования киберполигона; 3) формализовано операторное уравнение, описывающее киберполигон как организационно-техническую систему, обслуживаемую персоналом, и характеризующее условие гарантированного решения задач, стоящих перед киберполигоном; 4) обосновано введение новых элементов в структуру киберполигона: блока наблюдения и блока управления с учётом обратной связи.*

Научная новизна: *получена модель киберполигона, отличающаяся формализацией условия существования киберполигона как организационно-технической системы на всех уровнях (обеспечивающем, персонала, аппаратном, программном).*

Обсуждение: *конкретный вид формализованного в статье операторного уравнения может быть найден с помощью метода iSOFT.*

Ключевые слова: *модели управления информационной безопасностью, синтез организационно-технических систем, подготовка специалистов информационной безопасности, решения по обеспечению информационной безопасности.*

Введение

Существует тренд на повсеместное создание и применение киберполигонов, который обусловлен руководящими документами и объективной необходимостью подготовки специалистов информационной безопасности (ИБ), организации и проведения испытаний в сфере ИБ. Ориентировочная стоимость только одной аппаратно-программной части киберполигона составляет десятки миллионов рублей. При этом остаётся ряд нерешённых вопросов:

1) насколько эффективно киберполигон справляется с возложенными на него задачами;
2) какие временные, финансовые, организационные, человеческие ресурсы требуются для полноценного функционирования;
3) как киберполигон поведёт себя в условиях неопределённости и изменения объёма поставленных задач;
4) сможет ли он выдержать реальные внешние кибератаки и достигнуть поставленных целей;
5) где находятся пределы прочности киберполигона при увеличении нагрузки и другие аспекты.

Чтобы ответить на эти и другие вопросы, необходимо системно подойти к синтезу киберполигона как организационно-технической системы.

Анализ литературы

По большей части организационно-технические системы класса киберполигон строятся посредством рационального обобщения опыта противостояния киберугрозам и создания такой инфраструктуры, которая позволяет испытать и выбрать лучшие практики, а также реализовать их в нужном контексте.

В работе [1] на основе анализа более 200 источников международная группа исследователей представила статистические данные эволюции предметной области киберполигонов и их онтологий с учетом распределения сфер применения, участников, используемых методов проведения киберучений/кибертренировок и реализуемых сценариев

1 Грызунов Виталий Владимирович, доктор технических наук, доцент, профессор кафедры прикладной математики и информационных технологий Санкт-Петербургского университета ГПС МЧС России. Санкт-Петербург, Россия. E-mail: viv1313r@mail.ru, ORCID https://orcid.org/0000-0003-4866-217X

2 Шестаков Александр Викторович, доктор технических наук, старший научный сотрудник, ведущий научный сотрудник университета Санкт-Петербургского университета ГПС МЧС России. Санкт-Петербург, Россия. E-mail: alexandr.shestakov01@yandex.ru, ORCID https://orcid.org/0000-0002-8462-6515

кибератак, архитектур построения, стеков протоколов и технологий виртуализации.

На Международном семинаре *ESORICS 2023 International Workshops* представлены материалы исследования [2] существующих зарубежных платформ для применения в качестве киберполигонов, адаптированных с учетом методов организации обучения и экспериментов, информационных инфраструктур и их топологий, а также возможного применения искусственного интеллекта для их конфигурирования под различные целевые задачи – как эволюционный путь развития платформ следующего поколения.

Подбор аналитического материала [3] по проблематике усовершенствований киберплощадок для прикладных задач киберфизических систем и информационных сетевых систем базируется на методологических рекомендациях поиска статей по определенным критериям и оценкам их качества (*PRISMA*) и анализе более чем 100 специализированных работ, на основании которых подтверждаются системные проблемы в архитектуре и инфраструктуре киберплощадок, что приводит к значительному росту нагрузки при администрировании и управлении предоставляемыми сервисами и формируемой требуемой конфигурацией.

Норвежскими специалистами в [4] рассмотрены различные аспекты разработки и оценки так называемых «неклассифицированных» киберполигонов (киберплощадок), которые в отличие от применяемых для обучения специалистов в области информационной безопасности, привития навыков и повышения знаний о новейших киберугрозах, защите от них или смягчения последствий, предназначены для непрофильной аудитории с целью повышения их киберграмотности и киберкультуры (кибергигиены), в том числе для проведения тестирования безопасности.

Исследователи Чешского Университета им. Масарика (Брно) [5] представили отчет о десятилетнем опыте использования интеллектуального анализа данных поведенческих процессов участников киберучений/кибертренировок, таких как *Capture the Flag* с применением технологий *Domain-Driven Design* для моделирования процесса подготовки специалистов на базе киберполигона с целью улучшить качество их подготовки. Вместе с тем, инфраструктурные аспекты киберполигона и проблематика организационной части остаются за рамками исследования.

В работе [6] обосновываются технические решения по созданию ведомственной организационно-технической системы класса «киберполигон» на основе анализа существующих практик создания подобных систем, зафиксированных в руководящих документах. Итоговое техническое решение выбирается экспертами с применением метода анализа иерархий. Такой подход обладает определенной долей субъективизма и предполагает некоторую статичность объекта управления с чётко заданными границами.

Исследование [7] посвящено выбору рационального варианта формирования инфраструктуры киберполигона как мультифункциональной инфраструктуры при существующих организационно-технических, финансовых и прочих ограничениях. Задача решается методом перебора всех возможных вариантов, каждый из которых характеризуется своим интегральным показателем эффективности. Предполагается: во-первых, линейность системы управления; во-вторых, фиксированные границы системы; в-третьих, относительная стабильность структуры и функций киберполигона во времени.

