МОДЕЛЬ КОЛИЧЕСТВЕННОГО ОЦЕНИВАНИЯ АГЕНТА СЛОЖНОЙ СЕТИ В УСЛОВИЯХ НЕПОЛНОЙ ИНФОРМИРОВАННОСТИ

Калашников А.О.[1], Бугайский К.А.[2]

Цель статьи: *разработка механизма количественного оценивания элементов сложных информационных систем в условиях недостаточной информации о наличии уязвимостей.*

Метод исследования: *математическое моделирование оценки неопределенности на основе бинарной свертки и сложности Колмогорова. В качестве исходных данных для моделирования используются банки данных по уязвимостям (vulnerabilities) и слабостям (weakness).*

Полученный результат: *показано, что работа элемента сложной сети может быть представлена процедурами преобразования данных, которые состоят из последовательности во времени операций, описываемых слабостями и сопутствующими уязвимостями. Каждая операция может быть на качественном уровне оценена с точки зрения тяжести последствий в случае реализации потенциальных слабостей. Применение бинарной свертки и универсального кодирования позволяют перевести качественные оценки в двоичную последовательность – слово в алфавите {0,1}. Последовательность таких слов - как функция неопределенности - описывает возможные негативные последствия реализации процедур преобразования данных, обусловленные наличием слабостей в элементе сложной системы. Предложено использовать Колмогоровскую сложность для количественной оценки функции неопределенности. Использование машины Тьюринга для вычисления функции неопределенности предоставляет универсальный механизм оценки с точки зрения информационной безопасности элементов сложных информационных систем независимо от их программной и аппаратной реализации.*

Ключевые слова: *уязвимость, бинарная свертка, сложность Колмогорова, машина Тьюринга, универсальное кодирование, модель информационной безопасности, оценка сложных систем.*

DOI:10.21681/2311-3456-2021-6-26-35

Введение

Основным трендом в развитии современных информационных систем является широкое использование облачных технологий, включая, в том числе, такие варианты как: «Функция-как-Сервис» (FaaS) и «граничные вычисления» (Edge). Также необходимо отметить параллельный бурный рост использования киберфизических устройств в различных областях человеческой деятельности, взаимодействующая совокупность которых, называется обычно «Интернет вещей» (IoT). В целом же можно говорить о том, что современные информационные системы строятся на основе микросервисов и архитектуры «Инфраструктура как Код» (IaC) [1]. Как результат, вопросы информационной безопасности таких систем приходится рассматривать в рамках сложной сетевой структуры, являющейся по своей сути многоагентной системой (МАС, *Multi-agent system*), где роль агентов играют отдельные программно-аппаратные комплексы (далее – ПАК) указанных систем При этом, если учесть, что практически во всех современных вычислительных средствах даже процессоры могут быть описаны как набор микропрограмм, то при моделировании интеллектуального агента можно исходить из следующих предположений.

Предположение 1. Сложную сетевую структуру составляют интеллектуальные агенты, реализующие свои функции преимущественно программным способом.

Предположение 2. Описание интеллектуального агента сложной сети с точки зрения информационной безопасности (далее – ИБ) необходимо должно включать характеристики программного обеспечения (далее – ПО), обеспечивающие его функционирование.

Вопросы оценки качества ПО находятся в центре внимания исследователей. Также на нормативном уровне оценка свойств ПО определяется, в частности, через построение профилей защиты и стандартизации методов оценивания [2-8]. Вместе с тем, как показы-

1 Калашников Андрей Олегович, доктор технических наук, главный научный сотрудник лаборатории «Сложных сетей» ФГБУН Институт проблем управления им. В. А. Трапезникова РАН, г. Москва, Россия. E-mail: aokalash@ipu.ru

2 Бугайский Константин Алексеевич, младший научный сотрудник лаборатории «Сложных сетей» ФГБУН Институт проблем управления им. В. А. Трапезникова РАН, г. Москва, Россия. E-mail: kabuga@ipu.ru

вает практика[3] ИБ [9, 10], при рассмотрении работы агентов в составе сложных систем полагаться только на качество разработки, регламентируемое документами и методиками, недостаточно.

В этой связи, представляется целесообразным определить ряд дополнительных характеристик ПО агентов, не только ориентированных на оценку ИБ, но и позволяющих учесть динамику работы ПО и взаимодействия его составных частей. С этой точки зрения описание интеллектуального агента должно представлять собой отражение его свойств или влияющих на процессы обработки информации в системе, или подверженных влиянию извне в процессе обработки информации.

Модель агента

Интеллектуальный агент, как ПАК, может быть представлен в самом общем случае в виде конечного автомата $\langle S, D^{in}, D^{out}, \sigma, \delta \rangle$, где: S – множество состояний конечного автомата, D^{in} – множество входных данных, D^{out} – множество выходных данных, σ – функция переходов, δ – функция выходов.

Функции автомата могут быть представлены как: $\sigma : D^{in} \times S \to S$ и $\delta : D^{in} \times S \to D^{out}$.

Тогда с точки зрения категорного подхода (см. рисунок 1) может быть введен морфизм $\tau = \sigma^{\circ}\delta$, который соответствует работе агента по преобразованию данных.

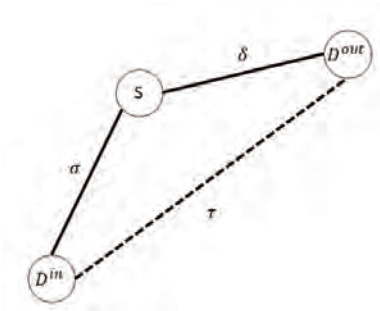

Рис.1. Морфизмы автомата

Входные и выходные данные агента могут относиться к разным типам. Например, входные данные могут быть и командами управления, а выходные данные – сообщениями об ошибках. Также входные и выходные данные могут относиться к решению задач маршрутизации полезной нагрузки (протоколы ICMP, ARP, DNS), синхронизации времени (протокол NTP) или аутентификации (протоколы Kerberos, RADIUS) и ряд других. Соответственно, агент может обладать множествами морфизмов σ, δ, τ. Иначе – интеллектуальный агент может выполнять несколько процедур преобразования данных (далее – ППД) в зависимости от типов данных.

В общем виде ППД агента может быть представлена как: $\tau_i \left(D^{in}, D^{out}, S', T' \right)$, где: $D^{in} \subset D^{in}$ – подмножество входных данных ППД, $D^{out'} \subset D^{out}$ – подмножество выходных данных ППД, $S' \subset S$ – подмножество состояний агента реализующих ППД, $T' = t_1 - t_0$ – интервал времени реализации ППД.

