ИССЛЕДОВАНИЕ ПРИМЕНИМОСТИ ПРОЦЕССОВ И МЕР, ОБЕСПЕЧИВАЮЩИХ ИНФОРМАЦИОННУЮ БЕЗОПАСНОСТЬ СИСТЕМЫ С ГРАФОВОЙ СУБД

Карапетьянц Марк[1], Плаксий К.В.[2], Никифоров А.А.[3]

Аннотация

Цель: *Исследование популярных процессов и мер информационной безопасности в информационных системах с графовой СУБД и оценка их применимости с использованием инструментов сканирования уязвимостей и методов тестирования безопасности.*

Методы. *Теория графов, системный анализ, защита от инъекций, фильтрация ввода, Brute force.*

Результаты: *Выявлены основные угрозы и уязвимости для графовой СУБД. Проведённый анализ используемых процессов и мер защиты информации для SQL СУБД позволил определить перечень мер, наиболее подходящих для применения в графовых СУБД. В ходе исследования производилось тестирование безопасности Neo4j посредством использования перечня программных средств и утилит для раскрытия уязвимостей, которые в дальнейшем были устранены выявленными процессами и мерами защиты информации. В заключение проведена проверка и оценка защищенности комбинации средств защиты графовой СУБД. Полученные результаты имеют практическую значимость для различных информационных систем, внедряющих графовую СУБД в бизнес-процессы. Они также могут быть использованы для разработки основных критериев, необходимых при создании или улучшении графовых систем управления базами данных.*

Научная новизна. *Новизна исследования заключается в доказательстве применимости процессов и мер, обеспечивающих информационную безопасность информационной системы с графовой СУБД.*

Ключевые слова: *графовые СУБД, процессы и меры, информационная безопасность, Acunetix, Nmap, OWASP ZAP proxy, Burp Suite, Neo4j, угрозы, уязвимости, сканер уязвимости.*

DOI: 10.21681/2311-3456-2023-6-96-111

Введение

Базы данных NoSQL стали известны с 2009 года, и за последние несколько лет понятие «NoSQL» приобрело очень большую огласку по всему миру, это дало толчок к активному развитию и продвижению на рынке от разных производителей систем управления базами данных (СУБД). Первоначальное использование этих технологий стимулировало прогресс в развитии web-технологий и социальных сервисов. Это также привело к пересмотру множества подходов к хранению и обработке данных. На это развитие также повлияла проблема, связанная с использованием традиционных SQL СУБД, которые для поставленных задач были дорогостоящими или имели низкую производительность. Примером начала применения нереляционных СУБД является использование технологии Больших данных, которые на сегодняшний день имеют большую корреляцию с рассматриваемыми СУБД[4].

Одной из разновидностей NoSQL являются графовые СУБД, которые имеют немалую популярность среди пользователей, так как хранение в традиционных табличных формах не всегда отвечают их требованиям. NoSQL базы данных отличаются высокой произво-

4 Keith D. Foote. Graph Databases: An Overview. Datavetsity.2019 [Электронный ресурс] // Режим доступа к ресурсу: https://www.dataversity.net/graph-databases-an-overview (Дата обращения: 10.01.2023).

1 Карапетьянц Марк, аспирант Национального исследовательского ядерного университета «МИФИ», Москва, Россия. E-mail: Mkarapetyants@mephi.ru, ORCID: 0009-0002-3262-1138.

2 Плаксий Кирилл Владимирович, старший преподаватель Национального исследовательского ядерного университета «МИФИ», Москва, Россия. E-mail: KVPlaksii@mephi.ru, ORCID: 0000-0002-8949-6772.

3 Никифоров Андрей Александрович, старший преподаватель Национального исследовательского ядерного университета «МИФИ», Москва, Россия. E-mail: andreinikiforov993@gmail.com, ORCID: 0000-0002-2726-0000.

дительностью и скоростью. Распределенная архитектура СУБД обеспечивает простое масштабирование, позволяет автоматически распределять данные между несколькими серверами и повышает скорость чтения данных. На данный момент самым популярным решением в этой области является Neo4j, которая занимает первое место среди графовых СУБД (рис. 1).

Несмотря на то, что графовые СУБД способны обрабатывать большой поток информации, обладая такими свойствами, как гибкость, масштабируемость и производительность, появляется проблема, связанная с плохой безопасностью данных в них, что может негативно сказаться на деятельности компании [1].

Обеспечение информационной безопасности графовых СУБД начинается с определения и устранения существующих уязвимостей. Для ликвидации уязвимостей необходимо использование комплексного подхода к защите данных в графовых СУБД. Также для создания эффективной системы обеспечения ИБ СУБД необходимо оценить актуальные угрозы ИБ, которые существуют на сегодняшний день.

Данное исследование продолжает работу, начатую авторами в [2], и ставит своей целью дополнить основной перечень уязвимостей и угроз ИБ для графовых СУБД, применить выработанный перечень методов защиты данных, используемых для устранения выявленных уязвимостей, и провести тестирование безопасности графовой СУБД на примере Neo4j [3]. Для выполнения поставленной цели решаются следующие задачи: актуализировать список угроз и уязвимостей графовых СУБД, применить средства тестирования СУБД для поиска существующих уязвимостей Neo4j, использовать перечень процессов и мер защиты информации для устранения уязвимостей СУБД, провести повторное тестирование после внедрения процессов и мер для оценки защищенности системы.

1. Угрозы, уязвимости и методы защиты данных в графовых СУБД

Исследований в области уязвимостей и угроз для графовых СУБД не так много по той причине, что данные системы только набирают популярность в IT-отрасли. Но, основываясь на предыдущих работах авторов и на других трудах в этой области, был дополнен основной перечень угроз и уязвимостей (табл. 1) и подобраны к ним соответствующие меры для их устранения[5].

Для противодействия данным угрозам был определен актуальный перечень процессов и мер, которые позволяют устранить выявленные уязвимости в графовой СУБД Neo4j [4].

