ОЦЕНИВАНИЕ ЗАЩИЩЕННОСТИ ИНФОРМАЦИОННЫХ СИСТЕМ НА ОСНОВЕ ГРАФОВОЙ МОДЕЛИ ЭКСПЛОЙТОВ

Федорченко Е.В.[1], Котенко И.В.[2], Федорченко А.В.[3], Новикова Е.С.[4], Саенко И.Б.[5]

Цель исследования: *автоматизация процессов выявления и оценивания эксплойтов, к которым уязвима информационная система, за счет определения их признаков на основе анализа исходного кода эксплойтов, связанных слабых мест и уязвимостей в целях их дальнейшего устранения и повышения защищенности информационных систем.*

Методы исследования: *статистический анализ исходных данных, семантическое и синтаксическое моделирование процесса выполнения исходного кода эксплойтов, методы классификации данных для оценивания эксплойтов на основе связанных слабых мест и уязвимостей.*

Полученные результаты: *предложена общая концепция динамического оценивания защищенности информационных систем в условиях неопределенности исходных данных, в рамках предложенной концепции выделены применяемые при оценивании защищенности данные, связи между ними и основные виды неопределенностей, связанных с использованием неизвестных ранее уязвимостей, слабых мест анализируемой системы или эксплойтов, в основе концепции лежат методы статического и динамического анализа эксплойтов в целях устранения выявленных неопределенностей; определены источники данных и исходные данные для экспериментов, проведен их статистический анализ; предложена методика устранения выделенных неопределенностей на основе классификации эксплойтов с использованием признаков связанных с ними уязвимостей; произведена экспериментальная оценка точности классификации эксплойтов, и выделены недостатки предложенной методики; для устранения выделенных недостатков разработана графовая модель эксплойтов и методика ее формирования; предложена методика классификации эксплойтов на основе признаков, сформированных с использованием разработанной модели и связанных слабых мест и уязвимостей. Полученные результаты могут использоваться в системах мониторинга и повышения защищенности информационных систем.*

Научная новизна: *предложенная общая концепция динамического оценивания защищенности информационных систем отличается от существующих выделенными видами неопределенности исходных данных и применением методики классификации эксплойтов для их устранения за счет обнаружения признаков реализации эксплойтов, причем в основе предложенной концепции лежит гипотеза о том, что неизвестные ранее эксплойты используют уже известные ранее фрагменты вредоносного программного кода; предложенная методика классификации эксплойтов отличается от известных методик как использованием известных признаков связанных уязвимостей, так и признаков, основанных на графовой модели эксплойтов; разработанная графовая модель эксплойтов является вариацией семантического графа, построена на основе графа потока управления и графа зависимостей вызовов функций и позволяет учесть как основной маршрут выполнения кода, так и функциональные зависимости между импортируемыми именами функций при формировании признаков выполнения эксплойтов.*

Вклад: *Федорченко Е.В. – разработка методики классификации эксплойтов на основе признаков, сформированных с использованием графовой модели исходного кода эксплойтов, и связанных слабых мест и уязвимостей; Котенко И.В. и Федорченко А.В. – анализ положения дел по представлению исходного кода эксплойтов*

1 Федорченко (Дойникова) Елена Владимировна, кандидат технических наук, старший научный сотрудник, ФГБУН «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» (СПб ФИЦ РАН), г. Санкт-Петербург, Россия. E-mail: doynikova@comsec.spb.ru

2 Котенко Игорь Витальевич, 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

3 Федорченко Андрей Владимирович, программист, ФГБУН «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» (СПб ФИЦ РАН), г. Санкт-Петербург, Россия. E-mail: fedorchenko@comsec.spb.ru

4 Новикова Евгения Сергеевна, кандидат технических наук, старший научный сотрудник, ФГБУН «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» (СПб ФИЦ РАН), г. Санкт-Петербург, Россия. E-mail: novikova@comsec.spb.ru

5 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

DOI:10.21681/2311-3456-2023-3-23-36

в целях динамического оценивания защищенности информационных систем, постановка задачи классификации эксплойтов, разработка подхода к получению классификационных признаков; Федорченко А.В. – сбор и предварительный анализ исходных данных; Новикова Е.С. – экспериментальное исследование предложенного подхода; Саенко И.Б. и Федорченко Е.В. – разработка концепции динамического оценивания защищенности информационных систем в условиях неопределенности исходных данных.

Ключевые слова: *слабое место, уязвимость, признаки, анализ данных, классификация данных, мониторинг защищенности.*

DOI:10.21681/2311-3456-2023-3-23-36

Введение

Одной из важнейших задач мониторинга информационной безопасности является автоматизация процесса выявления и оценивания слабых мест защищаемой системы, ее уязвимостей и эксплойтов. Решение данной задачи повышает эффективность работы систем проактивного выявления и противодействия кибератакам. В настоящее время существуют инструменты, способные обнаруживать известные уязвимости в анализируемой системе. К ним, в частности, относятся сканеры безопасности Nessus[6] и Nmap[7]. В основе их работы лежит использование открытых баз данных уязвимостей, таких как Common Vulnerabilities and Exposures (CVE)[8], National Vulnerabilities Database (NVD)[9] и др.

Другим практическим подходом к устранению уязвимостей в информационной системе является изменение конфигурации системы, политик безопасности и/или внедрение инструментов безопасности, выполняющих анализ системы на наличие известных слабых мест. Основным источником данных о слабых местах является база данных CWE[10]. Однако не существует баз данных, включающих информацию о прямых связях между программным и аппаратным обеспечением, представленных в формате CPE[11], и слабыми местами, представленными в формате CWE.

Задача определения слабых мест информационной системы и эксплойтов, к которым она уязвима, на основе используемого программного и аппаратного обеспечения может быть сформулирована как задача классификации уязвимостей по слабым местам и эксплойтам. Это позволяет косвенно связать программное и аппаратное обеспечение с их слабыми местами и потенциально применимыми эксплойтами через их уязвимости, а также оценить слабые места и эксплойты на основе оценок уязвимостей, которые они используют. Наиболее простым подходом к решению этой задачи является использование базы данных NVD, которая показывает связи между уязвимостями и слабыми местами анализируемой системы. Записи об уязвимостях в NVD имеют связи со слабыми местами в CWE. Однако существует ряд проблем, препятствующих автоматическому сопоставлению слабых мест и уязвимостей с использованием NVD и CWE. Например, для некоторых записей в NVD отсутствуют ссылки на слабые места. Другая проблема заключается в высоком уровне абстракции связанных экземпляров в CWE.

Для решения данных проблем предлагается использовать методы классификации данных, в которых учитываются признаки уязвимостей, определенные системой оценки уязвимостей CVSS (Common Vulnerability Scoring System) версии 2.0 [1] и 3.0 [2], в рамках методики выявления и оценивания слабых мест и эксплойтов, к которым уязвима анализируемая система. Предложенный способ подходит для случаев, когда в используемых базах данных определена связь эксплойта с уязвимостью или слабым местом. Оценка эффективности предложенной методики выполнялась с помощью базы данных эксплойтов EDB[12]. Для некоторых эксплойтов база EDB содержит данные о связанных с ними уязвимостях CVE, что позволяет использовать ее при тестировании данной методики.