Если рассматривать агента как ПАК, то согласно [11-14] можно утверждать, что:

- в процессе работы программ агента все обрабатываемые данные являются локальными (относительно поступающих и исходящих данных);
- работа программ агента по обработке данных и сами данные не должны рассматриваться отдельно от аппаратной части агента;
- состояние агента определяется текущим распределением аппаратных ресурсов агента между его исполняемыми процессами (программами).

Таким образом, состояния агента могут быть описаны тройкой $S = \langle H, P, D^l \rangle$, где: H – множество аппаратных ресурсов, P – множество программ (процессов), D^l – множество локальных данных.

Использование категорного подхода (см. рисунок 2) дает возможность описать взаимодействия между аппаратной и программной частями, а также локальными данными следующим образом: $\rho : H \times P \to P$ и $\gamma : H \times D^l \to D^l$. Что в свою очередь дает основание ввести отношение: $\omega = \rho^{\circ}\gamma$.

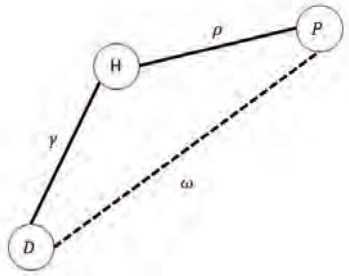

Рис.2. Морфизмы состояний

Диаграмма на рисунке 3 позволяет, с одной стороны, представить функционирование агента как последовательное выполнение следующих преобразований: $D^l = \omega\left(D^{in}\right)$ и $D^{out} = \omega\left(D^l\right)$, а с другой, делает необходимым учет отношений типа $P \times P \to P$, что в свою очередь позволяет в самом общем случае также рассматривать их, как реализацию одного или нескольких преобразований вида $D_i^l = \omega\left(D_j^l\right), i \neq j$.

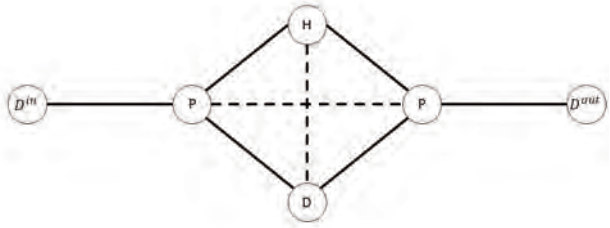

Рис.3. Диаграмма функционирования

3 В качестве примера можно привести следующие два отчета: «Déjà vu-lnerability. A Year in Review of 0-days Exploited In-The-Wild in 2020» (https://googleprojectzero.blogspot.com/2021/02/deja-vu-lnerability.html) и «Отчет о развитии киберугроз за I квартал 2021 г» (https://securelist.ru/it-threat-evolution-q1-2021/101485/)

Таким образом, процесс функционирования агента может быть представлен как последовательность шагов по преобразованию данных вида $D_i^* = \omega\left(D_j^*\right), i \neq j$. Следовательно, ППД агента можно определить как $\tau = \left\{\omega_1\left(D^*\right),\ldots,\omega_n\left(D^*\right)\right\}$. При этом время выполнения ППД агента может рассматриваться как последовательность шагов по преобразованию данных ω_i.

С точки зрения ИБ, каждый шаг реализации ППД агента, связанный с ошибками при преобразовании данных, может, как минимум – создавать условия для возникновения или реализации угроз, а как максимум – являться источником угрозы безопасности информации. Это дает основания говорить о безошибочности или корректности реализации функции $\omega_i\left(D^*\right)$.

Каждый шаг преобразования данных $\omega_i\left(D^*\right)$ в рамках выполнения ППД агента зависит от:

- корректности входных данных на каждом этапе преобразований;
- качества исходного кода алгоритмов преобразования входных данных;
- корректности использования предоставленных ресурсов (параметров работоспособности);
- качества обработки ошибок и исключений возникающих при преобразованиях данных.

Обозначим $D = D^{in} \cup D^{out} \cup D^{l}$. Фоннеймановская архитектура современных вычислительных систем позволяет программный код, а также сообщения об ошибках и исключениях рассматривать как данные D. Тогда можно ввести следующие функции:

$p(D)$ – соответствия качества кода заданным критериям;

$k(D)$ – соответствия использования ресурсов требуемым или допустимым значениям;

$e(D)$ – возможности обрабатывать ошибки и исключения;

$w(D)$ – соответствия данных требуемым составу и диапазону допустимых значений.

Таким образом, обобщенная корректность шага преобразования данных может быть представлена как: $u(\omega) = \mathfrak{S}(p,k,e,w)$.

Представим, что для каждого шага ППД $\omega_i(D)$ существует его идеальная реализация, гарантирующая корректное выполнение $u^0(\omega) = \mathfrak{S}\left(p^0,k^0,e^0,w^0\right)$. Тогда разница между идеальной и актуальной реализацией $v(h) = u^0(\omega) - u(\omega)$ представляет собой неопределенность (с точки зрения ИБ) реализации шага ППД – как потенциальной возможности ошибочного завершения или не выполнения ППД агента. Следует подчеркнуть, что функция $v(h)$ – функция неопределенности (далее – ФН) – характеризует потенциальную возможность нарушения конфиденциальности, целостности или доступности информации в процессе преобразования данных агентом. Как в случае наличия проявляющихся или не проявляющихся ошибок выполнения, так и в случае целенаправленного воздействия с целью вызова ошибок исполнения. Соответственно, ППД может быть представлена как набор ФН: $\tau = \left\{v_1(h),\ldots,v_n(h)\right\}$.

Наличие так называемых «ошибок нулевого дня» позволяет утверждать, что функция $u^0(\omega)$ практически не вычислима для агентов, построенных на базе современных ПАК. Существующие методы тестирования и проверки корректности программной реализации ППД позволяют оценивать функцию $u(\omega)$ только с позиций вероятности отсутствия ошибок или слабостей при преобразованиях данных. Таким образом, с точки зрения ИБ, при реализации отдельных шагов ППД агента можно оценивать ФН относительно и на основе некоторых качественных характеристик (сравнение одного с другим). Область определения ФН дает основания для подбора таких качественных характеристик.

Операции ФН

Предложенные ранее функции: соответствия качества кода заданным критериям, соответствия использования ресурсов требуемым или допустимым значениям, возможности обрабатывать ошибки и исключения, соответствия данных требуемым составу и диапазону допустимых значений – позволяют рассматривать их как область определения ФН, поскольку, $v(h) = u^0(\omega) - u(\omega) = \mathfrak{S}\left(p^0,k^0,e^0,w^0\right) - \mathfrak{S}(p,k,e,w)$.

