УДК 004.41

ИССЛЕДОВАНИЕ УЯЗВИМОСТЕЙ В СОВРЕМЕННОМ ПРОГРАММНОМ ОБЕСПЕЧЕНИИ, ИСПОЛЬЗУЯ СТАТИСТИЧЕСКИЙ АНАЛИЗ

Попова О.Б., Кушнир Н.В., Носова Ю.С., Тотухов К.Е., Резниченко Л.С., Яцкевич Е.С.

ФГБОУ ВО «Кубанский государственный технологический университет», Краснодар, e-mail: popova_ob@mail.ru

Цель статьи показать, как влияют уязвимости на современное программное обеспечение, используя статистический анализ. Для этого был проведён анализ рисков и требований, изучены ограничения, произведён сбор данных и применён статистический анализ. Были получены выводы и предложения по дальнейшим исследованиям в области быстрой диагностики ошибок программного обеспечения. Уязвимости были классифицированы и проанализированы по типам ошибок и по уровню критичности CVSS. Актуальной задачей службы безопасности любой организации является защита от внешних и внутренних угроз. В ходе атак злоумышленники используют средства и методы для проникновения в инфраструктуру, закрепления в ней и сокрытие следов атак. Этапы атак осуществляются посредством эксплуатации как уже найденных специалистами по защите информации, но не исправленных в атакуемой инфраструктуре, так и с помощью необнаруженных уязвимостей, называемых «уязвимостями нулевого дня». Статистический анализ уязвимостей поможет специалистам по информационной безопасности и разработчикам программного обеспечения понять, какому тестированию, каких ошибок нужно уделять больше времени. Для уменьшения объема работ, выполняемого вручную, специалистами принято использовать автоматические средства сканирования. Но сканеры уязвимостей имеют издержки (false positives и false negatives), поэтому данные, полученные в ходе сканирования, необходимо перепроверять вручную.

Ключевые слова: защита информации, статистический анализ уязвимости, ошибки программного обеспечения, алгоритм структурирования знаний, дерево решений

VULNERABILITY STUDY IN MODERN SOFTWARE USING STATISTICAL ANALYSIS

Popova O.B., Kushnir N.V., Nosova Yu.S., Totukhov K.E., Reznichenko L.S., Yatskevich E.S.

Federal State Budgetary Educational Institution of Higher Education Kuban State Technological University, Krasnodar, e-mail: popova_ob@mail.ru

The purpose of the article is to show how vulnerabilities affect modern software using statistical analysis. For this, a risk and requirement analysis was carried out, constraints examined, data collected and statistical analysis applied. On the basis of which conclusions and proposals for further research were obtained. Vulnerabilities were analyzed by error type and CVSS severity level. The actual task of the security service of any organization is to protect against external and internal threats. During attacks, cybercriminals use tools and methods to penetrate the infrastructure, anchor in it and hide the traces of attacks. Stages of attacks are carried out by exploiting both those already found by information security specialists, but not fixed in the attacked infrastructure, and using undetected vulnerabilities called «zero-day vulnerabilities». Statistical analysis of vulnerabilities will help information security specialists and software developers understand which testing and which errors need to spend more time. To reduce the amount of manual work, it is common practice for specialists to use automatic scanning tools. But vulnerability scanners have costs (false positives and false negatives), so the data obtained during the scan must be rechecked manually.

Keywords: information protection, statistical analysis of vulnerability, software errors, knowledge structuring algorithm, decision tree

Нами была поставлена цель: исследовать уязвимости в современном программном обеспечении с помощью статистического анализа. Для этого был проведён анализ рисков и требований [1], изучены ограничения, произведён сбор данных [2] и применён статистический анализ. Далее были получены выводы по результатам исследования и перечислены предложения по научным исследованиям в области своевременного выявления ошибок программного обеспечения и защиты информации.

