ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ТЕЛЕКОММУНИКАЦИИ
INFORMATION TECHNOLOGY AND TELECOMMUNICATIONS

УДК 004.056

DOI: 10.21822/2073-6185-2025-52-2-98-106 Оригинальная статья/ Original article

Причины возникновения, классификация и критичность уязвимостей программного обеспечения информационных систем

А.О. Ефимов
Воронежский институт МВД России,
394065, г. Воронеж, пр. Патриотов, 53, Россия

Резюме. Цель. Целью исследования является описание причин возникновения уязвимостей программного обеспечения, а также их классификация; устранение неопределённости понятия «критичность уязвимости программного обеспечения». **Метод.** Анализ существующих подходов к оценке, классификации, и определения уязвимостей программного обеспечения. Моделирование, построение UML модели, описание алгоритма. **Результат.** Дано определение критичности уязвимости программного обеспечения. Частично описаны причины возникновения уязвимостей программного обеспечения, расширена существующая классификация уязвимостей программного обеспечения. Приведен пример оценки критичности уязвимостей программного обеспечения на основе вычисляемых метрик. Приведен пример оценки и приоритезации метрик уязвимостей. Представлены UML модель и алгоритм оценки уязвимостей. **Вывод.** Результаты проведенного исследования расширяют перечень показателей и предметную область описания уязвимостей программного обеспечения.

Ключевые слова: уязвимость, критичность уязвимости, оценка уязвимостей, безопасность программного обеспечения, классификация уязвимостей.

Для цитирования: А.О. Ефимов. Причины возникновения, классификация и критичность уязвимостей программного обеспечения информационных систем. Вестник Дагестанского государственного технического университета. Технические науки. 2025; 52(2):98-106. DOI:10.21822/2073-6185-2025-52-2-98-106.

Causes, classification, and criticality of information system software vulnerabilities

A.O. Efimov
Voronezh Institute of the Ministry of Internal Affairs of Russia,
53 Patriotov Str., Voronezh 394065, Russia

Abstract. **Objective**. The purpose of this paper is to describe the causes of software vulnerabilities, as well as their classification. Eliminating the ambiguity of the concept of software vulnerability criticality. **Method**. Analysis of existing approaches to the assessment, classification, and identification of software vulnerabilities. Modeling, building a UML model, and describing an algorithm. **Result**. A definition of the criticality of a software vulnerability is proposed. The causes of the software are partially described, and the existing classification of software vulnerabilities is expanded. An example of assessing the criticality of software vulnerabilities based on calculated metrics is given. An example of vulnerability metrics evaluation and prioritization is given. The UML model and vulnerability assessment algorithm are presented. **Conclusion**. The results of the conducted research expand the list of indicators and the subject area of the description of software vulnerabilities.

Keywords: vulnerability, criticality of vulnerability, vulnerability assessment, software security, vulnerability classification

For citation: A.O. Efimov. Causes, classification, and criticality of information system software vulnerabilities. Herald of the Daghestan State Technical University. Technical Sciences. 2025; 52(2):98-106. (In Russ) DOI:10.21822/2073-6185-2025-52-2-98-106

Введение. Эксплуатация уязвимостей программного (программно-технического) обеспечения является одним из основных путей реализации угроз информационной безопасности. Рассмотрение одних лишь уязвимостей в отрыве от организационных и правовых мер защиты информации не даёт полноценной картины при рассмотрении защищенности информации в информационных (автоматизированных) системах (ИС), но является основным в случае полноценной реализации вышеуказанных мер, и причинения ущерба путем сетевых атак, преодоления разграничения доступа программных средств и подобных случаев.

В стандарте [1] определено, что уязвимость программного обеспечения – это недостаток (слабость) программного (программно-технического) средства или информационной системы в целом, который (ая) может быть использован (а) для реализации угроз безопасности информации.

Причины возникновения уязвимостей были рассмотрены в различных работах. Из их числа в работе [2] представлено моделирование атак и анализ защищенности компьютерных сетей на основе системы CVSS версии 3, что позволило устранить допущений при формировании графа действий злоумышленника, связанных с определением предусловий и постусловий атакующих действий. В [3] приведено обоснование показателей надёжности программной системы защиты информации от несанкционированного доступа, проведённое на основе анализа основных видов уязвимостей и взаимосвязи компонентов безопасности автоматизированной системы. В статье [4] разработаны модели безопасной системы, предложены критерии количественной оценки уровня безопасности. В [5] построена обобщенная модель комбинированного метода анализа параметрических данных на основе технологий интеллектуального анализа данных, позволяющая повысить эффективность работы специалиста по выявлению уязвимостей контролируемых информационных систем.