Если положить $f = \{p,k,e,w\}$, $f^0 = \left\{p^0,k^0,e^0,w^0\right\}$ и $op_i = f_i^0 - f_i$, то можно говорить об отображении $h_i \rightarrow op_i$. В дальнейшем $op = f^0 - f$ будем называть операциями (с точки зрения ИБ).

Тогда ФН может быть представлена как набор операций: $v(h) = \{op_i,\ldots op_n\}$.

Структура и принципы построения современных вычислительных систем для полного описания отношения $\omega_i\left(D^*\right) \rightarrow v(h)$ требуют расширить перечень операций op, как минимум, с учетом следующих, реализуемых в программах преобразования данных функций:

- обращения (адресации, модификации) к данным;
- управления (запуск, останов) процессами или потоками;
- разграничения полномочий (правил доступа, идентификации, аутентификация);
- взаимодействия (связей, соединений, совместного использования ресурсов).

Полный список операций, описывающих с точки зрения ИБ реализацию интеллектуальным агентом ППД можно определить на основе представлений Top-25, 7 Pernicious Kingdoms, Research Concepts и Software Fault Pattern Clusters базы данных cwe.mitre.org.

В пользу использования базы данных CWE для определения перечня операций можно привести следующие аргументы:

1) записи этой базы представляют собой описание качественных характеристик (прежде всего ПО), с точки зрения потенциальных возможностей для нарушения конфиденциальности, доступности и целостности информации в процессе реализации интеллектуальным агентом той или иной ППД;

2) записи CWE поддерживаются в актуальном состоянии и позволяют оперативно учитывать изменения, связанные с развитием информационных технологий, а также техник и процедур выполнения атак. Кроме того, предлагаемый подход обеспечивает единую платформу для оценки ФН, практически не зависящую от типа агента (микросервисы, киберфизические устройства или функциональные компоненты

объектов критической информационной инфраструктуры – КИИ);

3) иерархическая (древовидная) структура базы данных CWE позволяет определять наборы операций, описывающих ФН на каждом шаге ППД с различной детализацией в зависимости от целей анализа и наличия исходных данных, позволяющих описать шаги ППД.

Для иллюстрации сказанного приведем следующий пример. Отличительной особенностью интеллектуального агента сложной сети можно считать его способность обмениваться в автоматическом режиме данными с другими агентами. В настоящее время типовым шаблоном реализации таких способностей агента стало использование для приема-передачи данных фиксированного API с доступом по протоколу HTTP, что можно представить как сочетание вэб-сервера и сервера приложений: вэб-сервер (apache) обеспечивает взаимодействие с пользователем и для обработки запросов вызывает модули сервера приложений (php-программы). Максимально упрощая, но без потери общности, можно выделить следующие шаги обработки запроса пользователя:

- выделение памяти под запрос и ответ;
- прием пакетов и выделение тела запроса (открытие и обработка сокета);
- авторизация (запроса с помощью JWT, например);
- запуск процесса обработки (php-модуль);
- обмен данными с запущенным процессом;
- формирование ответа и его передача (открытие и обработка сокета).

При этом следует отметить, что запуск процесса обработки как минимум состоит из шагов, связанных с:

- перераспределением ресурсов агента (прежде всего памяти и времени процессора);
- изменением в таблицах идентификации и аутентификации процессов;
- перераспределением прав доступа процессов;
- формированием канала взаимодействия процессов (общая память, именованные каналы, сокеты и т. п.).

Каждый из перечисленных шагов может быть представлен определенной операцией, сформированной на основе базы данных cwe.mitre.org. Например, такими как:

- CWE-20: Improper Input Validation;
- CWE-787: Out-of-bounds Write;
- CWE-1038: Insecure Automated Optimizations;
- CWE-436: Interpretation Conflict;
- CWE-284: Improper Access Control.

Перечисленные примеры операций могут быть родительскими узлами дерева CWE для более детально описанных операций. Например, CWE-787 является родительским узлом для операции CWE-119: Improper Restriction of Operations within the Bounds of a Memory Buffer, а узел CWE-284 – для CWE-287: Improper Authentication.

Уровень детализации операций должен определяться целями анализа и возможностями по идентификации операций. Например, при наличии исходных кодов агента целесообразно использовать более детальный подход к определению операций op . Который по очевидным причинам мало применим к проприетарным киберфизическим устройствам с ограниченными возможностями по регистрации и анализу действий и ошибок работы.

Говоря об уровне детализации операций, следует особо отметить, что большая часть базы CWE содержит перечень типовых уязвимостей – CVE. С точки зрения операций, сопутствующие уязвимости можно рассматривать как мутации операции, вызванные конкретными особенностями реализации отдельных компонент агента.

С учетом современных паттернов проектирования, предусматривающих широкое повторное использование кода (а значит и вызовы и взаимодействия программ, библиотек и процессов) можно предположить, что на экспертном уровне возможно формирование эталонного перечня операций позволяющего описывать отображение $\omega_i\left(D^*\right)\rightarrow v\left(h\right)$, то есть реализацию определенного шага обработки данных агентом с точки зрения ФН как наличия потенциальных слабостей влияющих на обработку.

Применение операций в интересах ИБ влечет за собой учет определенных особенностей их использования:

1) записи в базе данных CWE, содержат оценки влияния операции на конфиденциальность, целостность и доступность информации. Но этого недостаточно. Помимо стандартного соотнесения операций с категорией (грифом) обрабатываемых данных в настоящее время возникает еще один аспект оценки влияния операции. Микросервисы, контейнеризация приложений (и FaaS) и в целом применение архитектуры «инфраструктура как код» делают критичными такие данные, как команды (и скрипты) управления, конфигурации, эталонные образцы и обновления версий агентов. В целом данные, используемые в задачах оркестрации (развертывания и управления составом и режимами функционирования) агентов сложных сетей, особенно для объектов КИИ, могут иметь большее значение, чем непосредственно обрабатываемые агентами данные. Также для оценки операций (как возможных последствий выполняемых преобразований данных) целесообразно учитывать непосредственные действия с данными: создание новых, удаление или модификация существующих данных. То есть, для каждой операции необходимо ввести префикс – pref (действия) и суффикс – suf (тип или категория данных): $op^{'} = pref, op, suf$;

2) некоторые операции могут иметь двойственный характер при их оценке с точки зрения ИБ. Например, операции, связанные с авторизацией и управлением памятью. Несложно представить ситуацию, когда отсутствие действий по идентификации, аутентификации или авторизации хуже, чем реализация подобных действий с уязвимостями. Аналогично, статически собранная со всеми необходимыми библиотеками (компонентами) программа гораздо лучше своего динамического варианта именно из-за отсутствия необходимости управления распределением памяти в процессе выполнения. Таким образом, набор операций, определенный по базе данных CWE, должен быть расширен за счет введения «обратных» операций, а также учета типа обрабатываемых данных и выполняемых с ними действий.