Кроме того, предлагается методика определения слабых мест для эксплойтов, не имеющих информацию об ассоциированных с ними уязвимостях. Методика позволяет извлекать признаки на основе анализа исходного кода эксплойтов с помощью графовых моделей представления кода. Сформированные признаки служат исходными данными для задачи клас-

6 https://networkguru.ru/tenable-nessus-vulnerability-scanner/
7 https://nmap.org/
8 https://cve.mitre.org/
9 https://nvd.nist.gov/
10 https://cwe.mitre.org/index.html
11 https://cpe.mitre.org/
12 https://www.exploit-db.com/

сификации слабых мест и эксплойтов, и их последующего оценивания. Сопоставление сформированных признаков событиям, генерируемым информационной системой в процессе ее эксплуатации, позволит выявлять и оценивать неизвестные ранее эксплойты. Данная методика может быть использована для определения слабого места эксплойта, если запись в EDB не содержит ссылки на уязвимость CVE, для повышения точности классификации слабых мест на основе анализа признаков уязвимостей CVE, а также для выявления неизвестных ранее эксплойтов в рамках динамического оценивания защищенности.

Для изложения особенностей полученных решений остановимся вначале на рассмотрении известных работ, посвященных данной тематике.

Анализ известных работ

Базы данных CVE и NVD традиционно используются для оценки уязвимостей [3-5], генерации атак [6-7], управления рисками [8] и обнаружения уязвимостей [9]. Другим источником данных, часто используемым для определения политик безопасности, требований безопасности и тестирования защищенности информационных систем, является база данных о шаблонах атак CAPEC[13]. Например, в [5] разработана концепция оценки рисков протоколов Интернета Вещей, в основе которой лежит модифицированная система оценок уязвимостей, которая учитывает особенности инфраструктуры Интернета Вещей.

Признаки уязвимостей CVSS характеризуют программные уязвимости с учетом особенностей их практической эксплуатации [10]. Характеристики уязвимостей CVSS версии 2.0 и 3.0 разделены на три подгруппы: базовые, временные и контекстные признаки. В настоящей работе используются только базовые признаки. Временные признаки отражают изменение уязвимости во времени, а контекстные признаки связаны с требованиями безопасности системы, что делает их в меньшей степени связанными со слабыми местами.

Признаки уязвимостей, определенных в CVSS, принимают количественные и качественные значения, определяемые экспертным путем. Они могут быть использованы для определения слабых мест на основе анализа уязвимостей. Слабые места, как и уязвимости, различаются по типу доступа и по возможному ущербу от их эксплуатации. Перечень известных слабых мест представлен в CWE.

13 https://capec.mitre.org/

Существует ряд работ, посвященных классификации уязвимостей. В [11] предложена новая структура оценок уязвимостей, формируемая на основе методов машинного обучения, позволяющая определить уровень критичности выявленных уязвимостей по шкале CVSS в автоматизированном режиме. Предлагаемая методика также решает проблему совместимости оценок CVSS различных версий, путем построения ансамбля моделей машинного обучения.

Для представления исходного кода программ известны следующие графовые модели: абстрактное синтаксическое дерево (AST), абстрактный семантический граф (ASG), граф потока управления (CFG), граф программных зависимостей (PDG) и другие [12].

Дерево AST строится на основе анализа грамматики языка программирования и отражает граф вызовов функций. Граф ASG генерируется на основе синтаксического дерева с помощью семантических правил. Граф CFG представляет собой направленный граф, в котором каждый узел соответствует базовому блоку, а ребра соединяют узлы, которые могут быть выполнены последовательно [13]. Подобные представления программного кода полезны для выявления уязвимостей в программном коде. В частности, в [14] показано, что совместный анализ динамически перестраиваемого графа программных зависимостей и исходного графа программных зависимостей позволяет эффективно выявлять ошибки на переполнение буфера.

Таким образом, анализ известных работ подтверждает правомерность предложенного подхода, в котором совместно используются базы данных по безопасности и графовые модели представления исходного кода эксплойтов.

Концепция динамического оценивания защищенности информационных систем в условиях неопределенности исходных данных

Для оценивания защищенности информационной системы необходимо иметь информацию о ее конфигурации, слабых местах, уязвимостях, возможных атакующих воздействиях и эксплойтах. Можно выделить следующие этапы оценивания защищенности и последующего автоматизированного выбора мер безопасности (рис. 1):

1) обнаружение уязвимостей анализируемой системы и/или доступных эксплойтов;

2) определение слабых мест анализируемой системы;

3) определение киберугроз для анализируемой системы;

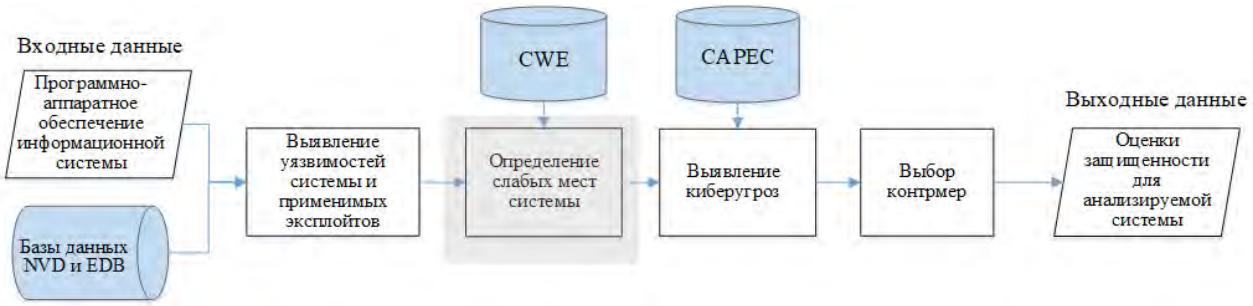

Рис. 1. Основные этапы анализа защищенности и выбора защитных мер

4) выбор мер безопасности для защиты от выявленных киберугроз.

В качестве входных данных предлагается использовать конфигурацию системы, включая программное и аппаратное обеспечение (ПО и АО), записи об уязвимостях из NVD и эксплойты из EDB.

На первом этапе определяются уязвимости системы или доступные эксплойты на основе анализа программного и аппаратного обеспечения системы, баз данных NVD и EDB, а также характеристики уязвимостей по системе CVSS из NVD.

На втором этапе определяются слабые места системы. В конце этого этапа должны быть известны уязвимости в формате CVE (далее CVE), слабые места в формате CWE (далее CWE) и применимые эксплойты анализируемой системы. Однако предварительный анализ баз данных по безопасности показал недостаточную связанность и информативность имеющихся данных, что обусловлено несогласованностью процессов обновления баз данных по безопасности. Так, могут существовать следующие виды неопределенностей:

1) CVE и эксплойты известны, а CWE неизвестны;
2) CVE и CWE известны, а эксплойты неизвестны;
3) эксплойты и CWE известны, а CVE неизвестны;
4) CVE известны, а эксплойты и CWE неизвестны;
5) эксплойты известны, а CVE и CWE неизвестны;
6) CWE известны, а эксплойты и CVE неизвестны;
7) CVE, эксплойты и CWE неизвестны.