Rank Jan 2023	Rank Dec 2022	Rank Jan 2022	DBMS	Database Model	Score Jan 2023	Score Dec 2022	Score Jan 2022
1.	1.	1.	Neo4j	Graph	55.84	-1.49	-2.19
2.	2.	2.	Microsoft Azure Cosmos DB	Multi-model	37.96	+0.01	-2.08
3.	3.	3.	Virtuoso	Multi-model	5.88	-0.07	+0.50
4.	4.	4.	ArangoDB	Multi-model	5.07	-0.27	+0.34
5.	5.	5.	OrientDB	Multi-model	4.48	-0.09	-0.07
6.	↑ 7.	↑ 7.	Amazon Neptune	Multi-model	2.81	-0.09	+0.18
7.	↓ 6.	↑ 8.	JanusGraph	Graph	2.64	-0.35	+0.25
8.	8.	↓ 6.	GraphDB	Multi-model	2.53	+0.06	-0.33
9.	9.	9.	TigerGraph	Graph	2.20	+0.13	+0.18
10.	↑ 11.	↑ 11.	Dgraph	Graph	1.80	+0.08	+0.29
11.	↓ 10.	↑ 12.	Fauna	Multi-model	1.77	-0.11	+0.41
12.	12.	↓ 10.	Stardog	Multi-model	1.62	-0.04	-0.27
13.	↑ 14.	13.	Giraph	Graph	1.53	+0.13	+0.22
14.	↑ 16.	↑ 15.	NebulaGraph	Graph	1.51	+0.37	+0.37
15.	↓ 13.	↓ 14.	AllegroGraph	Multi-model	1.39	-0.03	+0.15
16.	↓ 15.	↑ 18.	TypeDB	Multi-model	1.34	+0.07	+0.58
17.	↑ 18.	↑ 20.	Memgraph	Graph	1.32	+0.27	+0.94
18.	↓ 17.	↓ 16.	Blazegraph	Multi-model	1.13	-0.01	+0.17
19.	19.	↓ 17.	Graph Engine	Multi-model	1.07	+0.06	+0.22
20.	20.	↓ 19.	InfiniteGraph	Graph	0.61	+0.06	+0.14

Рис.1. Рейтинг DB-engines графовых хранилищ [6]

5 Hostingdata. List of NoSQLdatabase management systems [Электронный ресурс] // Режим доступа к ресурсу: https://hostingdata.co.uk/nosql-database/ (Дата обращения: 09.01.2023).

6 DB-Engines Рейтинг графовых БД. URL: https://db-engines.com/en/ranking/graph+dbms (дата обращения: 15.10.2020).

Таблица 1

Меры устранения уязвимостей графовых СУБД

Угрозы	Уязвимости	Процессы и меры устранения уязвимостей
Угроза обхода некорректно настроенных механизмов аутентификации	Уязвимость в системе аутентификации	Использование средств разграничения доступа: — Active Directory, OpenLDAP на основе сетевых протоколов аутентификации LDAP (Lightweight Directory Access Protocol) и Kerberos; — аутентификация с помощью токенов; использование компонентов экосистемы Apache Hadoop.
Угроза использования механизмов авторизации для повышения привилегий	Уязвимость в системе авторизации	
Угроза несанкционированного создания учётной записи пользователя	Уязвимость, связанная с недостатками разграничения доступа	Использование внутренней системы разграничения доступа для пользователей
Угроза несанкционированного удаления защищаемой информации		
Угроза повышения привилегий		
Угроза несанкционированного копирования защищаемой информации	Нешифрованный текст	Использование средств шифрования: — алгоритм шифрования AES (Advanced Encryption Standard); — использование HTTPS для шифрования сетевого взаимодействия; — использование компонента экосистемы Apache Hadoop, Cloudera, обеспечивающего шифрование данных HDFS-файлов (Hadoop Distributed File System).
Угроза приведения системы в состояние «отказ в обслуживании»	Уязвимость переполнения буфера и отказа в обслуживании	Использование резервных копий или сторонних продуктов Apache Hadoop для хранения данных: — Распределенная между узлами вычислительного кластера файловая система HDFS (Hadoop Distributed File System); — MapReduce для распределенных операций предварительной обработки
Угроза несанкционированной модификации защищаемой информации	Инъекции в регулярных выражениях	— Проверка входных данных; — использование компонента экосистемы Apache Hadoop Native Auditing, журналов аудита периметра на шлюзе Knox, мониторинга запросов доступа, операций обработки и изменения данных; — ограничение использования регулярных выражений и REST-интерфейса.
Угроза межсайтового скриптинга	Инъекции кода, манипуляции с REST-интерфейсом	
Угроза межсайтовой подделки запроса		
Угроза внедрения кода или данных	Уязвимость контроля доступа к файлам СУБД	Использование внутреннего разграничения доступа в ОС путём присваивания прав neo4j, а также обеспечивая выполнение функций: чтение, изменение и запуск, только от имени «neo4j»
Угроза доступа к защищаемым файлам с использованием обходного пути		
Угроза несанкционированного доступа к аутентификационной информации		
Угроза удаления аутентификационной информации		
Угроза использования слабостей кодирования входных данных	Уязвимость программного кода	Поддержка постоянного обновления ПО, так как данная уязвимость образуется на этапе разработки ПО
Угроза исследования механизмов работы программы		

2. Инструменты тестирования безопасности графовой СУБД на примере Neo4j

Необходимость в проведении тестирования заключается в том, что нужно определить, где у системы имеются недостатки или какие-либо уязвимости в конфигурации безопасности. Результаты проведенного тестирования позволят в дальнейшем снизить риски и смягчить последствия нежелательного доступа к БД [5]. Регулярные проверки безопасности также необходимы для защиты конфиденциальных данных организации от злоумышленников.

Процесс тестирования безопасности включает в себя 4 этапа:

1. Подготовка среды;
2. Проведение теста;
3. Оценка результатов;
4. Точная отчетность.

Выделяют основные типы проведения тестирования БД:

— Тест на проникновение – это процесс имитации кибератаки на сеть, компьютерную систему или веб-приложение для обнаружения в них любых уязвимостей;
— Сканер уязвимостей – это использование программы для сканирования системы на наличие известных уязвимостей с целью их устранения и исправления;
— Аудит безопасности – это процесс оценки реализации и соответствия политик и стандартов безопасности организации;
— Оценка рисков – это процесс определения и анализа потенциальных угроз и возможных негативных последствий для достижения целей или выполнения задач.