Вместе с тем, синтез организационно-технических систем в условиях неопределённости изучался рядом авторов достаточно давно. В некоторых работах[3] принято, что системы информационной безопасности в конкурирующих производственно-экономических структурах организационно включают в свой состав совокупность связанных единством цели элементов информационной безопасности (ИБ) на уровнях организационно-технических систем, технических систем и комплексов средств информационной безопасности. В работе упоминается, что синтез системы ИБ выполняется в условиях нечёткости и неопределённости исходных представлений о её задачах, составе, структуре и функционировании, при этом предполагаются фиксированные во времени границы системы и линейность системы управления, что является довольно сильным ограничением. По существу, сформулированная в исследовании задача решается методом последовательных приближений.

Более гибкое и менее формальное использование метода последовательных приближений для создания организационно-технических систем заложено в технологии Agile (гибкой разработки программного обеспечения), которая исследуется в работе [8], применительно к задачам формирования всестороннего организационного обеспечения вновь вводимых технологических систем компании. Авторы проинтервьюировали 52 респондента из Англии и Германии, выделили 4 модели компаний: бимодальная, полностью гибкая с межпродуктовой поддержкой, гибкая организация с проектной деятельностью, полностью гибкая организация без проектной деятельности; и 7 путей миграции компаний к полностью гибкой организации. Предлагаемый авторами подход учитывает неопределённость, с которой сталкивается

3 Мистров Л. Е. Модель синтеза систем информационной безопасности организационно-технических систем // Информационная безопасность регионов. – 2011. – №. 1. – С. 21–33

компания, и позволяет управлять изменениями компании во времени, но слабо формализован и, с точки зрения применимости к киберполигону, охватывает лишь часть киберполигона как объекта управления, в частности, только обеспечивающий уровень и уровень персонала согласно детализированной в [9] модели *FIST* (*Full Infrastructure of Sources Toolkit*), что является недостаточным для полного формирования системы.

Ещё одной группой вариантов синтеза системы управления организационно-техническими системами класса киберполигон или похожих на них путём последовательного приближения выступают технологии бизнес-моделирования: *IDEF0*, которая применена в [10] при формировании национальных систем наращивания потенциала в области кибербезопасности для стран с переходным этапом развития (*NCCBF, National Cybersecurity Capacity Building Framework*); *BPM* (*Business Process Management*), которая принята за основу в моделях жизненного цикла процессов [11] при персонализации киберучений; *UML* (*Unified Modeling Language*), применённая в [12] для представления инновационных платформенных решений обеспечения кибербезопасности на основе моделей с проверкой полученных навыков обучающихся в рабочей среде (*CYRA, CYber Range Assurance platform*) и другие.

Синтез начинается с формализации показателей эффективности результирующей системы, затем так или иначе фиксируется точка зрения на систему (специалист информационной безопасности, управленец, пользователь системы и т.д.), после чего путём достаточного количества итераций, включающих в себя опрос специалистов, синтезируются функции и, возможно, некоторые элементы системы.

Эти технологии, в целом, позволяют учесть неопределённости в объекте управления, гибко изменять алгоритмы управления, однако есть ряд существенных недостатков:

- ✓ во-первых, эффективность их применения существенно зависит от квалификации сотрудников, которые выполняют синтез;
- ✓ во-вторых, фиксированные точки зрения дают несколько, порой не вполне связных между собой моделей системы, что порождает отдельную сложную задачу интеграции этих моделей в целостную модель;
- ✓ в-третьих, переход от синтеза функций к синтезу структуры идёт интуитивно.

Методика формирования допустимых вариантов организационного состава и структуры автоматизированной системы управления информационной безопасностью, предложенная в [13], применяет декомпозицию на подграфы исходного графа организационного состава и структуры АСУ кибербезопасностью. При этом вопрос определения количества уровней в графе остаётся не решенным. В Методике принято, что структура и функции объекта управления и его элементов статичны.

В опубликованных работах не удалось обнаружить подходы, позволяющие найти и формализовать условие существования киберполигона как организационно-технической системы. Исключение составляет работа [14], где описывается метод *iSOFT*, позволяющий сформулировать условие[4] существование системы, заданное в виде операторного уравнения. Когда получается найти такое условие, возможно синтезировать систему, которая гарантированно решает поставленные задачи.

Целью настоящего исследования является формулировка условия существования киберполигона как организационно-технической системы.

Особенности киберполигона как объекта исследования

Под киберполигоном в настоящем исследовании понимается инфраструктура для отработки практических навыков специалистов, экспертов разного профиля, руководителей в области информационной безопасности и информационных технологий, а также для тестирования программного и аппаратного обеспечения путем моделирования компьютерных атак и отработки реакций на них[5].

Следовательно, киберполигон должен за заданное время:

- ✓ во-первых, формировать навыки и умения у заданного количества специалистов ИБ;
- ✓ во-вторых, тестировать программное и аппаратное обеспечение в сфере ИБ.

Относительно ведомственного киберполигона, на примере МЧС России, конкретизация исходных данных формулируется следующим образом.

Свойства киберполигона МЧС России

Киберполигон с учетом специфики МЧС России, представляет собой не просто виртуальную среду, а единую организационно-техническую систему, состоящую из территориально-распределенных сегментов сил и средств, объединенных цифровой сетью связи, с централизованным управлением на базе образовательного учреждения МЧС России.

Ключевые особенности киберполигона для МЧС России:

4 Условие — Обстоятельство, от которого что-нибудь зависит (Ожегов С. И., Шведова Н. Ю. «Ожегов С. И. Толковый словарь русского языка: 72500 слов и 7500 фразеологических выражений» /С. И. Ожегов, Н. Ю. Шведова. – М. : Азъ, 1992. – 960 с.)

5 Постановление Правительства Российской Федерации «Об утверждении Правил предоставления субсидий из федерального бюджета на создание киберполигона для обучения и тренировки специалистов и экспертов разного профиля, руководителей в области информационной безопасности и информационных технологий современным практикам обеспечения безопасности» от 12.10.2019 № 1320.