DOI:10.21681/2311-3456-2021-6-26-35

Структура записей базы данных CWE содержит набор качественных параметров, позволяющих дать оценку операции с точки зрения влияния выполняемых преобразований данных на конфиденциальность, целостность и доступность информации. К таким параметрам можно, в частности, отнести зависимость от платформы (языка программирования, операционной системы, архитектуры и т. п.), общие последствия вследствие нарушения конфиденциальности, целостности и доступности, вероятность наличия готового эксплойта для реализации слабости, а также связанные с CWE уязвимости (CVE) и шаблоны атак (CAPEC). Кроме того, по характеристикам связанных уязвимостей можно определить возможность и условия удаленной эксплуатации слабости и уровень необходимых для этого привилегий. То есть, значимость операции с точки зрения ИБ может быть представлена вектором качественных параметров. Применение бинарной свертки [15] к элементам вектора позволяет получить интегральное значение значимости операции или веса операции θ. Целесообразно матрицы сверки строить таким образом, чтобы операции, имеющие более значимое влияние на ИБ, имели больший вес. В результате получится набор пар $\left(op^{'},\theta\right)$, который может быть упорядочен по весу θ операций $op^{'}$.

Приведенный ранее пример, а также наличие связей типа «CanFollow, PeerOf and CanAlsoBe» между отдельными записями в базе данных CWE (цепочки) указывают на возможность повторения отдельных операций или их комбинаций при описании ППД и целевой функции агента как последовательности операций. Для обеспечения возможности повторения операций в процессе описания ППД целесообразно ввести «нулевую операцию» op^{0}, которая заведомо не влияет на ИБ при выполнении определенных преобразований данных. В качестве образа такой операции можно привести работу статически скомпилированной программы не требующей изменения среды исполнения и любого взаимодействия с окружением в процессе своей работы (например, программа «из времен MSDOS 3.3»). Важно отметить, что при описании ППД, как последовательности операций, вес операции не зависит от места операции в этой последовательности.

Введем функцию описания операций в соответствии с ее весом $C\left(op^{'},\theta\right)=a$. Положим, что для обеспечения машинной обработки выход функции должен формироваться на основе универсального множества $\{0,1\}$. Поскольку возможно только ранжирование весов операций на основании экспертной оценки, то в качестве функции $C\left(op^{'},\theta\right)$ целесообразно взять функцию универсального кодирования символов. Наиболее предпочтительным представляется код Фибоначчи, в частности потому, что он допускает использование комбинации 11 в качестве разделительного символа. Результатом работы в этом случае будет двоичная последовательность a, которая обладает следующими свойствами:

- каждая последовательность будет уникальна для конкретной операции в рамках определенного списка операций;
- длина последовательности будет отображать значимость или вес операции $l\left(a\right)\to\theta$.

Тогда можно говорить, что результаты работы функции $C\left(op^{'},\theta\right)=a$ образуют алфавит $\mathrm{A}^{*}=\{a_1,\ldots,a_n\}$ из элементов которого формируются слова, описывающие, как отдельные шаги, ППД $\omega_i\left(D^{*}\right)$. Соответственно, наборы слов дадут описание с точки зрения ИБ ППД агента в целом.

Поскольку получение любого элемента алфавита A^{*} происходит по одним и тем же правилам (бинарная свертка и универсальное кодирование) и на основе единых исходных данных (база данных CWE), то можно говорить об универсальности этого алфавита при описании ФН для различных ППД различных агентов. С другой стороны, наименование операции $op^{'}$ будем обозначать, например, символами латинского алфавита по мере определения этих операций, что можно представить, как алфавит $B^{*}=\{b_1,\ldots,b_n\}$. Очевидно, что эти алфавиты тождественны с точки зрения описания одного и того же объекта.

Пусть у нас есть шаг ППД $\omega_i\left(D^{*}\right)$, описываемый тремя операциями: $\omega=\{b_1,b_2,b_3\}$. В силу смыслового тождества алфавитов A^{*} и B^{*} (они описывают одни и те же операции) можно также записать $\omega=\{a_1,a_2,a_3\}$. Иными словами, существуют все основания считать, что элемент алфавита B^{*} является компактным (сжатым) описанием тождественного элемента алфавита A^{*} поскольку для одной и той же операции длина ее описания в алфавите B^{*} (один символ латинского алфавита) не больше, чем длина ее описания в алфавите A^{*} (несколько двоичных символов).

Следует еще раз отметить, что формирование списка операций должно выполняться экспертным сообществом и конечной целью этой деятельности является определение алфавитов A^{*} и B^{*}, позволяющих описать ФН агентов сложной сети единообразным образом. Это, в свою очередь, позволит проводить сравнение агентов и дальнейшее оценивание сложной сети с точки зрения ИБ.

Оценка ФН

Как было показано, описание каждого шага ППД с точки зрения ИБ представляет собой ФН, которая может быть представлена набором операций. В свою очередь каждая операция может быть представлена символом алфавитов A^{*} и B^{*}, при этом запись ФН символами алфавита B^{*} представляет собой сжатую запись той же ФН символами алфавита A^{*}. Длина двоичной последовательности символов алфавита A^{*} отображает определенные возможные негативные последствия преобразования данных в виде веса операции, сопоставленной этому символу.

При этом для ФН и входящих в ее состав операций не могут быть определены (вычислены) количественные (численные) показатели эталонной и текущей реализации с точки зрения ИБ. Записи CWE дают качественную оценку ФН: качественную оценку разницы между эталонной и текущей реализацией: $u^{0}-u$. То есть, можно говорить об отсутствии числовой меры для вычисления u^{0} и u для формирования количественной оценки ФН.

Иначе – отсутствует возможность оценить объект x при известной информации об объекте y: $K(x|y)$. То есть, оцениваемый объект должен оцениваться «сам по себе», на основании своих внутренних (собственных) качеств. При этом вес каждой операции определяет тяжесть потенциальных негативных последствий с точки зрения ИБ при реализации того или иного шага ППД, который состоит из определенного числа операций.