Постановка задачи. Целью исследования является описание причин возникновения уязвимостей программного обеспечения, а также их классификация; устранение неопределённости понятия «критичность уязвимости программного обеспечения».

В настоящее время существует несколько видов классификации уязвимостей программного обеспечения, в основном они основаны на понятии критичности (сравнения уязвимостей между собой по уровню опасности). Для унификации процесса оценки критичности уязвимостей применяются специализированные системы. Наиболее популярные из них:

1. Common Vulnerability Scoring System (CVSS). CVSS является международным стандартом для оценки тяжести уязвимостей. Он включает базовые, временные и контекстуальные метрики [6]. Ключевым преимуществом CVSS является его гибкость и возможность адаптации под различные сценарии. CVSS предоставляет числовую оценку (от 0 до 10), где более высокий балл указывает на более высокую критичность уязвимости [6].

2. OWASP Risk Rating Methodology. Методология OWASP ориентирована на анализ рисков, связанных с уязвимостями веб-приложений. Она предполагает использование факторов, таких как вероятность реализации угрозы и потенциальный ущерб. Данный подход позволяет адаптировать оценку под специфические бизнес-процессы [7].

3. Системы отраслевого применения. В определённых отраслях (например, промышленный IoT или системы SCADA) применяются специализированные методологии. Эти подходы акцентируют внимание на уникальных особенностях систем, таких как повышенная уязвимость к физическим атакам или строгие требования к отказоустойчивости. В работе [8] более подробно рассмотрены карты источников, содержащие сведения об уязвимостях программного обеспечения. В тоже время в [9] определено, что степень опасности уязвимости – это мера (сравнительная величина), характеризующая подверженность информационной системы уязвимости и ее влияние на нарушение свойств безопасности информации (конфиденциальность, целостность, доступность). Исходя из вышесказанного можно дать следующее определение:

Критичность уязвимости программного обеспечения – это показатель, указывающий на относительную величину (по сравнению с другими уязвимостями) возможного воздействия на защищаемую информацию (её конфиденциальность, целостность и доступность), а также на вероятность нарушения безопасного функционирования информационной (автоматизированной) системы в случае эксплуатации злоумышленником (или конфликта программных средств и иных способов непреднамеренной эксплуатации) рассматриваемой уязвимости программного обеспечения.

В целом можно сделать вывод что решение проблемы выявления, описания и классификации уязвимостей непосредственно влияет на возможность исключения несанкционированного доступа к ИС.

Методы исследования. Основываясь на существующих работах и исследованиях, в общем виде можно представить два основных множества уязвимостей:

1. Непреднамеренные уязвимости – возникают по причине ошибок при разработке программного кода, конфликта программ, существующих недостатков аппаратного обеспечения, недостатков интерфейсов и стандартов обработки информации. Возникновение непреднамеренных уязвимостей, как правило связано с неконтролируемыми факторами (с человеческим фактором, недостатками при проектировании, ошибками в процессе тестирования, недостаточным документированием, ошибками в зависимых программах и аппаратными сбоями).

2. Преднамеренные уязвимости – заведомо установленные разработчиком недостатки программного обеспечения в целях осуществления несанкционированного доступа к информации и выполнения недекларированных возможностей. Также преднамеренные уязвимости могут возникать по причине использования скомпрометированной среды разработки (в том числе компиляторов и интерпретаторов). Возникновение преднамеренных уязвимостей связаны с намеренными действиями людей или организаций, направленными на внедрение слабых мест в автоматизированные системы. Основные из них: добавление недекларированных возможностей, намеренное добавление скрытых интерфейсов доступа (бэкдоров), интеграция программных закладок в программное или аппаратное обеспечение, создание недекларированных функций для обхода механизмов и средств защиты.

Помимо названных выше причин (которые по большей части относятся к производителю и пользователю программного продукта), также имеют место действия злоумышленников, которые могут быть произведены в процессе дистрибуции посредниками, участвующими в распространении программного обеспечения.