- ведомственная принадлежность: киберполигон предназначен для решения задач, связанных с обеспечением информационной безопасности МЧС России, с учетом специфики деятельности ведомства и задач единой государственной системы предупреждения и ликвидации чрезвычайных ситуаций (РСЧС);
- многофункциональность: киберполигон объединяет несколько треков, каждый из которых направлен на решение определенных задач:
  - образовательный трек: предметно-ориентированное обучение и повышение квалификации специалистов по ИБ, интегрированное с электронной информационно-образовательной средой вузов МЧС России;
  - трек киберучений: организация киберучений, тренировок и соревнований для специалистов ИБ и руководителей (должностных лиц) МЧС России;
  - трек исследований и тестирования: апробация и тестирование новых технологий и средств защиты информации, исследование проблемных вопросов кибербезопасности, наполнение банка данных угроз ФСТЭК России;
- территориальная распределенность: киберполигон состоит из сегмента с функциями управления и территориальных сегментов, развернутых в различных подразделениях МЧС России;
- интеграция с информационной инфраструктурой МЧС России: киберполигон должен быть интегрирован с действующими системами (СЭД, КС АРМ ГС, ЕДДС АИУС РСЧС и т.д.), а также иметь возможность взаимодействия с внешними системами, например, ГосСОПКА;
- оперирование пространственными данными (данными о пространственных объектах и их наборах): расположение оборудования, объектов критической информационной инфраструктуры, геолокация пользователей и пр., следовательно, информационная система в основе киберполигона является геоинформационной системой[6];
- масштабируемость и развитие: киберполигон должен обеспечивать возможность поэтапного наращивания мощностей, добавления новых функций, модернизации и адаптации к новым задачам и угрозам.

Таким образом, в киберполигоне МЧС России присутствует несколько явно выраженных уровней, связанных с документальным сопровождением, работы персонала и аппаратно-программных средств, взаимодействие которых рассматривается в модели *FIST*.

Киберполигон согласно модели FIST

Киберполигон МЧС России представляет собой сложную организационно-техническую систему, включающую в себя не только программное и аппаратное обеспечение, но и персонал, пользователей, нормативно-правовые документы, финансовые потоки и другие взаимосвязанные элементы. Традиционные исследования информационной безопасности рассматривают эти элементы изолированно, упуская из виду их взаимосвязь и влияние друг на друга, что усложняет формализацию условия существования киберполигона, то есть условия, выполняя которое, киберполигон гарантированно достигает своей цели деятельности.

Модель *FIST* (*Full Infrastructure of Sources Toolkit*) [9] позволяет рассмотреть киберполигон как иерархическую систему с обеспечивающим уровнем, уровнем персонала, уровнями аппаратного и программного обеспечения (см. рис. 1).

6 ГОСТ Р 52155-2003 Географические информационные системы федеральные, региональные, муниципальные. Общие технические требования.

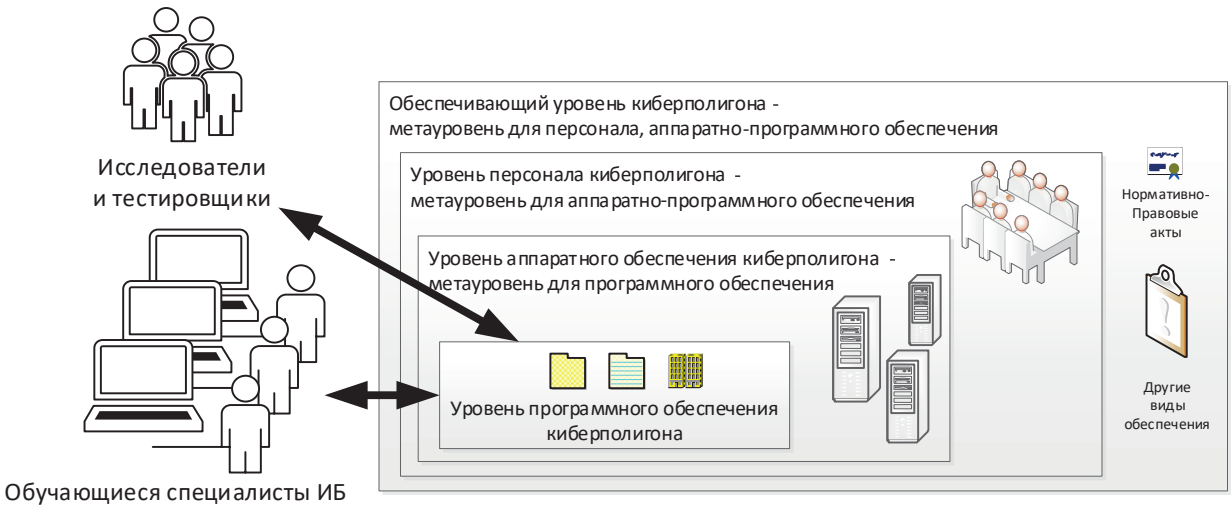


Рис. 1. Киберполигон согласно модели FIST

Метауровни задают требуемые пространственно-временные состояния вложенных уровней, например: руководящие документы формируют требования к персоналу, аппаратно-программному обеспечению; персонал настраивает и обеспечивает функционирование аппаратно-программного обеспечения; аппаратура предоставляет заданные ресурсы программному обеспечению.

От вложенных уровней в сторону метауровней идёт обратная связь, например: выбранное программное обеспечение не позволяет организовать многопользовательскую подготовку специалистов ИБ, распределённых в пространстве времени, что может потребовать применения распределённого в пространстве-времени аппаратного обеспечения, что в свою очередь изменяет требования к обслуживающему персоналу, что влечёт за собой изменения в руководящих документах или финансовом обеспечении.