На сегодняшний день для проверки безопасности в большинстве случаев используют инструменты, которые за счет быстрого выполнения своих задач существенно экономят время. Для проверки безопасности и проведения тестирования существуют разнообразные решения, включая как корпоративные, так и решения с открытым исходным кодом. Каждое из них предлагает свой уникальный набор функций, которые имеют различную специализацию и применимость. Это позволяет выявить ошибки и уязвимости в программном обеспечении перед его эксплуатацией [6].

В ходе исследования были использованы следующие варианты ПО:

— Burp Suite. Burp Suite — это интегрированная платформа для тестирования безопасности веб-приложений как в ручном, так и в автоматическом режимах. Программа кроссплатформенна и за счет наличия различных инструментов имеет возможность проводить процесс тестирования начиная от составления карты сайта до эксплуатации найденной уязвимости[7].
— Zed Attack Proxy. OWASP Zed Attack Proxy (ZAP) — это один из самых популярных бесплатных инструментов безопасности, он активно поддерживается сотнями волонтёров со всего мира. Он может помочь автоматически найти уязвимости безопасности в веб-приложениях во время разработки и тестирования. Этот инструмент также прекрасно подходит для опытных пентестеров, которые хотят проводить ручное тестирование безопасности.
— NMAP. Nmap или network mapper представляет собой набор инструментов функционального тестирования и тестирования на проникновение для всей сети, включая сканирование портов и обнаружение уязвимостей. Скрипты Nmap scripting engine (NSE) Script – одна из самых популярных и сильных возможностей Nmap. Данные скрипты сканирования уязвимостей Nmap имеют большую популярность у специалистов по тестированию на проникновение и злоумышленников для изучения общеизвестных уязвимостей.
— Acunetix. Acunetix (от Invicti) — это решение для сканирования кибербезопасности и веб-уязвимостей, предлагающее технологию автоматического тестирования веб-безопасности, которая позволяет организациям сканировать и проверять сложные, аутентифицированные веб-сайты с большим количеством HTML5 и JavaScript.

Выбор данных средств был сделан на основе доступности, результативности и надежности ПО. На сегодняшний день в связи с геополитической ситуацией в распоряжении у российских авторов находится небольшое количество инструментов для тестирования. При этом не существует полноценного теста безопасности гравофых СУБД, поэтому будут рассмотрены основные варианты, которые есть в открытых источниках.

3. Проверка безопасности Neo4j

На основе вышеописанных утилит и программ была проведена проверка безопасности графовой СУБД Neo4j без предварительного применения процессов и мер защиты информации. В тестах с Burp Suite происходит анализ защищенности графовой СУБД, кото-

7 SkillFactory. Burp Suite [Электронный ресурс] // Режим доступа к ресурсу: https://blog.skillfactory.ru/glossary/burp-suite/ (Дата обращения: 10.01.2023).

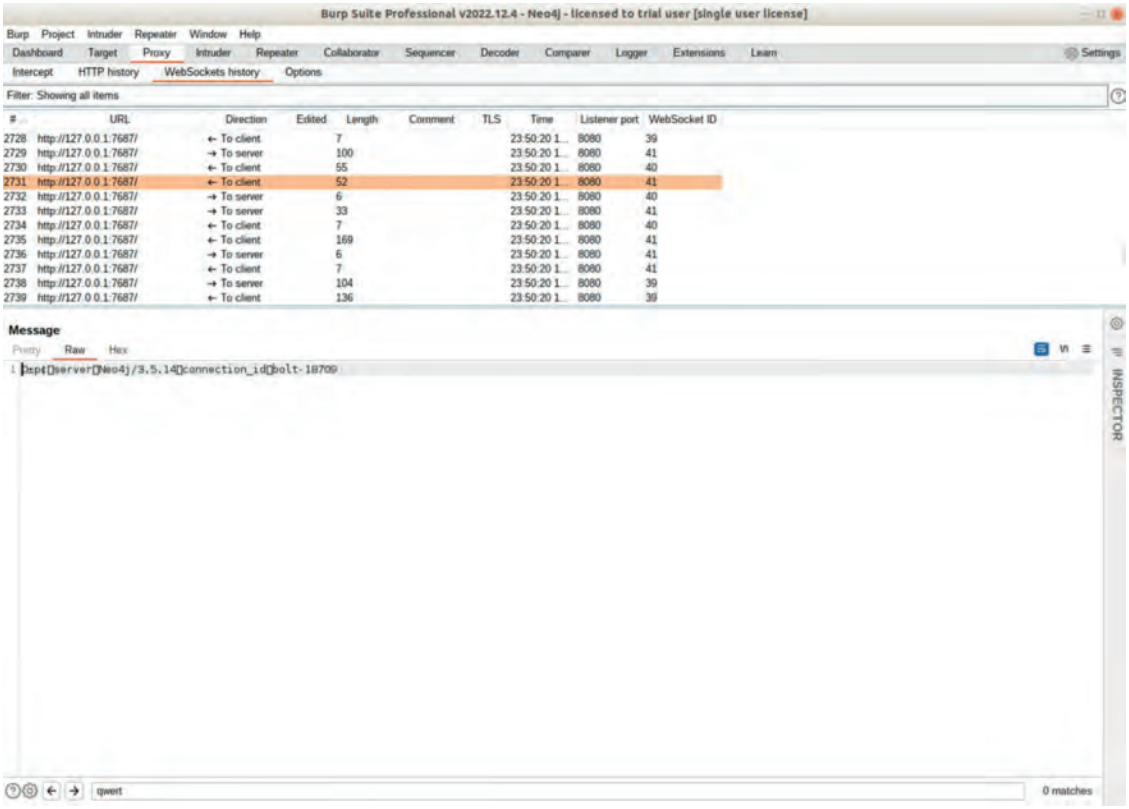

Рис.2. Отслеживание версии Neo4j

рая запускается по адресу «http://127.0.0.1:7474». Была поставлена задача выявить информацию, которая передается между клиентом и сервером. В большинстве веб-ресурсов плохо защищены поля ввода, из-за чего можно проводить атаку типа Brute force с помощью специальных утилит вроде Burp Suite. Но в данном случае провести такого типа атаку через интерфейс Neo4j не оказалось возможным [7].