Среди них можно выделить следующие: внедрение вредоносного кода в процессе поставки ПО/оборудования, изменение базового программного обеспечения устройств с целью создания уязвимостей, подмена или модификация компонентов программы на этапе поставки, намеренное внедрение уязвимостей в целях тестирования безопасности продукта, без последующего удаления. В целях расширения предметной области и развития существующих классификаций, предложена следующая классификация уязвимостей:

– по природе происхождения: «преднамеренные», «непреднамеренные»;

– по типу эксплуатации: «обладает возможностью последовательной эксплуатации» (имеется в виду возможность формирования многофакторной уязвимости – уязвимость, появившаяся в результате наличия нескольких недостатков различных типов [9]), «обособленная»;

– по направленности возможного ущерба: «сопряжена с рисками для целостности/конфиденциальности/доступности данных», «сопряжена с рисками работоспособности ИС» (в соответствии с [10] – безопасность информационной технологии: состояние защищенности информационной технологии, при котором обеспечивается выполнение изделием, реализующим информационную технологию, предписанных функций без нарушений безопасности обрабатываемой информации);

– по потенциалу эксплуатации: «эксплуатация не оказывает существенного влияния на информацию», «позволяет причинить ущерб, который возможно отследить и устранить», «позволяет причинить неисправимый ущерб».

Также можно произвести классификацию по привычной шкале «критичности», но подразумевается необходимость отхода от величин экспертных оценок, к вычисляемым и реальным параметрам. По величине критичности: «некритическая», «низкой критичности», «средней критичности», «высокой критичности», «критическая».

Данные величины могут быть соотнесены на основе следующих метрик: «число неиспользуемых, но объявленных переменных», «неограниченные динамические переменные» (например, массивы, не обладающие ограничением к максимальным размерам), «наличие интерфейсов с возможностью запуска скриптов в явном виде», «процент числа проприетарных библиотек», «ограничение числа возникающих процессов обработки ошибок», «процент объема данных, хранящихся в незащищенном пространстве».

Данные показатели могут быть соотнесены для качественной оценки по процентным соотношениям, или абсолютным величинам. Возникает проблема постановки требований к ИС, так как по большей части, требования индивидуальны в зависимости от используемой архитектуры и специфики использования, в связи с чем, в данный момент требования по своей сути все равно будут являться одним из подвидов экспертной оценки. В целях ухода от экспертных оценок необходимо проведение исследований соотношения возникновения уязвимостей программного обеспечения относительно вышеуказанных метрик на основе статистических данных, которые позволят построить более наглядную картину о данном явлении.

Стоит заметить, что полноценной модели, описывающей соотношение данных величин и критичность уязвимостей в настоящее время не существует, и данная проблема является целью будущих исследований.

Так как необходимые статистические данные в настоящее время отсутствуют в свободном доступе, далее будет рассмотрен гипотетический пример применения рассмотренной классификации и метрик.

Описание: «VulTracker» – пример приложения для учёта рабочих задач. Рассмотрим определенные метрики (показатели) уязвимостей.

1. Неиспользуемые, но объявленные переменные. 18% переменных в коде не используются (540 из 3000 объявленных переменных). Наличие неиспользуемых переменных усложняет код, затрудняет его понимание и увеличивает вероятность ошибок при рефакторинге. Помимо этого, открывает дополнительные возможности для использования атак, направленных на переполнение памяти, и внедрения вредоносного кода. Согласно классификации, возможные уязвимости: «преднамеренные», «обособленные», «сопряжены с рисками для целостности/конфиденциальности/доступности данных», «позволяют причинить ущерб, который возможно отследить и устранить».

2. Неограниченные динамические переменные (например, массивы без ограничения максимального размера). 12% используемых массивов (около 24 из 200) не имеют ограничения по размеру. Позволяют привести к инцидентам с переполнением памяти, и отказом в обслуживании. Согласно классификации, возможные уязвимости: «преднамеренные» и «непреднамеренные», «обладают возможностью последовательной эксплуатации», «сопряжены с рисками работоспособности ИС», «позволяют причинить ущерб, который возможно отследить и устранить».

3. Наличие интерфейсов с возможностью запуска скриптов в явном виде. 2 из 7 интерфейсов (14%) позволяют пользователям загружать и запускать собственные скрипты. Интерфейсы могут быть использованы для выполнения несанкционированный действий с базой данных, и выгрузки конфиденциальных данных на внешний сервер. Согласно классификации, возможные уязвимости: «преднамеренные» и «непреднамеренные», «обладают возможностью последовательной эксплуатации», «сопряжены с рисками

для целостности/конфиденциальности/доступности данных» и «сопряжены с рисками работоспособности ИС», «позволяют причинить неисправимый ущерб».