Киберполигон существует на всех уровнях модели *FIST*:

- ✓ во-первых, на обеспечивающем уровне:
  - а) в виде нормативно-правового обеспечения – регламентов, политики, распоряжений и т.д.;
  - б) как система, оперирующая с финансовыми средствами;
  - в) содержит регламентированные профессиональные требования к специалистам ИБ, мотивационные и воспитательные составляющие;
- ✓ во-вторых, на уровне персонала в качестве преподавателей и вспомогательного персонала;
- ✓ в-третьих, на аппаратном уровне как оборудование, на котором развёрнуто программное обеспечение, необходимое для жизнедеятельности киберполигона;
- ✓ в-четвертых, на уровне программного обеспечения как набор специальных и общесистемных программ, с использованием которых:
  - а) осуществляется непосредственная подготовка специалистов ИБ;
  - б) выполняются действия, обеспечивающие работу киберполигона: бухгалтерия, кадры, резервное копирование и пр.

Уровни непрерывно взаимодействуют между собой и ориентированы на достижение цели деятельности всего киберполигона, и значит, система, синтезированная с использованием модели *FIST*, является целостной [14].

Множество пространственно-временных состояний киберполигона S состоит из множеств пространственно-временных состояний каждого уровня.

$$S = S^E \cup S^P \cup S^{Hard} \cup S^{Soft}, \quad (1)$$

где S^E – множество пространственно-временных состояний обеспечивающего уровня; S^P – множество пространственно-временных состояний уровня персонала; S^{Hard} – множество пространственно-временных состояний уровня аппаратного обеспечения; S^{Soft} – множество пространственно-временных состояний уровня программного обеспечения.

Пространственно-временное состояние системы – это сложившиеся отношения между элементами системы на момент времени.

Производительность киберполигона

Киберполигон предназначен для решения задач по подготовке специалистов ИБ (*e, education*) и для проведения испытаний в сфере ИБ (*test, testbeds*). Значит, можно сказать, что он обладает производительностью

$$\Omega_{CR} = \{\Omega_e, \Omega_{test}\}, \quad (2)$$

где Ω_e – производительность киберполигона по подготовке специалистов ИБ: количество специалистов в единицу времени; Ω_{test} – производительность киберполигона по проведению испытаний: количество испытаний в единицу времени.

Производительность (Ω) – количество задач $|K|$, решённое за время t:

$$\Omega = |K| / t, \quad (3)$$

где K – множество решаемых задач.

Множество задач K^*, которые должен решить киберполигон, определяется метасистемой-заказчиком, то есть задаётся извне.

Производительность всего киберполигона зависит от производительности каждого уровня киберполигона, от того, насколько элементы согласованы между собой:

$$\Omega_{CR} = F_\Omega(\Omega^E, \Omega^P, \Omega^{Hard}, \Omega^{Soft}, S, T), \quad (4)$$

где Ω^E – производительность обеспечивающего уровня: скорость разработки нормативно-правовых документов, срок действия документов, объём финансирования в единицу времени и т.д. Ω^P – производительность уровня персонала: количество задач, решаемых персоналом в единицу времени, «время жизни» персонала и т.д.; Ω^{Hard} – производительность уровня аппаратного обеспечения: *MIPS*, *FLOPS*, бод и т.д.; Ω^{Soft} – производительность уровня программного обеспечения: скорость сходимости реализованных алгоритмов, вычислительная сложность алгоритмов, ресурсоёмкость применяемых команд и т.д.; T – множество моментов времени, в которые функционирует киберполигон.

Среда функционирования киберполигона

Среда, воздействие которой обрабатывает киберполигон, имеет разную природу [15]:

- ✓ во-первых, детерминированная среда (Q_d), воздействие которой известно заранее и может быть

описано аналитически: техническое обслуживание, расписание подготовки специалистов ИБ, проведения испытаний и т.д.;

- ✓ во-вторых, стохастическая среда (Q_{st}), воздействие которой на систему выбирается из *известного множества альтернатив случайным образом при полностью известном вероятностном описании «механизма» этого выбора*: естественные сбои и отказы, поток задач, согласованных с метасистемой-заказчиком и т.д.;
- ✓ в-третьих, среда нестохастическая (Q_{nst}), то есть среда, которая не является средой Q_d и Q_{st}. Эта среда характеризуется тем, что: а) воздействие на киберполигон выбирается из известного множества альтернатив согласно некоторой цели либо отсутствуют некоторые элементы вероятностного описания «механизма» выбора; б) воздействие на киберполигон не описывается в рамках других сред: новые неучтённые ранее задачи, поставленные метасистемой-заказчиком, новая активность злоумышленников, изменения в ландшафте киберугроз и пр.

Назначение киберполигона – тренировать специалистов ИБ, которые и атакуют инфраструктуру, и защищают её. Специалисты ИБ должны иметь актуальные навыки в сфере ИБ, то есть деятельности, которая сильно и непредсказуемо изменчива.

Следовательно, киберполигон:

а) функционирует в условиях изменчивости цели управления: нужно готовить требуемое количество специалистов ИБ с актуальными знаниями, при этом требуемое количество специалистов и актуальность знаний изменчива;

б) поскольку киберполигон имеет ведомственную принадлежность с некоторой автономией на местах, то управление им будет сочетать в себе элементы и централизации, и самоорганизации;

в) архитектура киберполигона в виде совокупности структуры и протоколов взаимодействия элементов структуры между собой и со средой практически непрерывно изменяется в пространстве – времени;

г) имеет тенденцию саморазрушаться, что является штатным режимом функционирования и должно учитываться управляющей системой;

д) может подвергаться нестохастическим воздействиям внешних систем: кибератаки, резкое изменение требований к количеству и качеству подготавливаемых специалистов ИБ, появление новых угроз и т.д.

Таким образом, киберполигон представляет собой объект изменяющейся целью управления, с динамично изменяемой архитектурой, функционирующий в среде всех возможных типов, и значит, целесообразно его рассматривать как адаптивную систему управления[7].

Система управления – это сочетание управляющей системы и объекта управления[8].

Операторное уравнение, описывающее работу системы управления киберполигоном

Цель управления киберполигоном – решить множество поставленных задач K^* за заданное время t, несмотря на воздействия окружающей среды и саморазрушение киберполигона.