Подход к решению подобного рода задач предложил в свое время А.Н. Колмогоров [16], показавший, что для оценки объекта «самого по себе» можно использовать понятие сложности объекта. С этой точки зрения можно говорить об оценке различных ППД и соответствующих им ФН, как об оценке их сложности. Сложность ФН, в этом случае, может трактоваться следующим образом: чем больше операций составляют ФН и чем больше вес каждой из операций, тем сложнее ФН.

Колмогоровская сложность зависит не только от состава объекта (набора операций), но и от внутренней структуры объекта (например, повтора операций). Для оценки сложности объекта А.Н. Колмогоров предлагал использовать длину программы, определяющий данный объект. При этом было доказано [см., например, 16, основная теорема], что выбор языка программирования для описания объекта не является принципиальным. Дальнейшее развитие теории Колмогоровской сложности и примеры ее применения [17-20] позволяют в качестве сложности объекта рассматривать длину программы восстановления полной строки символов из ее сжатого варианта. То есть, если шаг ППД ω описывается набором операций в алфавите A^* как $\{a_1, a_2, a_3\}$ и строка $s = a_1 + a_2 + a_3$ представляет собой конкатенацию двоичных строк символов, то сложность ФН, дающая оценку этого шага ППД с точки зрения ИБ равна длине программы, формирующей эту строку $K(\omega) = d(s)$. Конкатенация допустима, поскольку экспертные оценки закодированы универсальным кодом, то есть различимы без использования дополнительных разделителей.

Таким образом, можно сказать, что оценка ФН, соответствующая потенциальным негативным последствиям, могущим возникнуть в ходе выполнения преобразования данных равна Колмогоровской сложности длины строки, состоящей из символов алфавита A^*:

$h = K(s)$, где $s = a_1 + a_i + a_k$, $a_i \in A^*$.

Величина $K(s)$ дает верхнюю оценку сложности с точностью до некоторой постоянной $\log N$, где N – длина строки. Это ограничение связано с тем, что для точной оценки $K(s)$ необходимо найти программу минимальной длины, обеспечивающую формирование строки s.

Для частичного снятия данного ограничения примем, что программа формирования строки s реализуется машиной Тьюринга (далее – МТ). При этом, поскольку коды операций, составляющих любую конкретную ППД известны (экспертным образом определены заранее), то для выполнения требования минимальности целесообразно для каждой ППД создавать отдельную МТ. Кроме того, как будет показано далее, такой подход дает дополнительные возможности в оценке агента с точки зрения ИБ. МТ на входе получает операции из состава ППД, представленные в алфавите B^*, а на выходе формирует строку этих же операций в алфавите A^*.

Пусть на вход МТ поступает три символа операций в алфавите B^* – a, b, c, каждый из которых имеет тождественные символы в алфавите A^* с длиной в битах, соответствующей тяжести последствий этих операций с точки зрения ИБ – A^*. При расчете длины программы МТ будем учитывать только команды чтения и записи символов без учета команд перемещения головки и оптимизации команд для представления битовых последовательностей. Результаты определения примерной длины программы МТ, то есть неопределенности ППД, для различных сочетаний символов входной строки (операций в составе ППД) приведены в таблице 1.

Таблица 1

Результаты определения примерной длины программы МТ

Входная строка	Команды МТ	Длина программы
abc	R, 2W, R, 3W, R, 4W	12
aab	R, 2W, 2W, R, 3W	9
aac	R, 2W, 2W, R, 4W	10
abb	R, 2W, R, 3W, 3W	10
bbc	R, 3W, 3W, R, 4W	12
acc	R, 2W, R, 4W, 4W	12
bcc	R, 3W, R, 4W, 4W	13
aaa	R, 2W, 2W, 2W	7
bbb	R, 3W, 3W, 3W	10
ccc	R, 4W, 4W, 4W	13

Можно полагать, что с увеличением длины входной строки, то есть более детальным определением операций ППД, эффективность применения предлагаемого подхода к оценке агента увеличится.

Применение МТ в качестве аналогов ППД при оценке целевой функции агента позволяет анализировать различные комбинации ППД, описывающие работу агента. Сюда прежде всего относятся случаи ветвления и параллельного исполнения ППД. Например, случай параллельного выполнения нескольких экземпляров ППД (одновременные запросы пользователей) можно представить схемой параллельного соединения соответствующего числа однотипных МТ. Или использовать логические операции типа $K(s) \wedge \ldots \wedge K(s) = \sum_{i=1}^{N} K(s)$, где N – число МТ. В [17], в частности, приведены такие логические операции с Колмогоровской сложностью как импликация и стрелка Пирса.

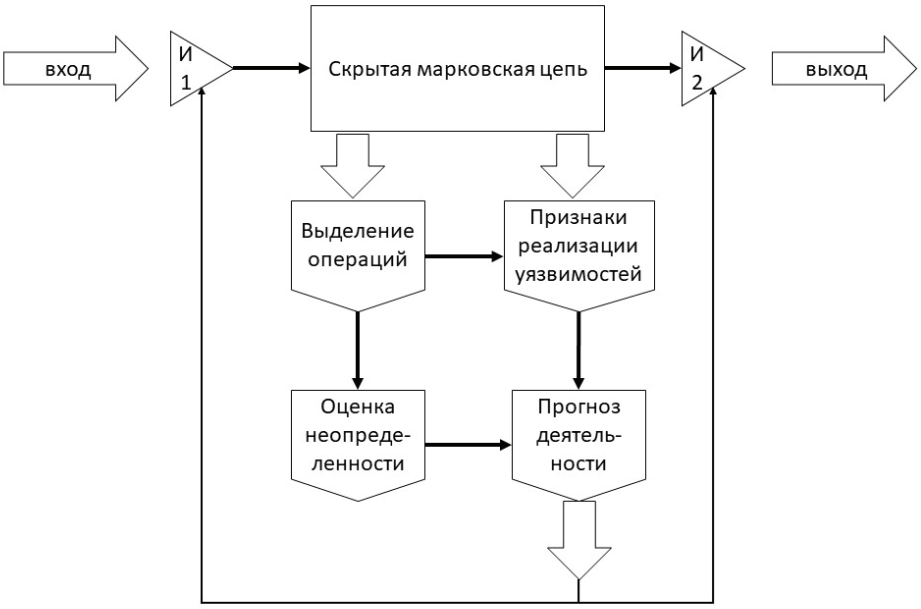

Рис. 4. Примерная схема цифрового ИБ-двойника агента

Кроме того, представление работы агента в виде схемы соединений МТ позволяет определять точки перемены категории данных или наличие веток с одновременной обработкой данных разных категорий.