4. Процент проприетарных библиотек. 65% библиотек (13 из 20) являются проприетарными и недоступными для проверки. Компоненты, отвечающие за обработку JSON, содержали устаревшую версию с известной уязвимостью CVE-2023-XXXX, что может позволить злоумышленникам обойти авторизацию и верификацию.

Согласно классификации, возможные уязвимости: «преднамеренные», «обладают возможностью последовательной эксплуатации», «сопряжены с рисками для целостности/конфиденциальности/доступности данных» и «сопряжены с рисками работоспособности ИС», «позволяют причинить неисправимый ущерб».

5. Ограничение числа возникающих процессов обработки ошибок. 22% (11 из 50) потенциальных исключений в коде остаются необработанными. Это может привести к сбоям в работе приложения при некорректных входных данных и оставить пользователю видимыми сообщения отладчика с конфиденциальной информацией.

Согласно классификации, возможные уязвимости: «преднамеренные» и «непреднамеренные», «обладают возможностью последовательной эксплуатации», «сопряжены с рисками для целостности/конфиденциальности/доступности данных» и «сопряжены с рисками работоспособности ИС», «позволяют причинить неисправимый ущерб».

6. Процент объёма данных, хранящихся в незащищённом пространстве.

37% (около 2,8 ГБ из 7,5 ГБ) данных хранятся в текстовом виде без шифрования. Данные могут быть легко скомпрометированы при получении доступа к ИС. Согласно классификации, возможные уязвимости: «преднамеренные», «обособленные», «сопряжены с рисками для целостности/конфиденциальности/доступности данных», «позволяют причинить неисправимый ущерб».

Помимо вышесказанного необходимо отметить, что достижимость уязвимости безусловно влияет на критичность, чем она выше, тем уязвимость критичнее. (Например, величина вероятности достижения уязвимости достаточно квалифицированным злоумышленником от 0% до 100%).

Обсуждение результатов. На основе полученных процентных соотношений возможно произвести градацию критичности возможных уязвимостей. Вопрос достижимости не рассматривается в связи со спецификой внешних и внутренних нарушителей для каждой конкретной ИС.

Критичность уязвимостей программного обеспечения. Рассмотрим простое процентное соотношение приведенного примера, график представлен на рис. 1.

При рассмотрении простого соотношения критичности с шагом в 25% ([0%] – некритическая, [0-25%] – низкий уровень критичности, [25-50%] – средний уровень критичности, [50-75%] – высокий уровень критичности, [75- 100%] – критичный уровень) получаем, что первые три показателя обладают низкой критичностью, показатель процента проприетарных библиотек обладает высокой критичностью, показатель обработки ошибок низкой критичностью, показатель незащищенных данных средней критичностью.

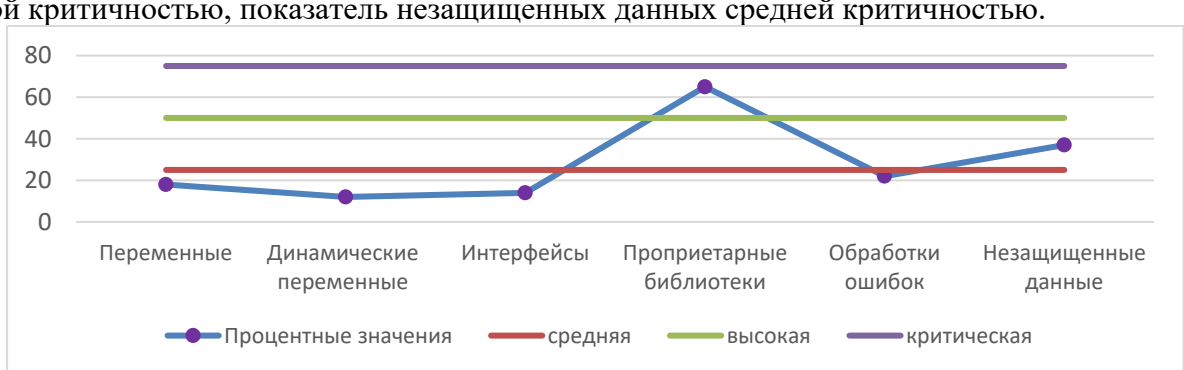

Рис. 1- Процентное соотношение уязвимостей
Fig. 1 - Percentage of vulnerabilities

Определим, что для нашей ИС куда большим потенциалом обладают уязвимости, которые могут быть достижимы посредством сетевой атаки (интерфейсы, незащищенные данные, проприетарные библиотеки). В таком случае имеющееся процентные значения уровня критичности снизим в целях повышения их приоритезации (например, на 15%). Получим (рис. 2).