Неопределенности видов 2, 3 и 6 рассматривать не будем, так как задача оценивания защищенности и выбора мер защиты в этих ситуациях вполне решаема. Так, при неопределенности вида 2 вначале определяются уязвимости системы. Затем по ссылкам из соответствующих записей в базах NVD и CWE определяются слабые места системы.

Однако достаточно большое количество записей об уязвимостях в базе NVD не имеет ссылок на используемые слабые места. Это соответствует случаям 1 и 4. Случай 5 соответствует ситуации, когда в базе данных EDB есть эксплойт, но он не связан с CVE. Случай 7 характерен для эксплуатации неизвестных ранее уязвимостей, слабых мест и эксплойтов.

Предлагаемая концепция динамической оценки защищенности информационных систем в условиях неопределенности ориентирована на объективную (эмпирическую) систему формального описания знаний о различных аспектах безопасности защищаемой инфраструктуры с реализацией функционала моделирования потенциальных угроз в реальном времени (рис. 2). Она предполагает проведение статического и динамического анализа эксплойтов. Статический анализ позволяет устранить неопределенности за счет классификации эксплойтов на основе признаков, сформированных с использованием признаков связанных уязвимостей или модели исходного кода эксплойта. Динамический анализ позволяет устранить неопределенности путем сопоставления этих признаков событиям, генерируемым в процессе атаки на систему. Таким образом, в рамках предлагаемой концепции для проведения статического анализа разработаны: методика классификации эксплойтов на основе признаков связанных уязвимостей, методика классификации эксплойтов на основе признаков, сформированных с использованием модели исходного кода эксплойта; модель исходного кода эксплойта и методика ее построения.

Последующее выявление фрагментов кода эксплойтов и признаков их выполнения происходит в рамках динамического анализа путем выполнения следующих этапов (рис. 2):

1) выявление наиболее часто используемых фрагментов злонамеренного кода и вычисление их статистических показателей;

2) вычисление вероятностей перехода от одного фрагмента кода к другому;

Рис. 2. Схема динамической оценки защищенности информационных систем в условиях неопределенности

3) выявление признаков выполнения фрагментов кода и их сопоставление концептам модели исходного кода эксплойта для последующего прогнозирования шагов кибератаки.

На рис. 2 прямоугольники и овалы представляют типы информации безопасности. Овалы представляют известные оценки уязвимостей и восстанавливаемые представления эксплойтов в виде моделей и общих признаков. Сплошные линии представляют семантические отношения [15]. Пунктирные линии представляют переходы между этапами.

Для устранения неопределенностей видов 1 и 4 на основе «оценки по CVSS» производится «классификации уязвимостей» для определения «слабого места» и «эксплойта» (в рамках первой методики, разработанной для статического анализа). Для устранения неопределенности вида 5 предлагается следующая методика (методика, построения модели эксплойта): «компиляция» «эксплойта», представленного «исходным кодом», в «исполняемый код». «Исполняемый код» используется для анализа. Процесс анализа исходного кода эксплойта намного сложнее, чем анализ исполняемого кода, обрабатываемого интерпретатором. Этап декомпиляции на рис. 2 опущен, поскольку он выходит за рамки данной работы. Декомпилированный «исполняемый код» используется для «моделирования» семантической «модели кода эксплойта» на основе графа потока управления и вызовов функций импортируемых модулей.

В результате генерации множества «моделей кода эксплойта» и их последующего синтеза «извлекаются» «общие признаки эксплойтов». Эти признаки используются в задаче классификации с целью определения соответствующих «слабых мест» и «уязвимостей» (вторая методика, разработанная для статического анализа). На данном этапе исследований эффективность моделирования кода эксплойтов «оценивается» на основе корреляции их признаков с известными признаками уязвимостей CVSS. Наконец, для устранения неопределенности вида 7 добавляется этап сопоставления «общих признаков эксплойтов» с признаками, сформированными в результате анализа событий, генерируемых в системе в процессе реализации атаки (методика динамического анализа).

На рис. 2 также представлены два типа информации безопасности, которые не используются в предлагаемом подходе, – «атака» и «продукт». Они необходимы для связи подхода с конкретной информационной системой для выбора мер безопасности (см. рис. 1), так как меры безопасности зависят от возможных атак, которые, в свою очередь, зависят от слабых мест и уязвимостей конкретной системы.

Сбор и анализ исходных данных

Как упоминалось выше, в качестве входных данных предлагается использовать конфигурацию системы, включая программное и аппаратное обеспечение, записи об уязвимостях из NVD и эксплойтах из EDB.

База NVD выбрана по следующим причинам:

1) она включает наибольшее количество известных уязвимостей [16];

2) она поддерживается экспертами NIST;

3) она содержит оценки CVSS для уязвимостей и

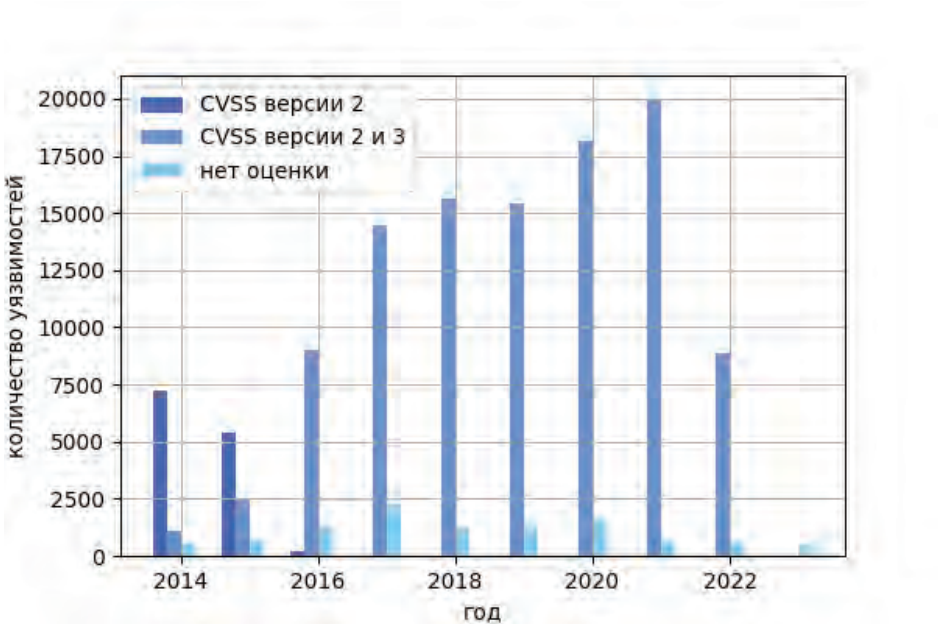

Рис. 3. Распределение количества уязвимостей с учетом используемой версии CVSS по годам

ссылки на CWE, необходимые для установления связей между уязвимостями и атаками.

База EDB, в свою очередь, является наиболее известной открытой базой данных эксплойтов. В ее записях есть ссылки на CVE и на описание уязвимой платформы и программного обеспечения.