Статистический анализ уязвимостей

Актуальной задачей службы безопасности любой организации является защита от внешних и внутренних угроз. В ходе атак злоумышленники используют средства и методы для проникновения в инфраструктуру, закрепления в ней и сокрытие следов атак. Эти этапы атак осуществляются посредством эксплуатации как уже найденных специалистами по защите информации, но не исправленных в атакуемой инфраструктуре, так же и с помощью необнаруженных уязвимостей так называемых «уязвимостей нулевого дня». Статистический анализ уязвимостей поможет специалистам по информационной безопасности и разработчикам программного обеспечения понять тестированию каких ошибок нужно уделять больше времени. Для уменьшения

объема работ, выполняемого вручную, специалистами принято использовать автоматические средства сканирования. Но сканеры уязвимостей имеют издержки (false positives и false negatives), поэтому данные, полученные в ходе сканирования, необходимо перепроверять вручную. На основании ГОСТ Р 56546-2015 «Защита информации. Уязвимости информационных систем. Классификация уязвимостей информационных систем» [3], классификации ошибок программного обеспечения CWE (Common Weakness Enumeration) и принятых в отрасли и международном сообществе специалистов практик информационной безопасности разработан перечень типов ошибок программного обеспечения. Описание типов ошибок представлено в табл. 1. Уровень критичности и уязвимости определяются согласно международному стандарту для оценки уязвимостей – CVSS. Описание стандарта представлено в табл. 2.

Нами были использованы данные для исследования с сайта компании «Рашн Роботикс» [4]. Компания «Рашн Роботикс» предоставила данные с одного из автоматических сканеров веб-уязвимостей. Датасет, выбранный нами для исследования, представляет собой отчет о сканировании 22520 целей, в котором содержится информация о 80611 ошибках программного обеспечения.

Ошибки можно распределить по уровню критичности уязвимостей (табл. 3, рис. 1). Далее были рассмотрены уязвимости, уровень критичности которых выше низкого. Ошибки программного обеспечения можно распределить по типу (табл. 4).

Также в ходе более детального анализа уязвимостей высокого и критического уровня критичности было выявлено, что автоматический сканер уязвимостей имеет проблему ложных срабатываний (false positives) в 25 % случаев, так как использует эвристические алгоритмы анализа (табл. 5). Уровень false negatives установить не представляется возможным, так как для его подсчета необходим доступ непосредственно к сканируемой информационной системе.

Таблица 1

Типы ошибок программного обеспечения

№ п/п	Тип	Описание
1	Ошибка конфигурирования	Они связаны с неправильной настройкой параметров ПО.
2	Ошибка валидации данных	Они связаны с неполнотой проверки вводимых (входных) данных.
3	Ошибка путей доступа	Они связаны с возможностью прослеживания пути доступа к каталогам.
4	Ошибка использования внешних ссылок	Они связаны с возможностью перехода по ссылкам.
5	Ошибка исполнения команд	Они связаны с возможностью внедрения команд ОС.
6	Ошибка межсайтового скриптинга	Они связаны с межсайтовым скриптингом (выполнением сценариев).
7	Ошибка внедрения исходного кода	Они связаны с внедрением интерпретируемых операторов языков программирования или разметки.
8	Ошибка внедрения исполняемого кода	Они связаны с внедрением произвольного кода.
10	Ошибка переполнения памяти	Они связаны с переполнением буфера памяти.
11	Ошибка динамических параметров функций	Они связаны с неконтролируемой форматной строкой.
12	Ошибка некорректных вычислений	Они связаны с вычислениями:
		некорректный диапазон
		ошибка числа со знаком
		ошибка усечения числа
		ошибка индикации порядка байтов в числах
13	Ошибка раскрытия информации	Они связаны с утечками/раскрытием информации ограниченного доступа.
14	Ошибка повышения привилегий	Они связаны с управлением разрешениями, привилегиями и доступом.
15	Ошибка обхода аутентификации	Они связаны с аутентификацией.
16	Ошибка криптографической защиты	Они связаны с криптографическими преобразованиями (недостатки шифрования).