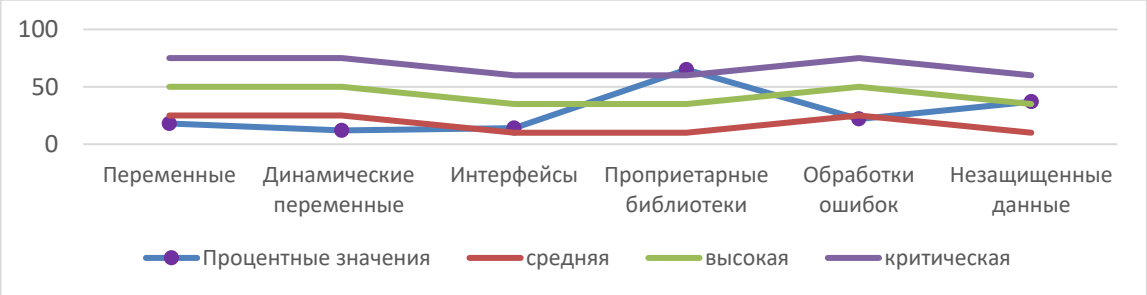

Рис. 2 - Процентное соотношение уязвимостей (приоритезация)
Fig. 2 - Percentage of vulnerabilities (prioritization)

При наличии достаточных статистических данных могут применяться более аккуратные значения приоритезации, либо иного ранжирования уровней критичности (с другим шагом) для уязвимостей различных типов.

Показатель уровня критичности должен быть согласован с действительной возможностью достижения уязвимостей того, либо иного типа, в зависимости от этого, для уязвимости должно быть присвоено значение критичности для каждой конкретной конфигурации. Также для оценки критичности уязвимостей всего продукта, согласно [11] сведем процентные показатели в одномерный массив:

$$V_{\text{П}} = [V_1 \quad V_2 \quad \dots \quad V_m], \tag{1}$$

где V_m – показатель критичности метрики уязвимости программы.

Наибольшее значение критичности будет отражать критичность возможных уязвимостей всего продукта.

$$V_{\text{П}_{\text{крит.}}} = \max [V_m]. \tag{2}$$

При этом качественная оценка дается в зависимости от приоритезации показателей.

Рассмотрим на нашем примере: $V_{\text{П}} = [18 \quad 12 \quad 14 \quad 65 \quad 25 \quad 35]$,

$$V_{\text{П}_{\text{крит.}}} = 65.$$

При рассмотрении с равным распределением критичностей, то получаем высокий уровень критичности. При приоритезации – критический. Также данный процесс возможно представить в виде UML-диаграммы, и алгоритма (рис. 3).

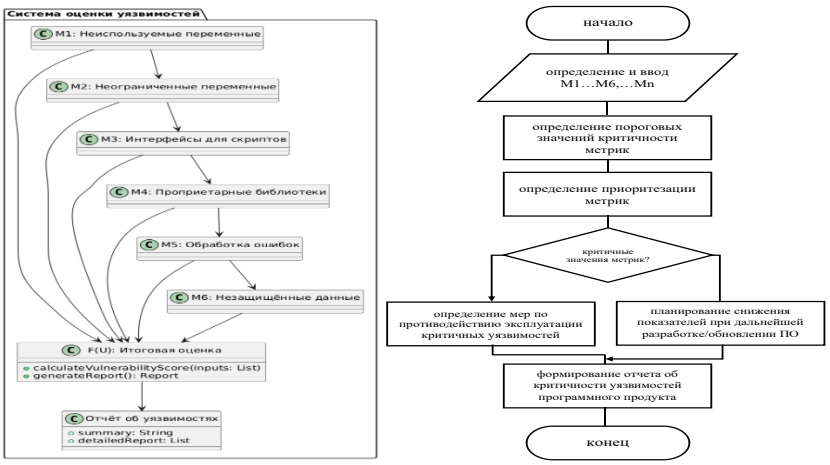

Рис. 3 - UML-диаграмма и алгоритм оценки уязвимостей
Fig. 3 - UML-diagram and vulnerability assessment algorithm

Вывод. В работе предложен подход к оценке уязвимостей автоматизированных систем, основанный на анализе ключевых метрик, влияющих на уровень защищённости программного обеспечения. В рамках исследования были определены и классифицированы метрики, отражающие наиболее значимые аспекты уязвимостей: наличие неиспользуемых переменных, динамических структур данных без ограничений, интерфейсов с возможностью выполнения скриптов, использование проприетарных библиотек, ограничение обработки ошибок и объём незащищённых данных. Дополнительная информация по предметной области представлена в [12-20].