Следовательно, должна быть разработана система показателей, описывающая насколько киберполигон способен достичь своей цели деятельности.

Решить проблему управления – решить проблему выбора из множества альтернатив[9].

$$\{U_{\text{доп}}, S\} \longrightarrow U_{sat}, \qquad (5)$$

где $U_{\text{доп}}$ – множество допустимых управляющих воздействий; U_{sat} – множество управляющих воздействий, реализующих сатисфакционное управление киберполигоном и адаптирующих киберполигон к обработке входящих воздействий.

Киберполигон является децентрализованной системой, распределённой в пространстве-времени, следовательно, в произвольное время к киберполигону подключаются или отключаются от него сегменты с разными наборами управляющих воздействий.

Это означает, что множество допустимых управляющих воздействий $U_{\text{доп}}$ изменяется во времени.

Задача управления может быть упрощена, если потребовать неизменность $U_{\text{доп}}$. Физически неизменность реализуется разработкой соответствующих регламентов и созданием на устройствах специальной среды для решения задач киберполигона: виртуальная машина, контейнер или какой-то другой вариант виртуализации.

Ограничение 1. Множество $U_{\text{доп}}$ неизменно во все моменты времени из множества T.

В ходе адаптации необходимо определить объект управления. Это означает, что задача адаптивного управления распадается на две крупные части:

- идентификация объекта управления, то есть наблюдение;
- выбор управляющего воздействия.

Следовательно, модель киберполигона примет вид, приведенный на рис. 2.

7 Цыпкин, Я. З. Основы теории автоматических систем: учеб. пособие для вузов / Я. З. Цыпкин.– М. : Наука, 1977. – 560 с.

8 Растригин, Л. А. Адаптация сложных систем / Л. А. Растригин. – Рига: Зинатне, 1981. – 375 с.

9 Калинин, В. Н. Теоретические основы системных исследований: краткий авторский курс лекций для адъюнктов академии / В. Н. Калинин. – СПб.: ВКА им. А. Ф. Можайского, 2011. – 278 с.

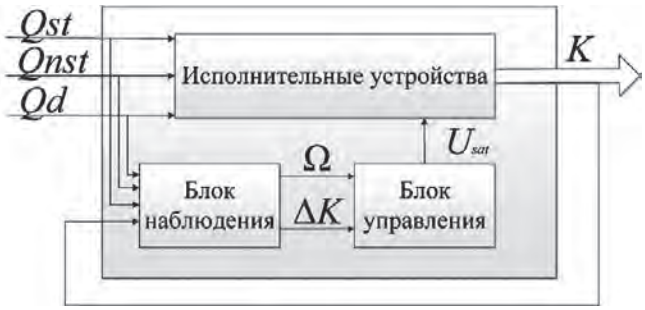


Рис. 2. Модель киберполигона с блоком наблюдения и управления

Согласно предложенной модели, наблюдение за киберполигоном и управление киберполигоном реализуются через производительности. Интегральная производительность формируется через оптимизацию структуры киберполигона и оптимизацию процессов выполнения задач.

Таким образом, проблема адаптации киберполигона к обработке входящих воздействий формулируется так.

Дано:

T – множество моментов времени, когда функционирует киберполигон;

$Q = \{Q_d, Q_{st}, Q_{nst}\}$ – множество входных ситуаций;

$\Omega_{CR} = F_\Omega\ (\Omega^E, \Omega^P, \Omega^{Hard}, \Omega^{Soft}, S, T)$ – производительность киберполигона;

$K^* = \cup_{i=1}^{|K^*|} k_i,\ k_i = \langle \Omega^*_{CR,i}, t_i^* \rangle$ – множество поставленных задач;

t_i^* – требуемое время решения i-ой задачи;

$\Omega^*_{CR,i}$ – производительность киберполигона, которая требуется i-ой задаче;

$U_{\text{доп}}$ – множество управляющих воздействий.

Ограничение 2. $t_i \ll \tau,\ \forall i = \overline{(1,|K^*|)}$, τ – среднее время стабильного существования киберполигона.

Требуется:

При фиксированном $U_{\text{доп}}$ найти такое управляющее воздействие $U_{sat} \subset U_{\text{доп}}$, которое позволит решить все задачи, поставленные перед киберполигоном за заданное время, то есть найти оператор:

$$R_U\,(\Omega_{CR}, Q, U_{sat}, T) = K^*. \qquad (5)$$

Операторное уравнение (5) является моделью системы адаптивного управления киберполигоном и реализует условие адаптации киберполигона к обработке входящих воздействий, то есть условие гарантированного решения киберполигоном поставленных задач. При создании уравнения используются все возможные субстанциальные закономерности киберполигона, то есть закономерности, которые влияют на достижение системой цели её деятельности [14].

Решение операторного уравнения (5) представляет собой вариационную задачу, так одно уравнение содержит несколько переменных.

Из представленной модели следует, что множество задач K, которые решает киберполигон, может не совпадать со множеством задач K^*, поставленных метасистемой-заказчиком перед киберполигоном. Так происходит в следующих случаях:

1) киберполигон решает меньше задач, чем поставили, потому что не справляется с нагрузкой в силу резкого увеличения задач или своего разрушения из-за кибератак или в ходе эксплуатации;
2) киберполигон решает задачи злоумышленников.

Пример применения разработанной модели

Без нарушения общности предположим, что перед киберполигоном стоит задача повышать квалификацию 10 специалистам ИБ в месяц:

$\Omega_{CR} = \Omega_e$ = 10 специалистов в месяц на протяжении года.

Киберполигон функционирует T = 1 год = 12 месяцев.

k^* – подготовить 10 специалистов ИБ.

$|K^*|$ = 12, так как всего 12 задач (10 специалистов каждый месяц на протяжении года).

$K^* = \cup_{i=1}^{12} k_i,\ k_i = \langle \Omega^*_{CR,i} = 10,\ t_i^* = 1 \rangle$.