Таким образом, учитывая изложенное выше, можно предположить, что использование Колмогоровской сложности и соответствующих МТ создают условия для формирования цифровых ИБ-двойников как агентов, так и отдельных подсистем или фрагментов сложных сетей.

В случае объектов КИИ, когда известен программно-аппаратный состав агентов на уровне исходных кодов, применение цифровых ИБ-двойников позволит оценивать состояние защиты информации объектов в режиме, близком к реальному времени. Примерная схема цифрового ИБ-двойника агента приведена на рис. 4.

Собственно, агент представлен в виде скрытой марковской цепи выходы которой представляют собой набор параметров необходимых для выделения операций и признаков реализации угроз. Выход модуля выделения операций используется для выделения признаков реализации угроз и оценки текущей неопределенности агента. В свою очередь эта оценка и признаки используются в качестве входов в модуле прогноза деятельности. Данный модуль обеспечивает управляющие воздействия на исполнительные устройства И1 и И2 которые имитируют реализацию функций защиты информации в агенте.

Заключение

В статье предложена модель агента сложной сети, которая предполагает представление его работы в виде последовательности операций во времени. Набор операций образует алфавит, из символов которого формируется слова и предложения описывающие процедуры преобразования данных и целевые функции агента. Предложены методы оценки на основе бинарной свертки и Колмогоровской сложности с точки зрения ИБ как операций, так и процедур преобразования данных и целевых функций на основе операций.

Предлагаемые подходы к оценке агента могут быть использованы в качестве вычислительных методов системы стохастического имитационного моделирования сложных многоагентных сетей.

Использование Колмогоровской сложности и соответствующих МТ создают условия для формирования цифровых ИБ-двойников как агентов, так и отдельных подсистем или фрагментов сложных сетей двойников в системах оценки и моделирования объектов КИИ.

Предлагаемые методы оценки агента позволяют расширить сферу применения имеющихся способов анализа и прогнозирования состояния ИБ (например, на основе деревьев атак [21-23]) за счет возможности замены оценок агентов на основе известных уязвимостей на оценки на основе Колмогоровской сложности.

В работе [24] показано, что время является ключевым ресурсом при проведении атак. Применение комбинационных схем из различных МТ, соответствующих определенным операциям, позволяет использовать временной анализ в различных системах анализа ИБ, если рассматривать время как последовательность операций при реализации ППД агента.

Литература

1. Калашников А. О. Инфраструктура как код: формируется новая реальность информационной безопасности / А. О. Калашников, К. А. Бугайский // Информация и безопасность. – 2019. – Т. 22. – № 4. – С. 495-506.
2. Герасимов А. Ю. Формальная модель обнаружения программных ошибок с помощью символьного исполнения программ / А. Ю. Герасимов, Д. О. Куц, А. А. Новиков // Труды Института системного программирования РАН. – 2019. – Т. 31. – № 6. – С. 21-32. – DOI 10.15514/ISPRAS-2019-31(6)-2.
3. Колбина А. О. Применение кибернетических методов Чёрного и белого ящиков в разработке и тестировании программ / А. О. Колбина, В. А. Баранов, П. В. Пересунько // Новая наука: От идеи к результату. – 2016. – № 9-1. – С. 25-27.
4. Липаев В. В. Тестирование компонентов и комплексов программ / В. В. Липаев. – М.|Берлин : Директ-Медиа, 2015. – 528 с. – ISBN 9785447538651.
5. Макаренко С. И. Анализ стандартов и методик тестирования на проникновение / С. И. Макаренко, Г. Е. Смирнов // Системы управления, связи и безопасности. – 2020. – № 4. – С. 44-72. – DOI 10.24411/2410-9916-2020-10402.
6. Барабанов А. В. Актуальные вопросы выявления уязвимостей и недекларированных возможностей в программном обеспечении / А. В. Барабанов, А. С. Марков, В. Л. Цирлов // Системы высокой доступности. – 2018. – Т. 14. – № 3. – С. 12-17. – DOI 10.18127/j20729472-201803-03.
7. Исследование уязвимостей программного обеспечения : Учебное издание / А. В. Барабанов, А. С. Марков, В. Л. Цирлов, Ю. В. Рауткин. – Москва : Федеральное бюджетное учреждение Научный центр правовой информации при Министерстве юстиции Российской Федерации, 2018. – 76 с. – ISBN 9785901167465.
8. Статистика выявления уязвимостей программного обеспечения при проведении сертификационных испытаний / А. В. Барабанов, А. С. Марков, А. А. Фадин, В. Л. Цирлов // Вопросы кибербезопасности. – 2017. – № 2(20). – С. 2-8. – DOI 10.21581/2311-3456-2017-2-2-8.
9. Барабанов А. В. О систематике информационной безопасности цепей поставки программного обеспечения / А. В. Барабанов, А. С. Марков, В. Л. Цирлов // Безопасность информационных технологий. – 2019. – Т. 26. – № 3. – С. 68-79. – DOI 10.26583/bit.2019.3.06.
10. Горбатова Е. В. Тенденции и перспективы развития наиболее опасных киберугроз / Е. В. Горбатова // Модернизация и устойчивое социально-экономическое развитие России и ее регионов в XXI веке сквозь призму роста производительности труда : Сборник статей II Всероссийской научно-практической конференции, Ступино, 19–20 февраля 2020 года. – Ступино: Московский финансово-юридический университет МФЮА, 2020. – С. 379-387.
11. Бойченко А. В. Основы открытых информационных систем. Учебное пособие / А. В. Бойченко, В. К. Кондратьев, Е. Н. Филинов. – Москва : Евразийский открытый институт, Московский государственный университет экономики, статистики и информатики, 2004. – 160 с. – ISBN 5776402840.
12. Вяткин А. И. Операционные системы, среды и оболочки : Учебное пособие / А. И. Вяткин. – Тюмень : Тюменский государственный университет, 2011. – 272 с. – ISBN 9785400004773.
13. Акинин М. В. Системное программирование в Linux. Часть 1. Управление процессами : Учебное пособие / М. В. Акинин, Н. В. Акинина, С. В. Засорин. – Москва : Общество с ограниченной ответственностью Издательство «КУРС», 2019. – 192 с. – (ИНФОРМАТИКА). – ISBN 9785907064805.
14. Таненбаум Э. Н. Операционные системы : Разработка и реализация (+CD). Классика CS / Э. Н. Таненбаум, А. Л. Вудхалл. – 3-е изд. – Санкт-Петербург : Питер, 2007. – 704 с. – ISBN 9785469014034.
15. Калашников А. О. Модель управления информационной безопасностью критической информационной инфраструктуры на основе выявления аномальных состояний (часть 1) / А. О. Калашников, Е. В. Аникина // Информация и безопасность. – 2018. – Т. 21. – № 2. – С. 145-154.
16. Колмогоров А. Н. Три подхода к определению понятия «количество информации» / А.Н. Колмогоров // Проблемы передачи информации. – 1965. – Т. 1. – № 1. – С.3-11.
17. Верещагин Н. К. Колмогоровская сложность и алгоритмическая случайность / Н. К. Верещагин, В. А. Успенский, А. Шень. – Москва : МЦНМО, 2013. – 575 с. – ISBN 9785443902128.
18. Печников А. А. Об использовании колмогоровской сложности для исследования схожести изображений / А. А. Печников // Международный научно-исследовательский журнал. – 2018. – № 5(71). – С. 59-61. – DOI 10.23670/IRJ.2018.71.024.
19. Губин А. Н. Информационная эффективность интеллектуальных систем поддержки принятия решений / А. Н. Губин, В. Л. Литвинов, Ф. В. Филиппов // Информационные системы и технологии в моделировании и управлении : Сборник трудов V Международной научно-практической конференции, Ялта, 20–22 мая 2020 года / Отв. редактор К.А. Маковейчук. – Ялта: Общество с ограниченной ответственностью «Издательство Типография «Ариал», 2020. – С. 19-23.
20. Лукьянова О. А. Эгоистичный искусственный интеллект / О. А. Лукьянова, О. Ю. Никитин // Cloud of Science. – 2019. – Т. 6. – № 3. – С. 462-474.
21. Дойникова Е. В. Анализ защищенности автоматизированных систем с учетом социо-инженерных атак / Е. В. Дойникова, И. В. Котенко, М. В. Степашкин // Проблемы информационной безопасности. Компьютерные системы. – 2011. – № 3. – С. 40-57.
22. Чечулин А. А. Методика оперативного построения, модификации и анализа деревьев атак / А. А. Чечулин // Труды СПИИРАН. – 2013. – № 3(26). – С. 40-53.