Записи об уязвимостях в NVD имеют признаки, оцениваемые с помощью CVSS версии 2.0. В последние годы записи в NVD пополняются признаками, оцениваемыми с помощью CVSS версии 3.0 (рис. 3). Так, с 2017 г. оценки по CVSS версии 2 практически не используются. При этом отметим, что поскольку база EDB перестала обновляться в 2018 г., а в работе при проведении экспериментов используется именно она, авторами используются признаки обеих версий CVSS.

Согласно матрице коэффициентов корреляции Пирсона [17] логическая корреляция значений признаков CVSS версии 2.0 и 3.0 для признаков подгруппы воздействия составляет 78 – 84%. Значения логической корреляции для этих признаков равны 53 – 61% и 15 – 54% в рамках CVSS версии 2.0 и 3.0, соответственно. Взаимосвязь между значениями других признаков CVSS обеих версий имеет коэффициент корреляции меньше 0,82.

Были проанализированы уязвимости из NVD с 2014 года по 2018 год. При оценке корреляции между признаками из исходной выборки были исключены записи об уязвимостях, не имеющих признаков CVSS Записи об уязвимостях, не имеющих ссылок на CWE и реализующих редкие классы CWE, также были исключены из рассмотрения.

Используя записи об уязвимостях из базы NVD и классификацию слабых мест CWE, были созданы исходные наборы данных для обучения и тестирования классификаторов. В таблице 1 приведены характеристики исходных наборов данных для двух различных представлений CWE.

Форматы представления исходных данных – JSON (JavaScript Object Notation)[14] и CSV (Comma Separated Values)[15] для баз данных NVD и CWE, соответственно. Загрузка и предварительная обработка данных осуществлялись с использованием Python.

Кроме того, была проанализирована база данных EDB. Ее анализ показал, что наиболее популярным языком для разработки эксплойтов является Python (рис. 4). Поэтому в данной работе анализируются эксплойты только этого типа. На рисунке 4 представлены результаты за 2014-2018 гг.

База EDB содержит следующие типы эксплойтов (рис. 5): 4292 webapps, 1829 dos, 1078 local, 1030 remote, 287 shellcode. Для аппаратной платформы существует 531 эксплойт, для операционной системы —

14 https://javaee.github.io/tutorial/jsonp001.html
15 https://www.ietf.org/rfc/rfc4180.txt

Таблица 1

Характеристики исходных наборов данных

CWE представление / Параметры набора данных	Для исследователей	Для разработчиков
Количество реальных объектов	17668	5786
Набор объектов	28000	2000
Количество классов	4	5
CWE классы	(118,664,693,707)	(116,119,284,345,682)

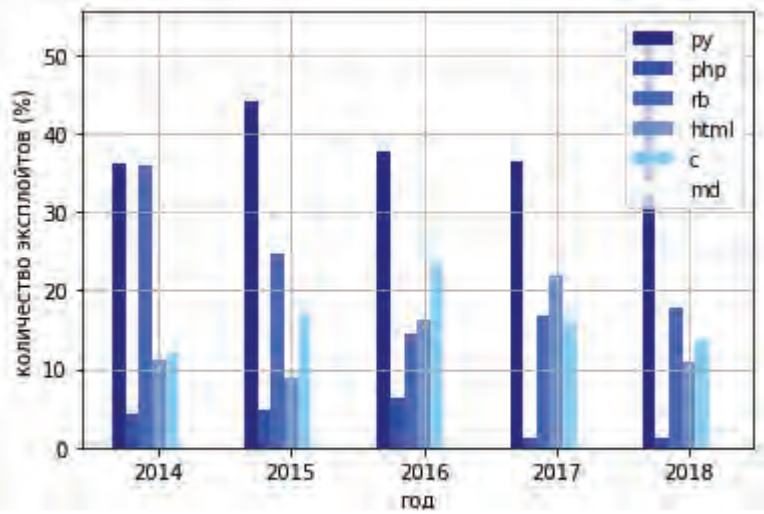

Рис. 4. Статистики языков разработки эксплойтов в EDB за 2014-2018 гг.

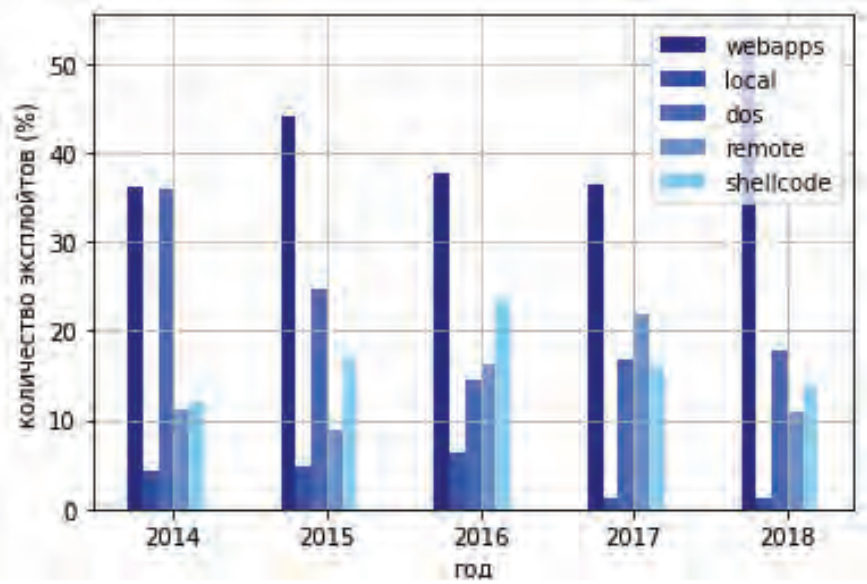

Рис. 5. Статистики типов эксплойтов в EDB за 2014-2018 гг.

3610 эксплойтов, для программной платформы — 132 эксплойта, для WEB-платформы — 3432 эксплойта и 1646 неопределенных.

Все эксплойты представлены одним файлом исходного кода. Это исключает применение пользовательских модулей, выходящих за рамки стандартных библиотек интерпретатора, или модулей, не описанных в файле (неизвестные модули). Исходя из этого, все модули, используемые эксплойтами, известны и будут перечислены в наборе имен декомпилированных кодов.

Модель и методика абстрактного представления исходного кода эксплойтов

Предлагаемая модель эксплойтов является вариацией семантического графа. Она построена на основе графа CFG, в котором узлы отображают базовые блоки исходного кода, а дуги соединяют узлы, которые могут исполняться последовательно, и графа зависимостей вызовов функций, в котором узлы отображают функции, а дуги – вызовы функций. В качестве концептов предлагаемой модели выступают фрагменты исходного кода эксплойтов.

Для формирования модели предлагается методика, включающая следующие этапы:

1) компиляция исходного кода;

2) декомпиляция исходного кода;

3) построение функционально-семантических моделей эксплойтов;

4) сравнение моделей;

5) извлечение устойчивых и значимых частей моделей эксплойтов для определения классов уязвимостей.