Q_{st} = равномерное появление 14±5 специалистов ИБ в месяц желающих повысить свою квалификации.

Q_{nst} = текущая ситуация в сфере ИБ, которая влияет на формирование перечня требований к специалисту ИБ. Может изменяться раз в неделю.

Исходя из условий примера, можно сказать, что:

1) киберполигон должен содержать какой-то элемент, который преобразует стохастическую величину на входе Q_{st} в детерминированную Ω_{CR} на выходе, например, за счёт управления количеством обучающихся в группе;
2) полностью формировать перечень требований к специалистам ИБ на уровне руководства МЧС нецелесообразно, потому что текущая ситуация в сфере ИБ Q_{nst}, влияющая на формирование требований, изменяется раз в неделю, что гораздо быстрее, чем формируются и утверждаются документы в МЧС (больше года, то есть 52 недели). Следовательно, частично требования к специалисту ИБ необходимо формировать в ходе самого обучения;
3) среднее время стабильного существования киберполигона должно быть много больше одного месяца.

Конкретный вид оператора (5) и его составляющих определяется на стадиях создания автоматизированных систем 4 «Эскизный проект» и 5 «Технический проект»[10] и реализуется методом *iSOFT* [14]. Этому посвящены последующие работы.

10 ГОСТ Р 59793-2021. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.

Поскольку оператор строится с использованием всех субстанциальных закономерностей [14], то пригодность и адекватность модели обеспечивается полнотой учёта закономерностей.

Выводы

Фактически представленные материалы конкретизируют базовые мероприятия стадии 2 «Разработка концепции автоматизированной системы» при создании автоматизированной системы.

В статье сформулированы два ограничения, при которых должна решаться задача синтеза системы управления киберполигоном.

Предлагаемая в виде операторного уравнения модель киберполигона как организационно-технической системы в отличие от существующих моделей *формализует условие гарантированного решения поставленных киберполигоном задач*, связывает все уровни киберполигона, выдвигает требование ко времени стабильного существования киберполигона и позволяет выявить необходимость:

- ✓ во-первых, рассмотрения киберполигона как адаптивной системы управления с изменяющейся архитектурой на всех уровнях, учитывающей возможность изменения цели деятельности;
- ✓ во-вторых, присутствия блока наблюдения и формирования обратной связи;
- ✓ в-третьих, формирования множества допустимых управляющих воздействий, фиксированных на весь период существования киберполигона;
- ✓ в-четвертых, разработки иерархии показателей оценивания достижения киберполигоном цели деятельности;
- ✓ в-пятых, поиска конкретного вида операторного уравнения (5) и его сатисфакционного решения.

Разработанная модель может применяться на всех этапах жизненного цикла киберполигона: от разработки до утилизации.

Литература

1. *Ukwandu, E. et al. A review of cyber-ranges and test-beds: Current and future trends // Sensors. – 2020. – v. 20(24). – №. 24. – P. 7148. DOI:10.3390/s20247148.*
2. *Grimaldi, A. et al. Toward Next-Generation Cyber Range: A Comparative Study of Training Platforms / In book: Computer Security. ESORICS 2023 International Workshops. Pp.271–290. DOI:10.1007/978-3-031-54129-2_16.*
3. *Stamatopoulos, D. et al. Exploring the Architectural Composition of Cyber Ranges: A Systematic Review // Future Internet, 16(7), June 2024. – P.16. DOI:10.3390/fi16070231.*
4. *Yamin, M., Katt, B., Gkioulos, V. Cyber Ranges and Security Testbeds: Scenarios, Functions, Tools and Architecture // Computers & Security. – October 2019. – v. 88. – P. 101636. DOI:10.1016/j.cose.2019.101636.*
5. *Macák, M., Oslejsek, R., Buhnova, B. Applying process discovery to cybersecurity training: an experience report // 2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). – IEEE, 2022. – Pp. 394–402. DOI:10.1109/EuroSPW55150.2022.00047.*
6. *Синещук М. Ю. Технические решения по созданию ведомственных организационно-технических систем класса «киберполигон» как средства обеспечения информационной безопасности ведомственного назначения // Научно-аналитический журнал «Вестник Санкт-Петербургского университета ГПС МЧС России». 2024. №.1. С. 179–200. DOI: https://doi.org/10.61260/2218-130X-2024-1-179-20.*
7. *Матвеев А. В., Синещук М. Ю., Шестаков А. В., Гавкалюк Б. В. Методика технико-экономической оценки вариантов построения организационно-технической системы класса «киберполигон» // Инженерный вестник Дона. – 2023. – №. 6 (102). – С. 187–200.*
8. *Gerster, D. et al. How Enterprises Adopt Agile Forms of Organizational Design: A Multiple-Case Study //ACM SIGMIS Database: the DATABASE for Advances in Information Systems. – 2020. – v. 51. – №. 1. – Pp. 84–103. DOI:10.1145/3380799.3380807.*
9. *Грызунов В. В. Модель геоинформационной системы FIST, использующей туманные вычисления в условиях дестабилизации // Вестник Дагестанского государственного технического университета. Технические науки. 2021. Т. 48. №. 1. С. 76–89. DOI: 10.21822/2073-6185-2021-48-1-76-89.*
10. *Naseir, M. A. B. National cybersecurity capacity building framework for counties in a transitional phase : Doctoral Thesis (Doctoral). – Bournemouth University. 2020.*
11. *Pfaller, T. et al. Towards Customized Cyber Exercises using a Process-based Lifecycle Model // EICC '24: Proceedings of the 2024 European Interdisciplinary Cybersecurity Conference: Association for Computing Machinery (ACM), New York, pp. 37–45.*
12. *Smyrlis, M. et al. CYRA: A Model-Driven CYber Range Assurance Platform // Applied Sciences. – 2021. – v. 11. – №. 11. – P. 5165. DOI:10.3390/app11115165/.*
13. *Селифанов В. В., Мещеряков Р. В. Методика формирования допустимых вариантов организационного состава и структуры автоматизированной системы управления информационной безопасностью // Моделирование, оптимизация и информационные технологии. – 2020. – Т. 8. – №. 1. – С. 39-40. DOI: 10.26102/2310-6018/2020.28.1.001.*
14. *Грызунов В. В. Формирование условия гарантированного достижения цели деятельности информационной системой на базе операторного уравнения // Информатизация и связь. – 2022. – № 4. – С. 67–74. – DOI 10.34219/2078-8320-2022-13-4-67-74.*
15. *Burlov, V. G., Gryzunov, V. V., Tatarnikova, T. M. Threats of information security in the application of GIS in the interests of the digital economy // Journal of Physics: Conference Series : 23 (St. Petersburg, 27–29.05.2020). – St. Petersburg : IOP Publishing Ltd, 2020. – P. 012023. – DOI: 10.1088/1742-6596/1703/1/012023.*
16. *Грызунов, В. В. Концептуальная модель адаптивного управления геоинформационной системой в условиях дестабилизации // Проблемы информационной безопасности. Компьютерные системы. – 2021. – № 1. – С. 102–108. – EDN GVCRHF.*