При помощи утилиты удалось отследить передачу информации, отчетливо проследить запросы, которые передаются как серверу, так и к клиенту. В данном случае был произведен ввод корректной информации учетной записи, которая имеется в базе данных Neo4j. Удалось увидеть, как в открытом виде передаются данные логина и пароля. Это происходит по причине использования протокола http, который почти не шифрует данные при передаче. Помимо учетных данных, авторы выяснили версию установленной СУБД (рис.2), а также были перехвачены основные роли, которые присвоены пользователю.

С помощью программы Burp Suite Community можно отслеживать запросы, передаваемые между клиентом и сервером. Это может означать, что при изучении ИС злоумышленник может получить доступ к идентификационным данным пользователя, что позволит ему несанкционированно войти в систему.

Также с помощью Burp Suite можно сканировать веб-приложения на наличие уязвимостей. В ходе исследования сканер смог выявить 3 проблемы (рис. 3):

— Отправка пароля открытым текстом. Степень тяжести: высокая.
— Незашифрованные сообщения. Степень тяжести: низкая.
— Уязвимая зависимость библиотек JavaScript. Степень тяжести: низкая.

Следующим инструментом для проверки безопасности был взят OWASP ZAP [8] с возможностью проводить тестирование в режиме Атаки. Данная программа предлагает пользователю использовать как традиционный spider (инструмент, который предназначен для автоматического обнаружения новых URL-адресов на проверяемом сайте), так и ajax spider, который позволяет сканировать веб-приложения, написанные на AJAX. (рис. 4)

ZAP проводит полный перебор всевозможных полей на веб-приложении посредством использования spider. После завершения атаки происходит активное сканирование всего ресурса для обнаружения возможных уязвимостей. В случае проведенного сканирования были выявлены следующие проблемы в Neo4j:

— Content Security Policy (CSP): style-src небезопасный встроенный. Средняя степень тяжести.