Предварительная компиляция исходного кода эксплойта в машинный код Python (формат *.pyc) на этапе 1 и последующая его декомпиляция (дизассемблирование) в последовательность инструкций (опкодов) интерпретатора на этапе 2 позволяет исключить эксплойты с синтаксическими ошибками в исходном коде. Ошибки могут быть связаны с применением эксплойта конкретной версии среды языка Python.

На этапе 3 используется графовая модель кода эксплойта. Узлы графа задаются «именами» исходного кода, которые представляют собой импортируемые модули и их функции. Такие модули составляют стандартную среду выполнения кода, поскольку все исходные коды успешно прошли этап компиляции. Ребра графа задают последовательность вызовов функций для таких модулей. Связь от исходного узла к узлу назначения отражает использование соответствующего результата функции в качестве аргумента функции узла назначения.

Модель генерируется на основе CFG, узлы которого заменяются функциями и классами объектов импортируемых модулей. Граф CFG отражает маршрут от точки входа (Entry Point, EP) до «неявного возврата» (Implicit Return, IR). Для генерации CFG проводится анализ дизассемблированного кода эксплойта с целью поиска условных и безусловных, относительных и абсолютных переходов, а также инструкций обработчика исключений и некоторых других способов ветвления кода. Для генерации результирующего графа семантической функциональной модели эксплойта определяются зависимости использования имен (вызовов) из импортирующих модулей для основного маршрута CFG. Промежуточные (локальные) имена, введенные авторами в процессе разработки эксплойта, не рассматриваются. Отличие предложенной модели заключается, с одной стороны, в строгом следовании основному маршруту выполнения кода, а с другой — в отражении только функциональных зависимостей между импортируемыми именами.

На этапе 4 выполняется сравнение моделей различных эксплойтов. Сгенерированные последовательности зависимостей между именами могут существенно отличаться даже при схожей семантике. Таким образом, для генерации общих признаков кодов эксплойтов на этапе 5 предлагается использовать только связанные пары имен. На данном этапе исследования для предварительного подтверждения гипотезы сравнивались сгенерированные последовательности зависимостей имен для эксплойтов и оценки CVSS для соответствующих уязвимостей.

Экспериментальная оценка предложенного подхода

Анализ данных проводился с использованием Python и библиотеки scikit-learn[16], реализующей выбранные обучающие модели. Выборка для тестирования моделей с оптимальными с точки зрения точности гиперпараметрами составляет 30%, а обучающая выборка – 70% от исходного набора данных.

В таблице 2 приведены результаты классификации уязвимостей для двух представлений CWE. Были использованы следующие методы классификации: дерево решений (Decision Tree, DT), k-ближайших соседей (k-Nearest Neighbors, kNN) и случайный лес (Random Forest, RF). Далее для каждого концепта представлены три характеристики прогностической модели: 1) гиперпараметры модели, дающие максимальную точ-

16 https://scikit-learn.org/stable/

Таблица 2

Результаты классификации уязвимостей

Метод	CWE представление / Гиперпараметр	Для исследователей			Для разработчиков		
		1	2	3	1	2	3
DT	max_depth	13	70.5%	71.5%	14	70.3%	70.4%
	max_features	7			10		
kNN	n_neighbors	16	68.5%	70.4%	9	68%	67.1%
RF	max_depth	10	70.6%	71.7%	10	70.4%	70.5%
	max_features	4			8		

ность; 2) точность классификации при перекрестной валидации (5 блоков); 3) точность классификации на отложенном валидационном наборе данных.

Гиперпараметрами прогностической модели для методов DT и RF являются максимальная глубина дерева (max_depth) и максимальное количество используемых признаков (max_features), для метода kNN – количество соседей. Как видно из таблицы 2, наилучшие результаты классификации получены при использовании метода RF. Так как текущая точность классификации еще является низкой, планируется повысить ее в дальнейшей работе, используя более сложные алгоритмы, включая глубокое обучение.

На рис. 6 представлены метрики, показывающие значимость признаков для классификации по рассматриваемым представлениям CWE. Показана разница в представлениях CWE с точки зрения информативности признаков, используемых для классификации. Наиболее значимыми признаками являются CVSSv3:Scope и CVSSv3:User Interaction из представления CWE Develop Concepts с показателем значимости 0,15. Показатель значимости признаков варьируется в диапазоне от 0 до 1.

Таким образом, классификация уязвимостей (по классам эксплойтов) с использованием различных методов достигает точности в 70%, что обусловлено характером исходных данных. Эти данные представляют собой признаки CVSS, в которых категориальные значения преобладают над количественными. В результате набор возможных значений признаков CVSS, используемых для задачи классификации, очень ограничен.

Для повышения точности классификации в ситуации, когда уязвимость неизвестна, а, например, эксплойт может быть известен или нет, предлагается вторая методика, основанная на анализе эксплойтов. Опишем эксперименты для первых трех этапов этой методики, поскольку именно они обусловливают ее жизнеспособность.

Этапы 1 и 2 (компиляция и декомпиляция). За последние 5 лет было разработано 1315 эксплойтов, использующих язык Python. Исходный код 1153 эксплойтов был скомпилирован без ошибок. Поэтому из набора данных было исключено не более 13% эксплойтов. Успешно скомпилированные эксплойты были использованы в качестве исходных данных для этапа 3.

Этап 3 (генерация модели). Из исполняемого кода эксплойтов извлекались «имена» для создания графовой модели. Эти «имена» формируют узлы графа. 4690 имен были извлечены из 1153 успешно скомпилированных эксплойтов. Минимальное количество имен в эксплойте – один, максимальное количество имен в эксплойте – 132.

Анализ извлеченных имен показал, что наиболее часто используемыми являются следующие имена: ‘close’ (количество использований — 582), ‘sys’ (487), ‘len’ (461), ‘open’ (452), ‘write’ (445), ‘socket’ (375), ‘exit’ (288), ‘buffer’ (273), ‘argv’ (266), ‘name’ (248), ‘f’ (242), ‘os’ (232), ‘struct’ (226), ‘payload’ (213), ‘AF_INET’ (212), ‘send’ (206), ‘connect’ (196), ‘SOCK_STREAM’ (195), ‘doc’ (190), ‘s’ (187), ‘time’ (182), ‘port’ (180), ‘shellcode’ (164), ‘requests’ (163), ‘str’ (161), ‘file’ (156), ‘host’ (151) и т.д. Они четко отражают функциональную принадлежность эксплойтов. Однако полученных данных недостаточно для выделения признаков, так как импорт имени не означает его использование в коде, особенно в основном потоке управления эксплойта. Более того, один эксплойт может использовать несколько высокоуровневых и часто