MODEL OF THE ADAPTIVE CONTROL SYSTEM OF THE CYBER RANGE OF THE RUSSIAN EMERGENCIES MINISTRY BASED ON THE OPERATOR EQUATION

Gryzunov V. V.[11], Shestakov A. V.[12]

The purpose of the research *is to formulate the conditions for the existence of a cyber range as an organizational and technical system that is guaranteed to solve the tasks.*

Research methods: *the proposed model is based on the FIST model, methods of the theory of adaptive control.*

Results: *1) it is shown that the cyber range of the Ministry of Emergency Situations of Russia has several tracks according to the areas of the tasks to be solved, is geographically distributed, integrates with the information infrastructure of the Ministry of Emergency Situations, operates with spatial data, functions in an environment of all possible types and manifests itself in the form of a set of performance levels of the supporting level, the level of personnel, the level of hardware and the level of software; 2) the limitations under which it is possible to synthesize a cyber range as an adaptive control system with a changing architecture are formulated: on the stability of a set of control actions, on the average time of stable existence of a cyber range; 3) the operator equation describing the cyber range as an organizational and technical system maintained by personnel and characterizing the condition for the guaranteed solution of the tasks facing the cyber range has been formalized; 4) the introduction of new elements into the structure of the cyber range is substantiated: an observation unit and a control unit taking into account feedback.*

Scientific novelty: *the author provides a model of a cyber range, which is distinguished by the formalization of the conditions for the existence of a cyber range as an organizational and technical system at all levels (support, personnel, hardware, software).*

Discussion: *A specific kind of operator equation formalized in the paper can be found using the iSOFT method.*

Keywords: *information security management models, synthesis of organizational and technical systems, training of information security specialists, information security solutions.*

References

1. *Ukwandu, E. et al. A review of cyber-ranges and test-beds: Current and future trends // Sensors. – 2020. – v. 20(24). – №. 24. – P. 7148. DOI:10.3390/s20247148.*
2. *Grimaldi, A. et al. Toward Next-Generation Cyber Range: A Comparative Study of Training Platforms / In book: Computer Security. ESORICS 2023 International Workshops. Pp.271–290. DOI:10.1007/978-3-031-54129-2_16.*
3. *Stamatopoulos, D. et al. Exploring the Architectural Composition of Cyber Ranges: A Systematic Review // Future Internet, 16(7), June 2024. – R.16. DOI:10.3390/fi16070231.*
4. *Yamin, M., Katt, B., Gkioulos, V. Cyber Ranges and Security Testbeds: Scenarios, Functions, Tools and Architecture // Computers & Security. – October 2019. – v. 88. – P. 101636. DOI:10.1016/j.cose.2019.101636.*
5. *Macák, M., Oslejsek, R., Buhnova, B. Applying process discovery to cybersecurity training: an experience report //2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). – IEEE, 2022. – Pp. 394–402. DOI:10.1109/EuroSPW55150.2022.00047.*
6. *Sineshhuk M. Ju. Tehnicheskie reshenija po sozdaniju vedomstvennyh organizacionno-tehnicheskih sistem klassa «kiberpoligon» kak sredstva obespechenija informacionnoj bezopasnosti vedomstvennogo naznachenija // Nauchno-analiticheskij zhurnal «Vestnik Sankt-Peterburgskogo universiteta GPS MChS Rossii». 2024. №.1. S. 179–200. DOI: https://doi.org/10.61260/2218-130X-2024-1-179-20.*
7. *Matveev A. V., Sineshhuk M. Ju., Shestakov A. V., Gavkaljuk B. V. Metodika tehniko-jekonomicheskoj ocenki variantov postroenija organizacionno-tehnicheskoj sistemy klassa «kiberpoligon» // Inzhenernyj vestnik Dona. – 2023. – №. 6 (102). – S. 187–200.*
8. *Gerster, D. et al. How Enterprises Adopt Agile Forms of Organizational Design: A Multiple-Case Study //ACM SIGMIS Database: the DATABASE for Advances in Information Systems. – 2020. – v. 51. – №. 1. – Pp. 84–103. DOI:10.1145/3380799.3380807.*
9. *Gryzunov V. V. Model' geoinformacionnoj sistemy FIST, ispol'zujushhej tumannye vychislenija v uslovijah destabilizacii // Vestnik Dagestanskogo gosudarstvennogo tehnicheskogo universiteta. Tehnicheskie nauki. 2021. T. 48. №. 1. S. 76–89. DOI: 10.21822/2073-6185-2021-48-1-76-89.*
10. *Naseir, M. A. B. National cybersecurity capacity building framework for counties in a transitional phase : Doctoral Thesis (Doctoral). – Bournemouth University. 2020.*