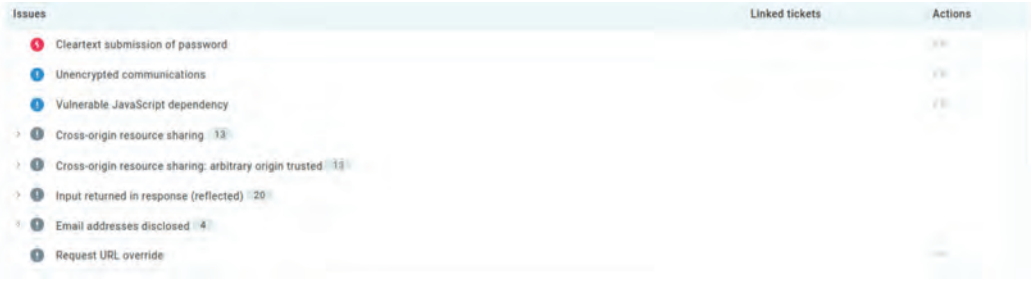

Рис.3. Результат сканирования Burp Suite

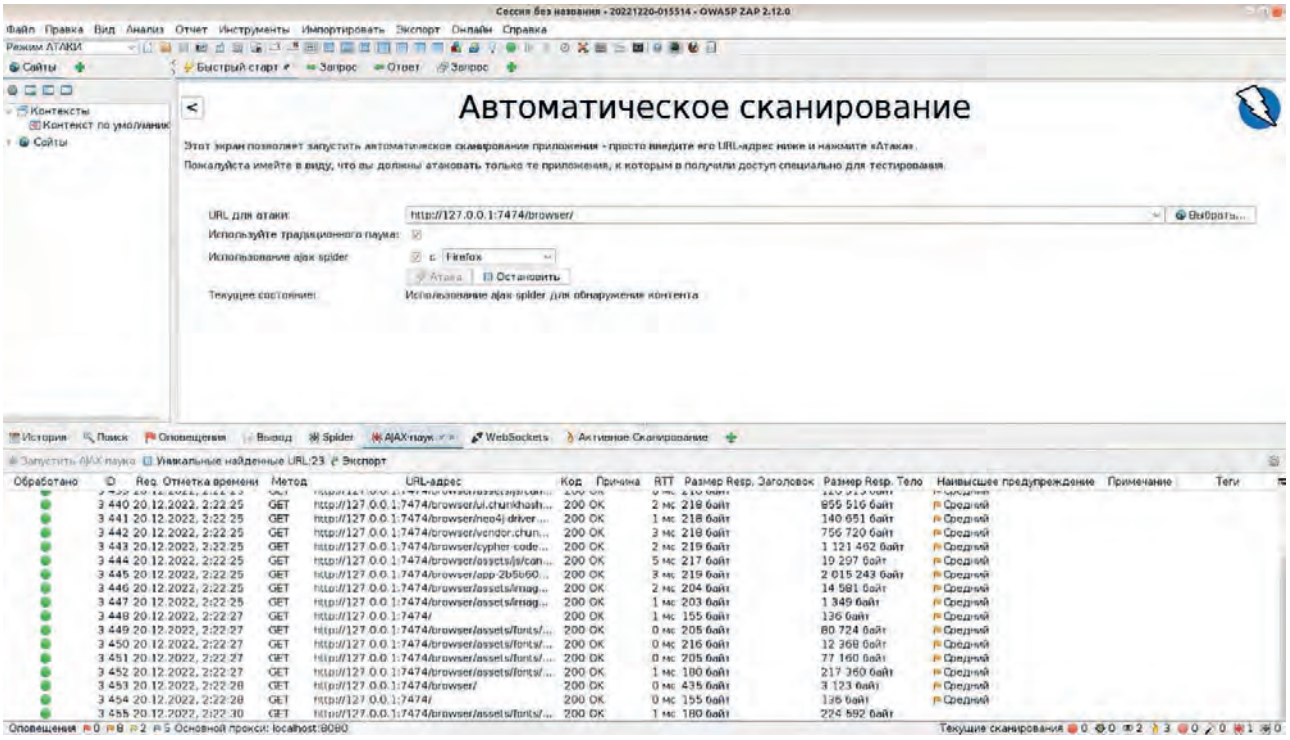

Рис.4. Результат сканирования ZAP

— CSP: Директива подстановочного знака. Средняя степень тяжести.
— CSP: скрипт-SRC небезопасный встроенный. Средняя степень тяжести.
— Заголовок CSP не задан. Средняя степень тяжести.
— Междоменная неправильная конфигурация. Средняя степень тяжести.
— Раскрытие ошибок приложения. Средняя степень тяжести.
— Недостаточно надежный метод. Средняя степень тяжести.
— Уязвимость JS Библиотеки. Средняя степень тяжести.
— Заголовок X-Content-Type-Options отсутствует. Низкая степень тяжести.
— Раскрытие отметки времени - Unix. Низкая степень тяжести.

Выявленные уязвимости в основном связаны с некорректной работой передачи данных между клиентом и сервером, а также с разметкой страницы, что может привести к негативным последствиям [9].

В ходе исследования был использован популярный сканер уязвимостей Acunetix. Данный сканер прост в использовании, достаточно ввести в качестве цели СУБД Neo4j по адресу «http://localhost:7474»[10]. После проведения сканирования системы Acunetix предоставляет полный перечень выявленных уязвимостей с возможностью выгрузить отчет (рис. 5). Вследствие проведения сканирования программа выявила следующие уязвимости:

— Базовая аутентификация через HTTP. Средняя степень тяжести.
— Clickjacking: отсутствует заголовок X-Frame-Options. Низкая степень тяжести.

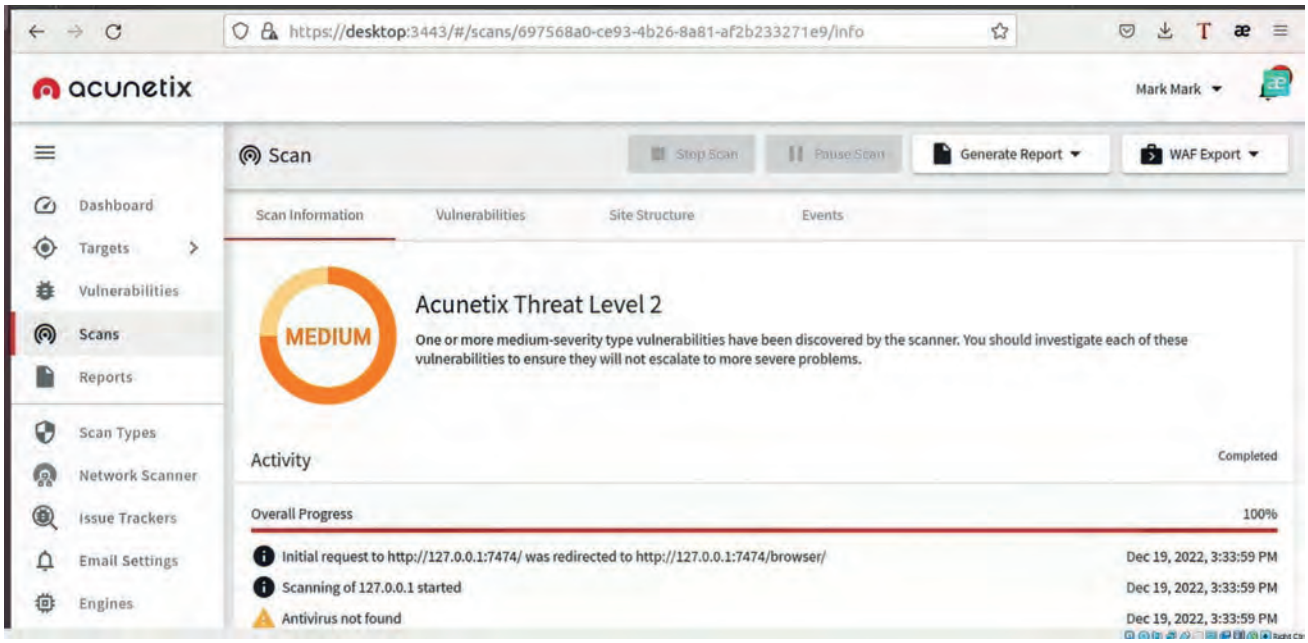

Рис.5. Результаты сканирования Acunetix

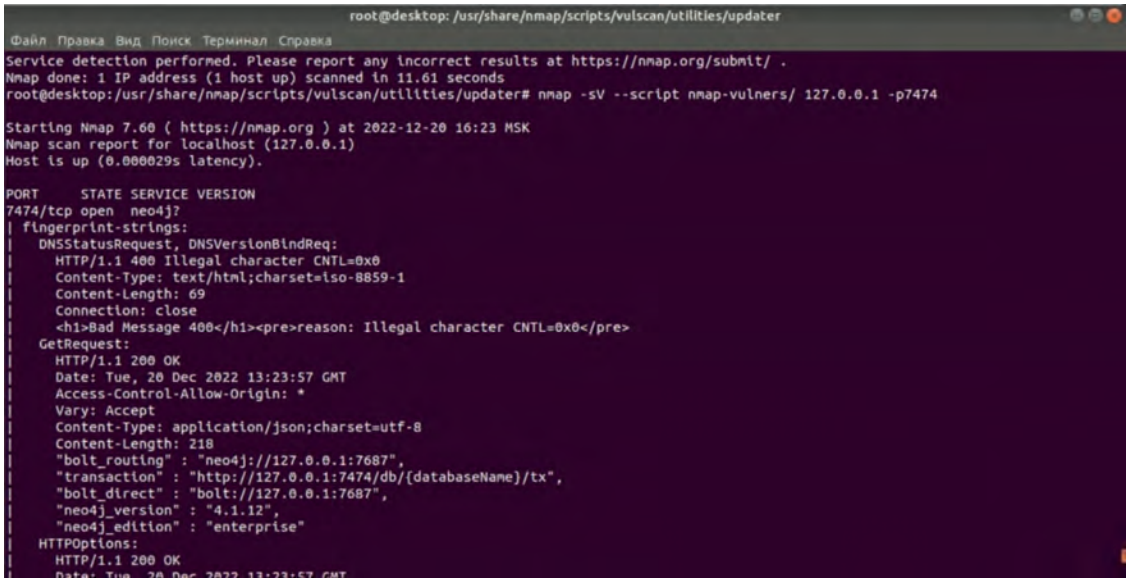

Рис.6. Результат сканирования скриптов Nmap

— Незашифрованное соединение. Низкая степень тяжести.
— Политика безопасности контента (CSP) не реализована. Низкая степень тяжести.