Предложена модель, позволяющая установить взаимосвязь между перечисленными метриками и итоговым уровнем уязвимости, что способствует более точной количественной оценке защищенности информации и ИС. Интеграция результатов исследования позволит повысить точность оценки уровня защищённости автоматизированных систем, автоматизировать процесс выявления наиболее уязвимых компонентов программных продуктов в целях доработки; ускорить и повысить качество принятия решений по устранению выявляемых уязвимостей.

Библиографический список:

1. ГОСТ Р 58142-2018. Информационная безопасность. Методы и средства защиты информации. Общие положения. – Введ. 2018-12-01. – М.: Стандартинформ, 2018. – 24 с.
2. Дойникова, Е. В. Оценка защищенности компьютерных сетей на основе метрик CVSS / Е. В. Дойникова, А.А. Чечулин, И.В. Котенко // Информационно-управляющие системы. – 2017. – № 6(91). – С. 76–87. – DOI 10.15217/issn1684-8853.2017.6.76. – EDN ZXWUWH.
3. Дровникова, И.Г. Основные виды уязвимостей и взаимосвязь компонентов безопасности при обосновании показателей надёжности системы защиты информации от несанкционированного доступа в автоматизированных системах / И.Г. Дровникова, А.С. Етепнев, Е.А. Рогозин // Приборы и системы. Управление, контроль, диагностика. – 2019. – № 3. – С. 59–64. – EDN VWGOHY.
4. Щеглов, К.А. Защита от атак на уязвимости приложений. Модели контроля доступа / К.А. Щеглов, А.Ю. Щеглов // Вопросы защиты информации. – 2013. – № 2(101). – С. 36–43. – EDN QAVHRX.
5. Коноваленко, С.А. Выявление уязвимостей информационных систем посредством комбинированного метода анализа параметрических данных, определяемых системами мониторинга вычислительных сетей / С.А. Коноваленко, И.Д. Королев // Альманах современной науки и образования. – 2016. – № 11(113). – С. 60–66. – EDN XEEDXH.
6. Forum of Incident Response and Security Teams. Common Vulnerability Scoring System version 4.0: Specification Document [Электронный ресурс]. – URL: https://www.first.org/cvss/specification-document (дата обращения: 16.01.2025).
7. OWASP Foundation. OWASP Risk Rating Methodology [Электронный ресурс]. – URL: https://owasp.org/www-community/OWASP_Risk_Rating_Methodology (дата обращения: 16.01.2025).
8. Карты источников, содержащих сведения об уязвимостях программного обеспечения / А.Л. Сердечный, М.А. Тарелкин, А.А. Ломов, К.В. Симонов // Информация и безопасность. – 2019. – Т. 22, № 3. – С. 411–422. – EDN ZOUMGN.
9. ГОСТ Р 56546-2015. Информационная безопасность. Защита информации. Показатели защищённости информационных систем. – Введ. 2016-01-01. – М.: Стандартинформ, 2016. – 12 с.
10. ГОСТ Р 53114-2008. Информационная технология. Защита информации. Основные термины и определения. – Введ. 2009-01-01. – М.: Стандартинформ, 2009. – 14 с.
11. Концептуальные основы оценки уровня защищенности автоматизированных систем на основе их уязвимости /А.О. Ефимов, И.И. Лившиц, Т.В. Мещерякова, Е.А. Рогозин // Безопасность информационных технологий. – 2023. – Т. 30, № 2. – С. 63–79. – DOI 10.26583/bit.2023.2.04. – EDN LGPQZP.
12. Аветисян А.И., Белеванцев А.А., Чукляев И.И. Технологии статического и динамического анализа уязвимостей программного обеспечения. Вопросы кибербезопасности. 2014, № 3(4), с. 20–28. – EDN SSYPXV.
13. Russell R. et al. Automated Vulnerability Detection in Source Code Using Deep Representation Learning. 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA. 2018, p. 757–762. DOI: http://dx.doi.org/10.1109/ICMLA.2018.00120.
14. Wang T., Wei T., Gu G. and Zou W. TaintScope: A Checksum-Aware Directed Fuzzing Tool for Automatic Software Vulnerability Detection. IEEE Symposium on Security and Privacy, Oakland, CA, USA. 2010, p. 497–512. DOI: http://dx.doi.org/10.1109/SP.2010.37.
15. Lin G., Wen S., Han Q. -L., Zhang J. and Xiang Y. Software Vulnerability Detection Using Deep Neural Networks: A Survey in Proceedings of the IEEE. Oct. 2020, vol. 108, no. 10, p. 1825–1848. DOI: http://dx.doi.org/10.1109/JPROC.2020.2993293.