23. Чечулин А. А. Обработка событий безопасности в условиях реального времени с использованием подхода, основанного на анализе деревьев атак / А. А. Чечулин, И. В. Котенко // Проблемы информационной безопасности. Компьютерные системы. – 2014. – № 3. – С. 56-59.
24. Калашников А. О. Модели количественного оценивания компьютерных атак / А. О. Калашников, К. А. Бугайский, Е. В. Аникина // Информация и безопасность. – 2019. – Т. 22. – № 4. – С. 517-528.

A MODEL FOR QUANTIFYING THE AGENT OF A COMPLEX NETWORK IN CONDITIONS OF INCOMPLETE AWARENESS

Kalashnikov A.O.[4], Bugajskij K.A.[5]

Purpose of the article: *development of a mechanism for quantitative evaluation of elements of complex information systems in conditions of insufficient information about the presence of vulnerabilities.*

Research method: *mathematical modeling of uncertainty estimation based on binary convolution and Kolmogorov complexity. Data banks on vulnerabilities and weaknesses are used as initial data for modeling.*

The result: *it is shown that the operation of an element of a complex network can be represented by data transformation procedures, which consist of a sequence of operations in time, described by weaknesses and related vulnerabilities. Each operation can be evaluated at a qualitative level in terms of the severity of the consequences in the event of the implementation of potential weaknesses. The use of binary convolution and universal coding makes it possible to translate qualitative estimates into a binary sequence – a word in the alphabet {0,1}. The sequence of such words — as the uncertainty function — describes the possible negative consequences of implementing data transformation procedures due to the presence of weaknesses in an element of a complex system. It is proposed to use the Kolmogorov complexity to quantify the uncertainty function. The use of a Turing machine for calculating the uncertainty function provides a universal mechanism for evaluating elements of complex information systems from the point of view of information security, regardless of their software and hardware implementation.*

Keywords: *vulnerability, binary convolution, Kolmogorov complexity, Turing machine, universal coding, information security model, evaluating complex systems.*

References

1. Kalashnikov, A. O. Infrastruktura kak kod: formiruetsya novaya real`nost` informacionnoj bezopasnosti / A. O. Kalashnikov, K. A. Bugajskij // Informaciya i bezopasnost`. – 2019. – Tom 22. – № 4. – pp. 495-506.
2. Gerasimov, A. Yu. Formal`naya model` obnaruzheniya programmny`x oshibok s pomoshh`yu simvol`nogo ispolneniya programm / A. Yu. Gerasimov, D. O. Kucz, A. A. Novikov // Trudy` Instituta sistemnogo programmirovaniya RAN. – 2019. – Tom 31. – № 6. – pp. 21-32. – DOI 10.15514/ISPRAS-2019-31(6)-2.
3. Kolbina, A. O. Primenenie kiberneticheskix metodov Chyornogo i belogo yashhikov v razrabotke i testirovanii programm / A. O. Kolbina, V. A. Baranov, P. V. Peresun`ko // Novaya nauka: Ot idei k rezul`tatu. – 2016. – № 9-1. – pp. 25-27.
4. Lipaev, V. V. Testirovanie komponentov i kompleksov programm / V. V. Lipaev. – M.|Berlin : Direkt-Media, 2015. – P. 528 – ISBN 9785447538651.
5. Makarenko, S. I. Analiz standartov i metodik testirovaniya na proniknovenie / S. I. Makarenko, G. E. Smirnov // Sistemy` upravleniya, svyazi i bezopasnosti. – 2020. – № 4. – pp. 44-72. – DOI 10.24411/2410-9916-2020-10402.
6. Barabanov, A. V. Aktual`ny`e voprosy` vy`yavleniya uyazvimostej i nedeklarirovanny`x vozmozhnostej v programmnom obespechenii / A. V. Barabanov, A. S. Markov, V. L. Cirlov // Sistemy` vy`sokoj dostupnosti. – 2018. – Tom 14. – № 3. – pp. 12-17. – DOI 10.18127/j20729472-201803-03.
7. Issledovanie uyazvimostej programmnogo obespecheniya : Uchebnoe izdanie / A. V. Barabanov, A. S. Markov, V. L. Cirlov, Yu. V. Rautkin. – Moskva : Federal`noe byudzhetnoe uchrezhdenie Nauchny`j centr pravovoj informacii pri Ministerstve yusticii Rossijskoj Federacii, 2018. – P. 76 – ISBN 9785901167465.
8. Statistika vy`yavleniya uyazvimostej programmnogo obespecheniya pri provedenii sertifikacionny`x ispy`tanij / A. V. Barabanov, A. S. Markov, A. A. Fadin, V. L. Cirlov // Voprosy` kiberbezopasnosti. – 2017. – № 2(20). – pp. 2-8. – DOI 10.21581/2311-3456-2017-2-2-8.
9. Barabanov, A. V. O sistematike informacionnoj bezopasnosti cepej postavki programmnogo obespecheniya / A. V. Barabanov, A. S. Markov, V. L. Cirlov // Bezopasnost` informacionny`x texnologij. – 2019. – Tom 26. – № 3. – pp. 68-79. – DOI 10.26583/bit.2019.3.06.