Были обнаружены аналогичные уязвимости, сходные с теми, которые были обнаружены при использовании предыдущих инструментов. Помимо вышеописанных программ в ходе исследования использовался встроенный инструмент для сканирования не только портов, но и уязвимостей Nmap. Самыми распространёнными и эффективными в поиске скриптами обнаружения уязвимостей являются Nmap-vulners, vulscan и vuln [11]. В основном они используют открытую базу знаний CVE. Данные скрипты позволяют исследовать и находить важную информацию об уязвимостях систем[8].

После запуска каждого из скриптов получен одинаковый результат, который не выявил критических уязвимостей кроме информации о версии СУБД Neo4j. Такие данные могут быть использованы злоумышлен-

8 Подвальчик Хакера. Основы Burp Suite. Что это и как им пользоваться [Электронный ресурс] // Режим доступа к ресурсу:https://hacker-basement.com/2021/01/11/osnovy-burp-suite/ (Дата обращения: 10.01.2023).

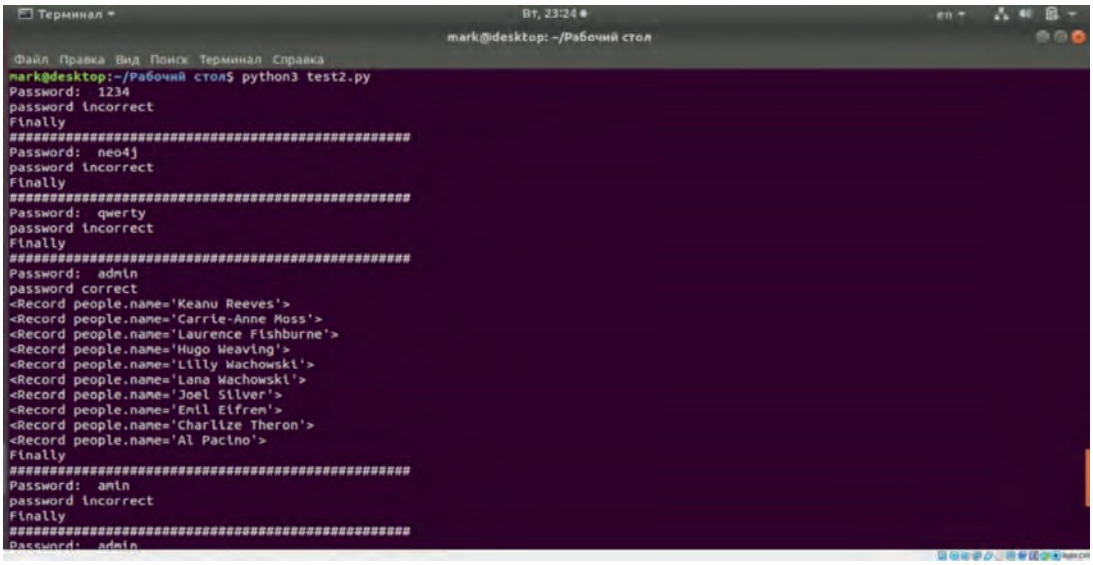

Рис.7. Результат программного кода

никами, которые в перспективе могут изучить уязвимости базы в открытых источниках. (рис.6)

В большинстве случаев ручное тестирование на безопасность проводится опытными специалистами для возможности определить уязвимости там, где не смог определить сканер, с учетом особенностей системы или ПО. Был рассмотрен вариант, когда злоумышленник может проникнуть в СУБД не только напрямую, но и косвенно через ОС, точнее через службы, которые в ней запущены.

Руководствуясь данным суждением, были выявлены следующие проблемы:

1. Доступ к файлам СУБД.

В процессе исследования были выявлены документы, которые либо не защищены, либо имеют недостаточно надежную систему защиты. Если основное устройство, на котором развернута система управления базами данных (СУБД), имеет парольную защиту, нет необходимости вводить пароль для чтения данных из файлов. Это указывает на низкую уровень безопасности системы. В качестве ценных файлов можно отметить следующие:

- Security.log. Журнал безопасности, который используется для отслеживания информации, связанной с безопасностью в компьютерной системе;
- Auth. Файл хранит аутентификационную информацию пользователя, зашифрованную посредством использования алгоритма SHA256;
- Roles. Файл хранит информацию о ролях пользователей.

2. Попытка проведения Brutforce-атаки.

В ходе исследования проведена атака по словарю. Данный метод реализован на языке программирования Python. Выбор сделан в его пользу по той причине, что остальные методы либо не реализуемы в нынешнем исследовании, либо не применимы, как, например, гибридный метод в связи со спецификой веб-сайта графовой СУБД Neo4j.

Реализовав программный код (рис. 7), можно наблюдать, что в случае некорректного ввода пароля выводится сообщение «password incorrect», в противном случае «password correct». Посредством перебора было рассмотрено 8 паролей, 6 из них вывели сообщение о неправильном пароле, а 2 вывели результат. В качестве результата были переданы первые 10 человек из БД, связанные с «Фильмом».

4. Применение процессов и мер обеспечения ИБ для графовой СУБД

В ходе проведения тестирования безопасности Neo4j выявлены уязвимости, которые нужно устранить. Для того, чтобы обеспечить надежную безопасность графовой СУБД, необходимо проверить применимость существующих процессов и мер СУБД посредством тестирования с помощью инструментов, описанных выше.