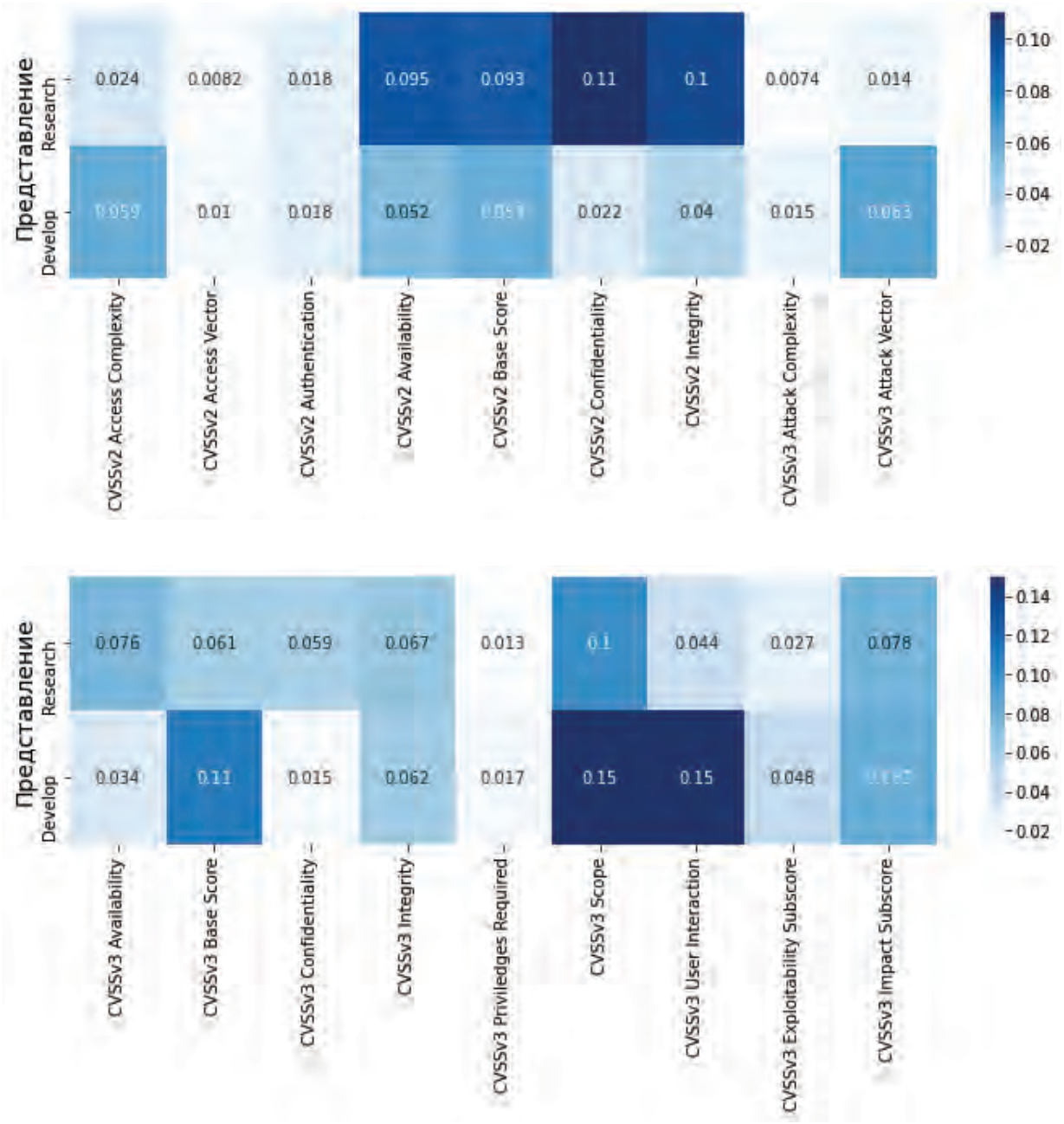

Рис. 6. Значимость признаков для классификации с использованием метода RF

используемых имен, которые имеют сложные связи зависимостей и отношения потока выполнения.

Модель эксплойта в виде CFG, узлы которого заменяются функциями и классами объектов импортируемых модулей, была сгенерирована на основе извлеченных имен и зависимостей использования имен из импортируемых модулей. Пример извлечения модели кода эксплойта приведен на рис. 7, на котором показан граф CFG (слева) и его преобразование в модель эксплойта (справа) для эксплойта с идентификатором EDB-ID = 30688. В графе CFG узлы соответствуют блокам кода. Число в узле представляет собой смещение первой инструкции каждого блока. Кратчайший путь выполнения кода выделен жирным шрифтом и соответствует обычному выполнению программы. Узлы за пределами основного пути представляют блоки, обрабатывающие исключения и нерегулярные ситуации.

Данная модель эксплойта отражает порядок и зависимости вызовов импортируемых имен. В рассматриваемом случае – это имена функций urlencode() и urlopen() и конструктора класса Request. Первые два вызова выполняются в одном блоке, поэтому выход

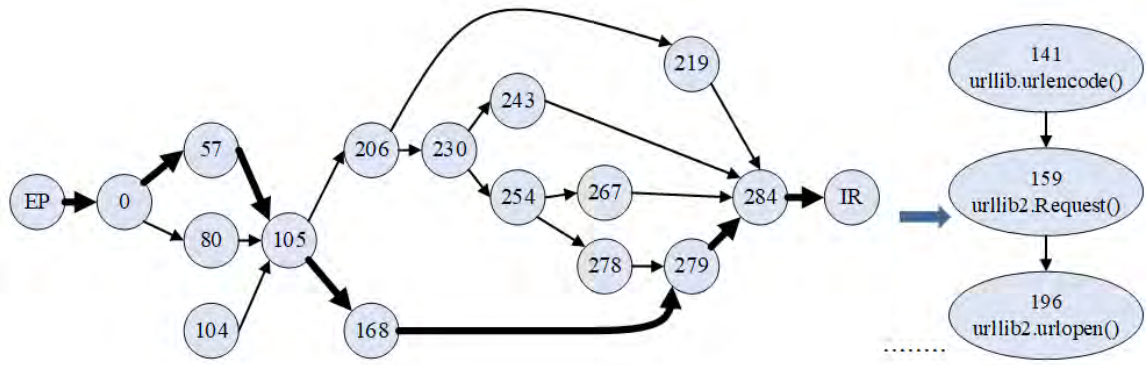

Рис. 7. Пример преобразования CFG кода эксплойта

функции urlencode() будет безоговорочно отправлен в конструктор класса Request в случае отсутствия исключений. Эта связь основана только на зависимости вызовов. Вызов функции urlopen() производится в другом блоке. Но в нем используется объект класса Request, созданный ранее. Эта связь строится как на основе CFG, так и на основе функциональной зависимости.

Кроме того, код эксплойта содержит еще два импортированных модуля. Однако вызовы этих модулей не отражают функциональных особенностей кода. Упрощение CFG путем нахождения единственного кратчайшего пути позволяет исключить множество имен, которые не характеризуют функциональность кода.

Статический анализ исходного кода является достаточно исследованной областью анализа уязвимостей, что было подробно рассмотрено при анализе известных работ. К сожалению, существующие модели и практические реализации графа для Python-кода не применимы для этой цели, поскольку необходимо формализованное описание функциональности кода. Поэтому была предложена новая графовая модель, объединяющая CFG и PDG. С ее помощью была подтверждена гипотеза о том, что эксплойты для уязвимостей, использующих схожие слабые места, имеют семантически схожий функционал.