Окончание табл. 1

№ п/п	Тип	Описание
17	Ошибка подмены межсайтовых запросов	Они связаны с подменой межсайтовых запросов.
18	Ошибка вида «состояние гонки»	Они связаны с «состоянием гонки».
19	Ошибка управления ресурсами	Отражают недостатки, связанные с управлением ресурсами.
20	Ошибка политик разграничения доступа	Они связаны с управлением полномочиями (учетными данными).
21	Другая ошибка	

Таблица 2

Международный стандарт оценки уязвимостей

CVSS v2.0		CVSS v3.0	
Уровень угрозы	Диапазон баллов	Уровень угрозы	Диапазон баллов
Низкий	0.0–3.9	Отсутствует	0
		Низкий	0.1–3.9
Средний	4.0–6.9	Средний	4.0–6.9
Высокий	7.0–10.0	Высокий	7.0–8.9
		Критический	9.0–10.0

Таблица 3

Количество ошибок в датасете по уровню критичности

CVSS v2.0		CVSS v3.0		Количество ошибок в датасете
Уровень угрозы	Диапазон баллов	Уровень угрозы	Диапазон баллов	
Низкий	0.0–3.9	Отсутствует	0	39 759
		Низкий	0.1–3.9	23 137
Средний	4.0–6.9	Средний	4.0–6.9	17 009
Высокий	7.0–10.0	Высокий	7.0–8.9	698
		Критический	9.0–10.0	8

Рис. 1. Количество ошибок в датасете по уровню критичности

Таблица 4

Количество ошибок в датасете по типу ошибки

№	Тип	Количество ошибок в датасете
1	Ошибка конфигурирования	427
2	Ошибка валидации данных	50
3	Ошибка путей доступа	9
4	Ошибка использования внешних ссылок	474
5	Ошибка исполнения команд	8
6	Ошибка межсайтового скриптинга	200
7	Ошибка внедрения исходного кода	69
8	Ошибка внедрения исполняемого кода	4
10	Ошибка переполнения памяти	25
11	Ошибка динамических параметров функций	0
12	Ошибка некорректных вычислений	0
13	Ошибка раскрытия информации	134
14	Ошибка повышения привилегий	0
15	Ошибка обхода аутентификации	2
16	Ошибка криптографической защиты	14 744
17	Ошибка подмены межсайтовых запросов	0
18	Ошибка вида «состояние гонки»	0
19	Ошибка управления ресурсами	1517
20	Ошибка политик разграничения доступа	0
21	Другая ошибка	52

Таблица 5

Количество ошибок в датасете высокого и критического уровня

Уязвимость	Количество	Количество false positives
Cross-Origin Resource Sharing validation error	474	0
XSS	166	166
Remote code execution	4	4
Dos by long password	3	0
SQL injection	8	6
Ошибки связанные с SSL/TLS	23	0
CVE-2014-0133	25	0
CVE-2018-8719	2	2
Information disclosure	1	1
Итого:	706	179

Чтобы сократить объем работ, выполняемых специалистами по информационной безопасности без программного обеспечения, предлагается использовать дерево выбора, основанное на бинарном дереве системы вопросов и ответов [5], в сочетании с инструментами автоматического сканирования. Полученные в данной статье результаты исследования позволят правильно настроить бинарное дерево системы вопросов и ответов [5]. Для этого необходимо использовать алгоритм структурирования знаний предметной области «ошибки программного обеспечения», расположив их в дереве согласно их уровню критичности (рис. 2) [6]. Это поможет быстро выбирать наиболее подходящий метод решения проблемы, связанной с появлением ошибки в программном обеспечении. Ошибки с наивысшим уровнем критичности будут расположены на более высоком уровне дерева, что поможет их определять как можно быстрее.