В основном, планируется использовать комбинацию ранее изученных средств, учитывая уже существующие процессы и меры обеспечения информационной безопасности реляционных баз данных, а также адаптированные методы защиты, применяемые в графовых СУБД [12].

1. Настройка конфигурации.

```
# Bolt connector
dbms.connector.bolt.enabled=true
dbms.connector.bolt.tls_level=REQUIRED
#dbms.connector.bolt.listen_address=:7687
#dbms.connector.bolt.advertised_address=:7687

# HTTP Connector. There can be zero or one HTTP connectors.
dbms.connector.http.enabled=true
#dbms.connector.http.listen_address=:7474
#dbms.connector.http.advertised_address=:7474

# HTTPS Connector. There can be zero or one HTTPS connectors.
dbms.connector.https.enabled=true
#dbms.connector.https.listen_address=:7473
#dbms.connector.https.advertised_address=:7473
```

Рис.8. Настройка защиты транспортного уровня

Основным файлом конфигурации СУБД Neo4j является neo4j.conf, который содержит в себе основные параметры конфигурации в Neo4j[9].

В качестве основных протоколов для безопасной передачи данных используют Bolt, который помогает обеспечить связь «клиент-сервер» в базе данных Neo4j, а также протокол https, позволяющий использовать сертификаты SSL для безопасной передачи данных[10].

Коннекторы настраиваются в формате dbms.connector.<connector-name>.<setting-suffix>> (рис. 8).

2. Разграничение доступа и защита полей.

Необходимой мерой для СУБД является определение ролей. Данная СУБД предоставляет встроенные роли и привилегии по умолчанию. Всего данных ролей 6: Public, Reader, Editor, Publisher, Architector, Admin.

Для обеспечения мер безопасности СУБД с использованием команды «drop» удален пользователь «neo4j». Перед проведением этой манипуляции заранее создан другой суперпользователь под именем «mark». Данное действие необходимо для того, чтобы злоумышленникам было труднее получить доступ в базу данных, используя атаку типа brute force.

3. Аутентификация пользователя.

У Neo4j есть собственный провайдер аутентификации, который хранит всю информацию о пользователях и ролях в system-базе. Аутентификация настраивается в конфигурационном файле при помощи параметра dbms.security.auth_enabled. По умолчанию аутентификация включена (рис. 9).

Дополнительно можно настроить параметры, такие как «dbms.security.auth_max_failed_attempts» и «dbms.security.auth_lock_time». В первом случае можно задать максимальное количество неудачных попыток при входе, что позволяет существенно помешать злоумышленнику подобрать учетные данные. Второй параметр позволяет настроить время блокировки после исчерпания количества неудачных попыток.

4. Выбор расширенной лицензии для получения поддержки и дополнительных мер безопасности, таких как журналирование.

Стоимость услуги является проблемой при использовании корпоративной лицензии, а лицензия сообщества не предоставляет основных параметров безопасности, таких как журнал. Это будет способствовать атакам и утечкам информации.

5. Ограничение диска.

Для обеспечения безопасности диска, где расположена графовая СУБД, необходимо определить корректный объем памяти. Посредством использования команды «neo4j-admin memrec» можно получить первоначальную рекомендацию о том, как распределить определенный объем памяти.

6. Использование журналов безопасности.

Neo4j предоставляет механизмы отслеживания и анализа состояния базы данных при помощи мониторинга. При этом есть возможность проверять выполняемые запросы[11].

9 Инструменты Kali Linux. Zaproxy [Электронный ресурс] // Режим доступа к ресурсу: https://kali.tools/?p=2299 (Дата обращения: 10.01.2023).

10 Anti-malware. Обзор Acunetix Premium 14, DAST-платформы контроля безопасности веб-приложений [Электронный ресурс] // Режим доступа к ресурсу: https://www.anti-malware.ru/reviews/Acunetix-Premium (Дата обращения: 10.01.2023).

11 Технологии баз данных и знаний. Система управления базы данных [Электронный ресурс] // Режим доступа к ресурсу: http://bseu.by/it/tohod/lekcii5_4.htm (Дата обращения: 10.01.2023).

```
# Whether requests to Neo4j are authenticated.
# To disable authentication, uncomment this line
dbms.security.auth_enabled=true
```

Рис.9. Настройка аутентификации

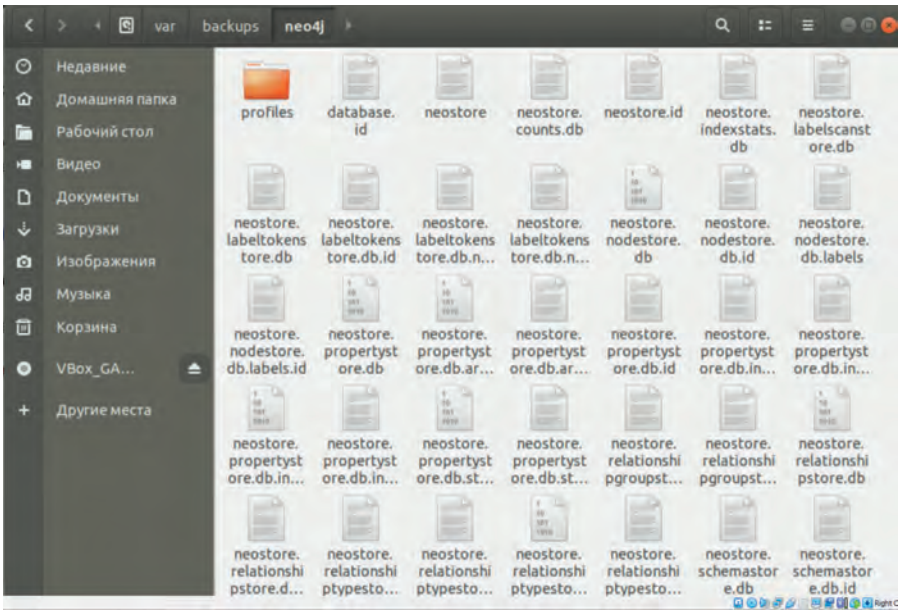

Рис.10. Директория резервной копии Neo4j

Все журналы, которые используются в графовой СУБД на дистрибутиве Linux, расположены по следующему адресу: «/var/log/neo4j». При этом настройка данных логов производится в параметре «dbms.directories.logs» [13].