Заключение

В работе предложена концепция динамического оценивания защищенности информационных систем в условиях неопределенности исходных данных. В ее основе лежит модель исходного кода эксплойта и методика ее построения. Их применение позволяет устранить ряд неопределенностей исходных данных при оценивании защищенности. Разработанные концепция, модель и методика ее построения основаны на открытых базах данных по безопасности и анализе представленных в них данных. Предложены две методики статического анализа в целях оценивания защищенности, первая из которых проводит анализ общедоступных признаков уязвимостей CVSS из NVD, а вторая дополняет первую методику в случае, если в EDB есть эксплойты, не относящиеся к уязвимостям, и поэтому индексы для классификации отсутствуют. Она основана на анализе кода эксплойтов для извлечения признаков с помощью графовых моделей. Далее извлеченные признаки используются для определения и оценивания слабых мест и эксплойтов по первой методике.

Приведенная статистика по используемым открытым базам данных по безопасности обосновывает актуальность рассматриваемой задачи и разработанных методик. Описан процесс сбора и предварительной обработки данных, от которого зависят результаты экспериментов. Проведенные эксперименты продемонстрировали высокую эффективность и потенциал разработанных методик. При этом экспериментально подтверждена гипотеза о том, что эксплойты для уязвимостей, использующих схожие слабые места (то есть относящиеся к одному классу), имеют семантически схожий функционал, а значит применимости предложенных модели и методики для динамического выявления и оценивания эксплойтов, и как следствие – динамического оценивания защищенности информационных систем.

В дальнейшей работе планируется развить разработанные методики и уточнить механизм их совместного применения для улучшения классификации уязвимостей и эксплойтов по слабым местам, определить набор признаков для классификации на основе модели эксплойта, а также развить эксперименты, подтверждающие эффективность подхода.

Благодарность. *Работа выполнена при поддержке гранта РНФ № 23-21-00498 в СПб ФИЦ РАН.*

Рецензент: *Лаута Олег Сергеевич, доктор технических наук, профессор кафедры комплексного обеспечения информационной безопасности Государственного университета морского и речного флота имени адмирала С.О. Макарова, Санкт-Петербург, Россия.*

E-mail: laos-82@yandex.ru

Литература

1. Mell P., Scarfone K., Romanosky S. A complete guide to the Common Vulnerability Scoring System. Version 2.0. – URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=51198 (дата обращения 10.04.2023).
2. CVSS v3.1 Specification Document – Revision 1. June 2019. – URL: https://www.first.org/cvss/v3-1/cvss-v31-specification_r1.pdf (дата обращения 10.04.2023).
3. Wang T., Lv Q., Hu B., Sun D. CVSS-based multi-factor dynamic risk assessment model for network system // Proceedings of the 2020 IEEE 10th International Conference on Electronics Information and Emergency Communication, Beijing, China, 2020. – pp. 289-294. DOI: 10.1109/ICEIEC49280.2020.9152340.
4. Debnath J.K., Xie D. CVSS-based vulnerability and risk assessment for high performance computing networks // Proceedings of the 2022 IEEE International Systems Conference, Montreal, QC, Canada, 2022. – pp. 1-8. DOI: 10.1109/SysCon53536.2022.9773931.
5. Figueroa-Lorenzo S., Añorga J., Arrizabalaga S. A survey of IIoT protocols: A measure of vulnerability risk analysis based on CVSS // ACM Computing Survey, 2021, vol. 53, no. 2, art. 44, 53 p. DOI: 10.1145/3381038.
6. Aksu M.U., Bicakci K., Dilek M.H., Ozbayoglu A.M., Tatli E. Automated generation of attack graphs using NVD // Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy, Tempe, AZ, USA, 2018. – pp. 135-142. DOI: 10.1145/3176258.3176339.
7. Özdemir Sönmez F., Hankin C., Malacaria P. Attack dynamics: An automatic attack graph generation framework based on system topology, CAPEC, CWE, and CVE databases // Computers & Security, 2022, vol. 123, pp. 102938. DOI: https://doi.org/10.1016/j.cose.2022.102938.
8. Дойникова Е.В., Котенко И.В. Оценивание защищенности и выбор контрмер для управления кибербезопасностью. Монография. – М.: Российская академия наук, 2021.
9. Longueira-Romero Á., Iglesias R., Flores J.L., Garitano I. A novel model for vulnerability analysis through enhanced directed graphs and quantitative metrics // Sensors, 2022, vol. 22, no. 6, pp. 2126. DOI: 10.3390/s22062126.
10. Doynikova E., Kotenko I. CVSS-based probabilistic risk assessment for cyber situational awareness and countermeasure selection // Proceedings of the 2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing, St. Petersburg, Russia, 2017, pp. 346-353. DOI: 10.1109/PDP.2017.44.
11. Syed R., Zhong H. Cybersecurity Vulnerability Management: An ontology-based conceptual model // Americas Conference on Information Systems. – 2018.
12. Kalgutkar V., Kaur R., Gonzalez H., Stakhanova N., Matyukhina A. Code Authorship Attribution: Methods and Challenges // ACM Computing Survey, 2020, vol. 52, no. 1, art. 3, 36 p. DOI: 10.1145/3292577.
13. Patterson E., Baldini I., Mojsilovic´ A., Varshney K.R. Semantic representation of data science programs // Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 2018. – pp. 5847-5849.
14. Zhang Y., Chen L., Nie X., Shi G. An effective buffer overflow detection with super data-flow graphs // Proceedings of the 2022 IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking, Melbourne, Australia, 2022. – pp. 684-691. DOI: 10.1109/ISPA-BDCloud-SocialCom-SustainCom57177.2022.00093.
15. Kotenko I., Doynikova E., Fedorchenko A., Chechulin A. An ontology-based hybrid storage of security information // Information Technology and Control, 2018, vol. 4, pp. 655-667. DOI: 10.5755/j01.itc.47.4.20007.
16. Kotenko I., Fedorchenko A., Doynikova E. Data analytics for security management of complex heterogeneous systems // EAI/Springer Innovations in Communication and Computing. Springer, Cham, 2020, vol. 3, pp. 79-116. DOI: 10.1007/978-3-030-19353-9_5.
17. Doynikova E., Fedorchenko A., Kotenko I. Determination of security threat classes on the basis of vulnerability analysis for automated countermeasure selection // Proceedings of the 13th International Conference on Availability, Reliability and Security, Hamburg. Germany, 2018. – pp. 621-628. DOI: 10.1145/3230833.3233260.