4 Andrey Kalashnikov, Dr.Sc., Chief Scientist of the Laboratory «Complex networks» Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russia. E-mail: aokalash@ipu.ru

5 Konstantin Bugajskij, Junior Researcher of the Laboratory «Complex networks» Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russia. E mail: kabuga@ipu.ru

10. Gorbatova, E. V. Tendencii i perspektivy` razvitiya naibolee opasny`x kiberugroz / E. V. Gorbatova // Modernizaciya i ustojchivoe social`no-e`konomicheskoe razvitie Rossii i ee regionov v XXI veke skvoz` prizmu rosta proizvoditel`nosti truda : Sbornik statej II Vserossijskoj nauchno-prakticheskoj konferencii, Stupino, 19–20 fevralya 2020 goda. – Stupino: Moskovskij finansovo-yuridicheskij universitet MFYuA, 2020. – pp. 379-387.
11. Bojchenko, A. V. Osnovy` otkry`ty`x informacionny`x sistem. Uchebnoe posobie / A. V. Bojchenko, V. K. Kondrat`ev, E. N. Filinov. – Moskva : Evrazijskij otkry`ty`j institut, Moskovskij gosudarstvenny`j universitet e`konomiki, statistiki i informatiki, 2004. – P. 160. – ISBN 5776402840.
12. Vyatkin, A. I. Operacionny`e sistemy`, sredy` i obolochki : Uchebnoe posobie / A. I. Vyatkin. – Tyumen` : Tyumenskij gosudarstvenny`j universitet, 2011. – P. 272. – ISBN 9785400004773.
13. Akinin, M. V. Sistemnoe programmirovanie v Linux. Chast` 1. Upravlenie processami : Uchebnoe posobie / M. V. Akinin, N. V. Akinina, S. V. Zasorin. – Moskva : Obshhestvo s ogranichennoj otvetstvennost`yu Izdatel`stvo «KURS», 2019. – P. 192. – (INFORMATIKA). – ISBN 9785907064805.
14. Tanenbaum, E`. N. Operacionny`e sistemy` : Razrabotka i realizaciya (+SD). Klassika CS / E`. N. Tanenbaum, A. L. Vudxall. – 3-e izd. – Sankt-Peterburg : Piter, 2007. – P. 704. – ISBN 9785469014034.
15. Kalashnikov, A. O. Model` upravleniya informacionnoj bezopasnost`yu kriticheskoj informacionnoj infrastruktury` na osnove vy`yavleniya anomal`ny`x sostoyanij (chast` 1) / A. O. Kalashnikov, E. V. Anikina // Informaciya i bezopasnost`. – 2018. – Tom 21. – № 2. – pp. 145-154.
16. Kolmogorov A. N. Tri podxoda k opredeleniyu ponyatiya «kolichestvo informacii» / A.N. Kolmogorov // Problemy` peredachi informacii. – 1965. – Tom 1. – № 1. – pp.3-11.
17. Vereshhagin, N. K. Kolmogorovskaya slozhnost` i algoritmicheskaya sluchajnost` / N. K. Vereshhagin, V. A. Uspenskij, A. Shen`. – Moskva : MCzNMO, 2013. – P. 575. – ISBN 9785443902128.
18. Pechnikov, A. A. Ob ispol`zovanii kolmogorovskoj slozhnosti dlya issledovaniya sxozhesti izobrazhenij / A. A. Pechnikov // Mezhdunarodny`j nauchno-issledovatel`skij zhurnal. – 2018. – № 5(71). – pp. 59-61. – DOI 10.23670/IRJ.2018.71.024.
19. Gubin, A. N. Informacionnaya e`ffektivnost` intellektual`ny`x sistem podderzhki prinyatiya reshenij / A. N. Gubin, V. L. Litvinov, F. V. Filippov // Informacionny`e sistemy` i texnologii v modelirovanii i upravlenii : Sbornik trudov V Mezhdunarodnoj nauchno-prakticheskoj konferencii, Yalta, 20–22 maya 2020 goda / Otv. redaktor K.A. Makovejchuk. – Yalta: Obshhestvo s ogranichennoj otvetstvennost`yu «Izdatel`stvo Tipografiya «Arial», 2020. – pp. 19-23.
20. Luk`yanova, O. A. E`goistichny`j iskusstvenny`j intellekt / O. A. Luk`yanova, O. Yu. Nikitin // Cloud of Science. – 2019. – T. 6. – № 3. – pp. 462-474.
21. Dojnikova, E. V. Analiz zashhishhennosti avtomatizirovanny`x sistem s uchetom socio-inzhenerny`x atak / E. V. Dojnikova, I. V. Kotenko, M. V. Stepashkin // Problemy` informacionnoj bezopasnosti. Komp`yuterny`e sistemy`. – 2011. – № 3. – pp. 40-57.
22. Chechulin, A. A. Metodika operativnogo postroeniya, modifikacii i analiza derev`ev atak / A. A. Chechulin // Trudy` SPIIRAN. – 2013. – № 3(26). – pp. 40-53.
23. Chechulin, A. A. Obrabotka soby`tij bezopasnosti v usloviyax real`nogo vremeni s ispol`zovaniem podxoda, osnovannogo na analize derev`ev atak / A. A. Chechulin, I. V. Kotenko // Problemy` informacionnoj bezopasnosti. Komp`yuterny`e sistemy`. – 2014. – № 3. – pp. 56-59.
24. Kalashnikov, A. O. Modeli kolichestvennogo ocenivaniya komp`yuterny`x atak / A. O. Kalashnikov, K. A. Bugajskij, E. V. Anikina // Informaciya i bezopasnost`. – 2019. – Tom 22. – № 4. – pp. 517-528.