11 Vitaliy V. Gryzunov, Dr.Sc., Associate Professor, professor of department of applied mathematics and information technologies, St. Petersburg University of State Fire Service of EMERCOM of Russia, E-mail: viv1313r@mail.ru, ORCID https://orcid.org/0000-0003-4866-217X

12 Alexander V. Shestakov, Dr.Sc., senior researcher, leading researcher, St. Petersburg University of State Fire Service of EMERCOM of Russia. E-mail: alexandr.shestakov01@yandex.ru, ORCID https://orcid.org/0000-0002-8462-6515

11. Pfaller, T. et al. Towards Customized Cyber Exercises using a Process-based Lifecycle Model // EICC '24: Proceedings of the 2024 European Interdisciplinary Cybersecurity Conference: Association for Computing Machinery (ACM), New York, pp. 37–45.
12. Smyrlis, M. et al. CYRA: A Model-Driven CYber Range Assurance Platform //Applied Sciences. – 2021. – v. 11. – №. 11. – P. 5165. DOI:10.3390/app11115165/.
13. Selifanov V. V., Meshherjakov R. V. Metodika formirovanija dopustimyh variantov organizacionnogo sostava i struktury avtomatizirovannoj sistemy upravlenija informacionnoj bezopasnost'ju //Modelirovanie, optimizacija i informacionnye tehnologii. – 2020. – T. 8. – №. 1. – S. 39-40. DOI: 10.26102/2310-6018/2020.28.1.001.
14. Gryzunov V. V. Formirovanie uslovija garantirovannogo dostizhenija celi dejatel'nosti informacionnoj sistemoj na baze operatornogo uravnenija // Informatizacija i svjaz'. – 2022. – № 4. – S. 67–74. – DOI 10.34219/2078-8320-2022-13-4-67-74.
15. Burlov, V. G., Gryzunov, V. V., Tatarnikova, T. M. Threats of information security in the application of GIS in the interests of the digital economy // Journal of Physics: Conference Series : 23 (St. Petersburg, 27–29.05.2020). – St. Petersburg : IOP Publishing Ltd, 2020. – P. 012023. – DOI: 10.1088/1742-6596/1703/1/012023.
16. Gryzunov, V. V. Konceptual'naja model' adaptivnogo upravlenija geoinformacionnoj sistemoj v uslovijah destabilizacii / V. V. Gryzunov // Problemy informacionnoj bezopasnosti. Komp'juternye sistemy. – 2021. – № 1. – S. 102–108. – EDN GVCRHF.

CYBERSECURITY ISSUES

SCIENTIFIC PEER-REVIEWED JOURNAL

2024, № 6 (64)

Cybersecurity Issues is a research periodical scientific and practical publication specializing in information security.
Published six times a year

https://cyberrus.info

The journal is being published from 2013
(Registration Certificate PI No. FS 77-75239).
CrossRef number (DOI): 10.21681/2311-3456

The journal is included in the Russian list of peer-reviewed academic publications of the Higher Attestation Commission (VAK), it is registered in the Russian Science Citation Index (RSCI/RINTs) on the Web of Science (WoS) platform and holds the 1st place in its cyber security rating. The journal's articles are available in full text

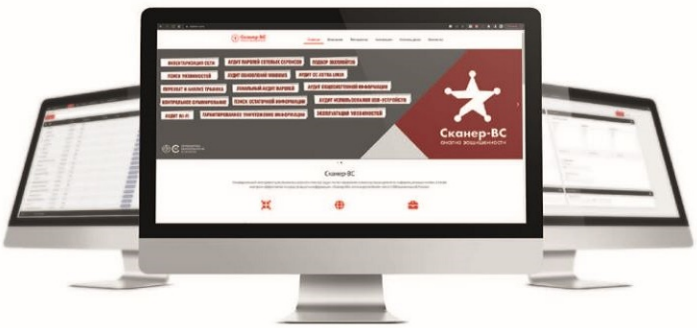
Founder and publisher
JSC «NPO «Echelon»
Postal address: Elektrozavodskaya str., 24, bld. 1, 107023, Moscow, Russia
E-mail: editor@cyberrus.info

CONTENTS

Эшелон
комплексная безопасность

Сканер-ВС
анализ защищенности

СКАНИРОВАНИЕ НА УЯЗВИМОСТИ НИКОГДА НЕ БЫЛО ТАКИМ БЫСТРЫМ!



ГК «Эшелон» представляет новый релиз системы управления уязвимостями Сканер-ВС 6. Сканер-ВС используется более чем в 5 000 организаций в России и позволяет как проводить периодическое сканирование на поиск уязвимостей, так и организовать непрерывный контроль защищенности.

Решение является ключевым компонентом, позволяющим внедрить эффективный процесс управления уязвимостями.

Скачать демо-версию «Сканер-ВС 6» (количество IP: 16, пробный период: 2 месяца) можно на сайте продукта: https://scaner-vs.ru/.

Получить техническую консультацию в группе продукта в телеграм: https://t.me/scanervs

Высокая скорость поиска

Сканер-ВС 6 обладает высокой скоростью поиска уязвимостей благодаря технологии «без скриптов»

Актуальная база уязвимостей

Ежедневно обновляемая база данных уязвимостей позволяет держать руку на пульсе последних изменений

Комплексный подход

Комплексное тестирование защищенности позволяет выявлять максимальное количество нарушений ИБ

Работа в защищенной среде

Работа в среде защищенной операционной системы Astra Linux 1.7

Отчетность

Единая среда для проведения тестирования и формирования отчетов, содержащих различную информацию в зависимости от степени детализации

Исполнение

Наличие исполнений в виде дистрибутива под Astra Linux 1.7 и LiveUSB с предустановленной ОС и с поддержкой режима сохранения изменений.

CYBERSECURITY ISSUES
VOPROSY KIBERBEZOPASNOSTI

№6 2024

DOI: 10.21681/2311-3456

- Information security risk management
- Secure artificial intelligence
- Software resource security

www.cyberrus.info
editor@cyberrus.info