ASSESSMENT OF INFORMATION SYSTEM SECURITY BASED ON THE EXPLOIT'S GRAPH MODEL

Fedorchenko E.V.[17], Kotenko I.V.[18], Fedorchenko A.V.[19], Novikova E.S.[20], Saenko I.B.[21]

The purpose of the study: *automating the processes of identifying and evaluating exploits, an information system is vulnerable, by identifying their features based on the analysis of the exploit source code, related weaknesses and vulnerabilities in order to further eliminate them and increase the security of information systems.*

Research methods: *statistical analysis of source data, semantic and syntactic modeling of the process of executing the source code of exploits, data classification methods for evaluating exploits based on related weaknesses and vulnerabilities.*

Results obtained: *a general concept of dynamic security assessment of information systems under conditions of initial data uncertainty is proposed; within the framework of the proposed concept, the data used in security assessment, the relationships between them and the main types of uncertainties associated with the use of previously unknown vulnerabilities, weaknesses of the analyzed system or exploits, are identified. there are methods of static and dynamic analysis of exploits in order to eliminate the identified uncertainties; data sources and initial data for experiments were identified, their statistical analysis was carried out; proposed a technique for eliminating the identified uncertainties based on the classification of exploits using the signs of their associated vulnerabilities; an experimental assessment of the accuracy of the classification of exploits was made, and the shortcomings of the proposed methodology were highlighted; to eliminate the identified shortcomings, a graph model of exploits and a methodology for its formation were developed; a technique for classifying exploits based on features generated using the developed model and related weaknesses and vulnerabilities is proposed. The results obtained can be used in monitoring systems and improving the security of information systems.*

Scientific novelty: *the proposed general concept of dynamic assessment of the security of information systems differs from the existing ones in the identified types of uncertainty in the initial data and the use of exploit classification techniques to eliminate them by detecting signs of exploit implementation, and the proposed concept is based on the hypothesis that previously unknown exploits use previously known fragments malicious program code; the proposed methodology for classifying exploits differs both in the use of known features of related vulnerabilities and features based on a graph model of exploits; The developed graph model of exploits is a variation of the semantic graph, built on the basis of the control flow graph and function call dependency graph, and allows taking into account both the main code execution route and functional dependencies between imported function names when generating signs of exploit execution.*

Contribution: *Elena Fedorchenko – development of a methodology for classifying exploits based on features generated using a graph model of the source code of exploits and related weaknesses and vulnerabilities; Igor Kotenko and Andrey Fedorchenko – analysis of the state of affairs in the presentation of the source code of exploits for the purpose of dynamic assessment of the security of information systems, setting the problem of classifying exploits, developing an approach to obtaining classification features; Andrey Fedorchenko – collection and preliminary analysis of initial data; Evgenia Novikova – experimental study of the proposed approach; Igor Saenko and Elena Fedorchenko – development of the concept of dynamic assessment of the security of information systems in the face of uncertainty in the initial data.*

17 Elena V. Fedorchenko, Ph.D., Senior 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: doynikova@comsec.spb.ru

18 Igor V. Kotenko, Dr.Sc., Professor, 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

19 Andrey V. Fedorchenko, Programmer of the Laboratory of Computer Security Problems at St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS), St. Petersburg, Russia. E-mail: fedorchenko@comsec.spb.ru

20 Evgenia S. Novikova, Ph.D., Senior 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: novikova@comsec.spb.ru

21 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

DOI:10.21681/2311-3456-2023-3-23-36

Keywords: *weakness, vulnerability, features, data analysis, data classification, security monitoring.*

References

1. Mell P., Scarfone K., Romanosky S. A complete guide to the Common Vulnerability Scoring System. Version 2.0. – URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=51198 (accessed on 10.04.2023).
2. CVSS v3.1 Specification Document – Revision 1. June 2019. – URL: https://www.first.org/cvss/v3-1/cvss-v31-specification_r1.pdf (accessed on 10.04.2023).
3. Wang T., Lv Q., Hu B., Sun D. CVSS-based multi-factor dynamic risk assessment model for network system // Proceedings of the 2020 IEEE 10th International Conference on Electronics Information and Emergency Communication, Beijing, China, 2020. – pp. 289-294. DOI: 10.1109/ICEIEC49280.2020.9152340.
4. Debnath J.K., Xie D. CVSS-based vulnerability and risk assessment for high performance computing networks // Proceedings of the 2022 IEEE International Systems Conference, Montreal, QC, Canada, 2022. – pp. 1-8. DOI: 10.1109/SysCon53536.2022.9773931.
5. Figueroa-Lorenzo S., Añorga J., Arrizabalaga S. A survey of IIoT protocols: A measure of vulnerability risk analysis based on CVSS // ACM Computing Survey, 2021, vol. 53, no. 2, art. 44, 53 p. DOI: 10.1145/3381038.
6. Aksu M.U., Bicakci K., Dilek M.H., Ozbayoglu A.M., Tatli E. Automated generation of attack graphs using NVD // Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy, Tempe, AZ, USA, 2018. – pp. 135-142. DOI: 10.1145/3176258.3176339.
7. Özdemir Sönmez F., Hankin C., Malacaria P. Attack dynamics: An automatic attack graph generation framework based on system topology, CAPEC, CWE, and CVE databases // Computers & Security, 2022, vol. 123, pp. 102938. DOI: https://doi.org/10.1016/j.cose.2022.102938.
8. Doynikova E., Kotenko I. Assessment of security and choice of countermeasures for cybersecurity management. Monography. – Moscow: Russian Academy of Sciences, 2021.
9. Longueira-Romero Á., Iglesias R., Flores J.L., Garitano I. A novel model for vulnerability analysis through enhanced directed graphs and quantitative metrics // Sensors, 2022, vol. 22, no. 6, pp. 2126. DOI: 10.3390/s22062126.
10. Doynikova E., Kotenko I. CVSS-based probabilistic risk assessment for cyber situational awareness and countermeasure selection // Proceedings of the 2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing, St. Petersburg, Russia, 2017, pp. 346-353. DOI: 10.1109/PDP.2017.44.
11. Syed R., Zhong H. Cybersecurity Vulnerability Management: An ontology-based conceptual model // Americas Conference on Information Systems. – 2018.
12. Kalgutkar V., Kaur R., Gonzalez H., Stakhanova N., Matyukhina A. Code Authorship Attribution: Methods and Challenges // ACM Computing Survey, 2020, vol. 52, no. 1, art. 3, 36 p. DOI: 10.1145/3292577.
13. Patterson E., Baldini I., Mojsilovic´ A., Varshney K.R. Semantic representation of data science programs // Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 2018. – pp. 5847-5849.
14. Zhang Y., Chen L., Nie X., Shi G. An effective buffer overflow detection with super data-flow graphs // Proceedings of the 2022 IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking, Melbourne, Australia, 2022. – pp. 684-691. DOI: 10.1109/ISPA-BDCloud-SocialCom-SustainCom57177.2022.00093.
15. Kotenko I., Doynikova E., Fedorchenko A., Chechulin A. An ontology-based hybrid storage of security information // Information Technology and Control, 2018, vol. 4, pp. 655-667. DOI: 10.5755/j01.itc.47.4.20007.
16. Kotenko I., Fedorchenko A., Doynikova E. Data analytics for security management of complex heterogeneous systems // EAI/Springer Innovations in Communication and Computing. Springer, Cham, 2020, vol. 3, pp. 79-116. DOI: 10.1007/978-3-030-19353-9_5.
17. Doynikova E., Fedorchenko A., Kotenko I. Determination of security threat classes on the basis of vulnerability analysis for automated countermeasure selection // Proceedings of the 13th International Conference on Availability, Reliability and Security, Hamburg. Germany, 2018. – pp. 621-628. DOI: 10.1145/3230833.3233260.