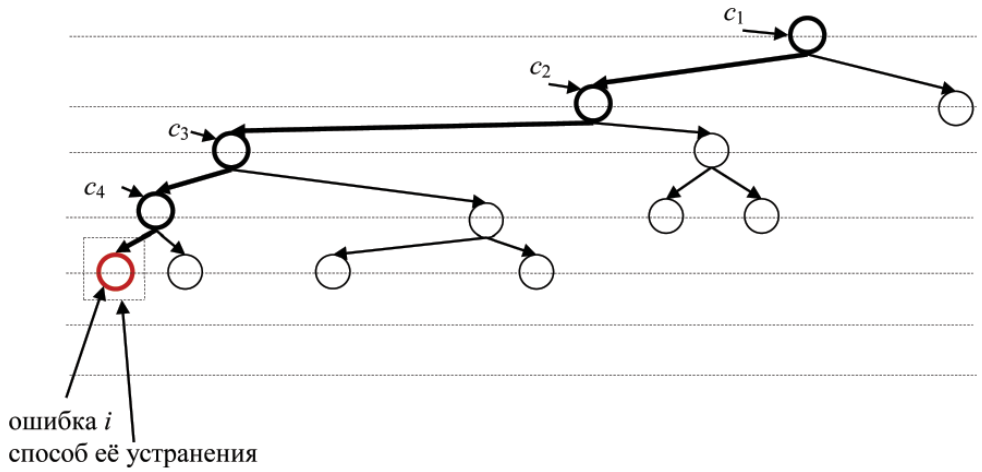

Рис. 2. Путь по дереву до ошибки i и её способа устранения, обладающей свойствами критичности c_1, c_2, c_3 и c_4

Далее приведём формулу представления знаний о новой потенциально возможной ошибке из предметной области «ошибки программного обеспечения» [6]:

НАЗВАНИЕ ОШИБКИ, **которая обладает свойствами** $\{x_i\}$, **отличающаяся тем, что** имеет свойство z_i, которое не соответствует свойствам $\{y'_i\}$ и $\{y''_i\}$,

где $\{x_i\}$ – полный путь по дереву или не полный путь по дереву к прототипу полученной ошибки;
$\{y'_i\}$ – множество свойств, которые расположены слева от указанного пути по дереву;
$\{y''_i\}$ – множество свойств, которые расположены справа от указанного пути по дереву.

Выводы

В процессе проведения исследования и анализа, полученных и внесённых в таблицу данных, было установлено следующее. При разработке программного обеспечения необходимо уделять большое внимание своевременному выявлению ошибок программного обеспечения, что способствует ликвидации различного уровня уязвимостей ещё на ранней стадии разработки и реализации программного обеспечения. Это позволяет добиться необходимого уровня информационной безопасности программного продукта. При этом следует заметить, что использование исключительно сканеров уязвимостей недопустимо. Учитывая масштаб современных киберпреступлений можно сделать вывод, что наряду с системным администратором в каждой организации необходимо иметь специалиста по информационной безопасности. Данный специалист должен обладать опытом фиксации разного рода возникающих ошибок, уметь использовать современный опыт структурирования ошибок программного обеспечения и иметь нужный вариант решения зафиксированной проблемы.

Исследование проведено при финансовой поддержке РФФИ. Название проекта: «Развитие теории качественной оценки информации с учётом её структурной составляющей», № 19-47-230004 от 19.04.2019 г.

Список литературы

1. CWE Version 4.2 // The MITRE Corporation. 2020. [Электронный ресурс]. URL: https://cwe.mitre.org/data/published/cwe_latest.pdf (дата обращения: 12.01.2021).

2. Common Vulnerability Scoring System v3.1: Specification Document. Forum of Incident Response and Security Teams. 2020. [Electronic resource]. URL: https://www.first.org/cvss/v3.1/specification-document (date of access: 12.01.2021).

3. ГОСТ Р 56546-2015. Защита информации. Уязвимости информационных систем. Классификация уязвимостей информационных систем. М.: Стандартинформ, 2018. 11 с.

4. Russian Robotics. 2020. [Electronic resource]. URL: https://www.rusrobots.ru (date of access: 12.01.2021).

5. Popova O., Popov B., Karandey V., Gerashchenko A. Entropy and Algorithm of Obtaining Decision Trees in a Way Approximated to the Natural Intelligence. International Journal of Cognitive Information and Natural Intelligence. 2019. Vol. 13. No. 3. P. 50–66.

6. Попова О.Б., Кушнир Н.В., Носова Ю.С., Тотухов К.Е. Уменьшение влияния когнитивного феномена скупости и когнитивных искажений на процесс выбора в поисковых исследованиях с помощью рефлективного подхода // Современные наукоемкие технологии. 2020. № 11–2. С. 305–312.