16. Кубарев, А.В. Подход к формализации уязвимостей информационных систем на основе их классификационных признаков / А.В. Кубарев // Вопросы кибербезопасности. – 2013. – № 2(2). – С. 29–33. – EDN SZEDHH.
17. Ланкин, О.В. Системно-комплексный кибернетический подход к формированию методологических основ интеллектуальной защиты информации от несанкционированного доступа / О.В. Ланкин, В.И. Сумин, Е.В. Воронова // Вестник Воронежского государственного технического университета. – 2011. – Т. 7, № 8. – С. 174–176. – EDN NYIJQT.
18. Лившиц И.И. Метод оценивания безопасности облачных ИТ-компонент по критериям существующих стандартов / И. И. Лившиц // Труды СПИИРАН. – 2020. – Т. 19, № 2. – С. 383–411. – DOI 10.15622/sp.2020.19.2.6. – EDN DPCIDQ.
19. Методика оценки уровня критичности уязвимостей программных, программно-аппаратных средств : утв. ФСТЭК России 28 октября 2022 г. : методический документ ФСТЭК России от 28.02.2022 г. URL: https://fstec.ru/dokumenty/vse-dokumenty/spetsialnye-normativnye-dokumenty/metodicheskij-dokument-ot-28-oktyabrya-2022-g-2 (дата обращения: 04.10.2024).
20. Методики оценивания надежности систем защиты информации от несанкционированного доступа в автоматизированных системах/О.И. Бокова, И.Г. Дровникова, А.С. Етепнев [и др.]//Труды СПИИРАН. – 2019. – Т. 18, № 6. – С. 1301–1332. – DOI 10.15622/sp.2019.18.6.1301-1332. – EDN YBHXOB.

References:

1. GOST R 58142-2018. Information Security. Methods and Tools for Information Protection. General Provisions. – Enacted on 2018-12-01. Moscow: Standartinform, 2018;24 p.(In Russ)
2. Doinikova, E.V. Evaluation of Computer Network Security Based on CVSS Metrics / E.V. Doinikova, A.A. Chechulin, I.V. Kotenko. *Information and Control Systems*. 2017;6(91):76–87. DOI: 10.15217/issn1684-8853.2017.6.76. – EDN ZXWUWH. (In Russ)
3. Drovnikova, I.G. Main Types of Vulnerabilities and the Relationship of Security Components in Justifying Indicators of Information Protection System Reliability Against Unauthorized Access in Automated Systems / I.G. Drovnikova, A.S. Etepnev, E.A. Rogozin. *Instruments and Systems: Monitoring, Control, and Diagnostics*. 2019; 3: 59–64. – EDN VWGOHY. (In Russ)
4. Shcheglov, K.A. Protection Against Application Vulnerability Attacks. Access Control Models/ K.A. Shcheglov, A.Yu. Shcheglov. *Information Security Issues.* 2013; 2(101):36–43. EDN QAVHRX. (In Russ)
5. Konovalenko, S.A. Detection of Information System Vulnerabilities Using a Combined Method for Analyzing Parametric Data Determined by Network Monitoring Systems / S.A. Konovalenko, I.D. Korolev. *Almanac of Modern Science and Education.* 2016;11(113): 60–66. – EDN XEEDXH. (In Russ)
6. Forum of Incident Response and Security Teams. Common Vulnerability Scoring System version 4.0: Specification Document [Electronic resource]. – URL: https://www.first.org/cvss/specification-document (accessed on 16.01.2025).
7. OWASP Foundation. OWASP Risk Rating Methodology [Electronic resource]. – URL: https://owasp.org/www-community/OWASP_Risk_Rating_Methodology (accessed on 16.01.2025).
8. Sources Containing Information on Software Vulnerabilities / A.L. Serdechny, M. A. Tarelkin, A.A. Lomov, K.V. Simonov. *Information and Security*. 2019; 22(3): 411–422. – EDN ZOUMGN. (In Russ)
9. GOST R 56546-2015. Information Security. Information Protection. Security Indicators of Information Systems. – Enacted on 2016-01-01. – Moscow: Standartinform, 2016;12 p. (In Russ)
10. GOST R 53114-2008. Information Technology. Information Protection. Basic Terms and Definitions. – Enacted on 2009-01-01. – Moscow: Standartinform, 2009; 14 p. (In Russ)
11. Conceptual Foundations for Assessing the Security Level of Automated Systems Based on Their Vulnerabilities / A.O. Efimov, I.I. Livshits, T.V. Meshcheryakova, E.A. Rogozin. *Information Technology Security.* 2023;30(2):63–79. – DOI: 10.26583/bit.2023.2.04. – EDN LGPQZP. (In Russ)
12. Avetisyan, A.I., Belevantsev, A.A., Chuklyaev, I.I. Technologies for Static and Dynamic Analysis of Software Vulnerabilities. *Cybersecurity Issues*. 2014;3(4): 20–28. – EDN SSYPXV. (In Russ)
13. Russell, R. et al. Automated Vulnerability Detection in Source Code Using Deep Representation Learning. 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA. 2018; p. 757–762. DOI: http://dx.doi.org/10.1109/ICMLA.2018.00120.
14. Wang, T., Wei, T., Gu, G., and Zou, W. TaintScope: A Checksum-Aware Directed Fuzzing Tool for Automatic Software Vulnerability Detection. IEEE Symposium on Security and Privacy, Oakland, CA, USA. 2010; p. 497–512. DOI: http://dx.doi.org/10.1109/SP.2010.37.
15. Lin, G., Wen, S., Han, Q. -L., Zhang, J., and Xiang, Y. Software Vulnerability Detection Using Deep Neural Networks: *A Survey in Proceedings of the IEEE*. Oct. 2020;108(10):1825–1848. DOI: http://dx.doi.org/10.1109/JPROC.2020.2993293.
16. Kubarev, A.V. Approach to Formalizing Information System Vulnerabilities Based on Their Classification Features / A.V. Kubarev. *Cybersecurity Issues*. 2013; 2(2): 29–33. – EDN SZEDHH. (In Russ)

17. Lankin, O.V. Systemic-Cybernetic Approach to Forming Methodological Foundations for Intelligent Protection of Information from Unauthorized Access / O.V. Lankin, V.I. Sumin, E.V. Voronova. *Bulletin of Voronezh State Technical University*. 2011; 7(8):174–176. – EDN NYIJQT. (In Russ)
18. Livshits, I.I. A Method for Evaluating the Security of Cloud IT Components Based on Existing Standards / I.I. Livshits. *SPIIRAS Proceedings*. 2020;19(2):383–411. DOI: 10.15622/sp.2020.19.2.6. – EDN DPCIDQ. (In Russ)
19. Methodology for Assessing the Criticality Level of Vulnerabilities in Software and Hardware-Software Tools: Approved by FSTEC of Russia on October 28, 2022: Methodological Document of FSTEC of Russia from October 28, 2022. URL: https://fstec.ru/dokumenty/vse-dokumenty/spetsialnye-normativnye-dokumenty/metodicheskij-dokument-ot-28-oktyabrya-2022-g-2 (accessed on 04.10.2024). (In Russ)
20. Methodologies for Assessing the Reliability of Information Protection Systems Against Unauthorized Access in Automated Systems / O.I. Bokova, I.G. Drovnikova, A.S. Etepnev [et al.]. *SPIIRAS Proceedings*. 2019;18(6):1301–1332. – DOI: 10.15622/sp.2019.18.6.1301-1332. – EDN YBHXOB. (In Russ)

Сведения об авторе:

Алексей Олегович Ефимов, преподаватель кафедры автоматизированных информационных систем органов внутренних дел; ea.aleksei@yandex.ru

Information about author:

Aleksey O. Efimov, Lecturer, Department of Automated Information Systems of Internal Affairs Bodies; ea.aleksei@yandex.ru

Конфликт интересов / Conflict of interest

Автор заявляет об отсутствии конфликта интересов/The author declare no conflict of interest.

Поступила в редакцию/Received 17.01.2025.

Одобрена после/рецензирования Reviced 01.03.2025.

Принята в печать/ Accepted for publication 29.05.2025.