В распоряжении администратора находятся 3 журнала, которые хранят в себе конфиденциальную информацию (табл. 2).

Таблица 2

Существующие журналы в графовой СУБД Neo4j

Название журнала	Описание
debug.log	Информация, полезная при отладке проблем с Neo4j.
query.log	Журнал выполненных запросов, которые занимают больше времени, чем указанный порог.
security.log	Журнал событий безопасности.

7. Резервное копирование.

В Neo4j резервная копия реализуется посредством использования команды «neo4j-admin backup». Она имеет дополнительные аргументы в виде директории сохранения и базы данных. Использование команды позволяет создать резервную копию в папке «var/backups» (рис. 10).

8. Защита передачи данных

В ходе исследования для создания собственного сертификата SSL была использована утилита mkcert. Основные параметры настройки SSL конфигурации находятся в «SSL policy configuration». В данном разделе большая часть полей закомментирована и не работает по умолчанию. Были указаны прямые ссылки на расположение открытого и закрытого ключей и при этом выставлены параметры «dbms.ssl.policy.bolt.enabled» в значении «true» [14] (рис. 11).

После внесения изменения и сохранения данных появилась возможность войти в интерфейс СУБД через защищенный канал связи по выделенному порту 7473. (рис. 12)

9. Разграничение прав доступа к файлам

Нельзя забывать, что СУБД не является отдельно работающим ПО. Доступ к основным файлам можно получить через командную строку ОС, на которой Neo4j расположена. Neo4j рекомендует[12] производить разграничения прав доступа на следующие директории:

— Conf
— Import
— Bin
— Lib

12 Neo4j. Default file locations URL: https://neo4j.com/docs/operations-manual/current/configuration/file-locations/ (accessed: 10.01.2023).

```
# Bolt SSL configuration
dbms.ssl.policy.bolt.enabled=true
dbms.ssl.policy.bolt.base_directory=/var/lib/neo4j/certificates/bolt
dbms.ssl.policy.bolt.private_key=/var/lib/neo4j/certificates/bolt/localhost-key.pem
dbms.ssl.policy.bolt.public_certificate=/var/lib/neo4j/certificates/bolt/localhost.pem
dbms.ssl.policy.bolt.client_auth=NONE

# Https SSL configuration
dbms.ssl.policy.https.enabled=true
dbms.ssl.policy.https.base_directory=/var/lib/neo4j/certificates/https
dbms.ssl.policy.https.private_key=/var/lib/neo4j/certificates/https/localhost-key.pem
dbms.ssl.policy.https.public_certificate=/var/lib/neo4j/certificates/https/localhost.pem
dbms.ssl.policy.https.client_auth=NONE
```

Рис.11. Настройка SSL конфигурации

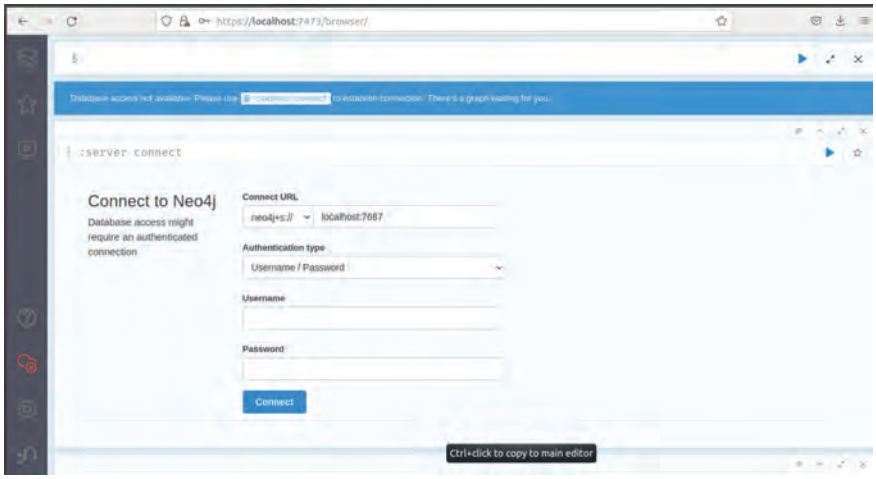

Рис.12. Подключение к Neo4j через протокол https

— Plugins
— Data
— Log

Для коррекции доступа к файлам Neo4j были использованы команды Linux «chmod» и «chown». Первая функция позволяет изменить права доступа к файлу, а вторая изменить владельца или группу. Необходимо, чтобы у всех важных файлов Neo4j были права чтения, изменения и запуска только для суперпользователя. При помощи команды «chmod 700 neo4j.conf» присваиваются права только пользовательскому администратору. Для передачи прав администратора neo4j нужно выполнить команду «chown neo4j:neo4j neo4j.conf».

5. Проверка защищенности методов защиты данных в Neo4j

Для определения эффективности предлагаемых методов защиты, требуется осуществить проверку защищенности графовой СУБД путем использования соответствующих процессов и мер, обеспечивающих информационную безопасность. Кроме того, необходимо повторно применить эти процессы и меры на защищенной СУБД с помощью специальных инструментов тестирования. Это позволит оценить степень применимости предлагаемых методов защиты.

Посредством использования инструмента Burp Suite была проведена попытка отследить передаваемые аутентификационные данные через клиент-серверное подключение. В данном случае инструмент не отслеживает передачу информации, то есть нельзя отследить данные, так как они передаются в зашифрованном виде, а при этом сам факт отслеживания попытки входа в графовую СУБД отслеживается. (рис. 13)

В итоге можно сказать, что инструмент Burp Suite не выявил критических уязвимостей у графовой СУБД после применения методов защиты.

Что касается проверки через Zed attack Proxy, то было проведено тестирование на безопасность с использованием данного инструментария, что позволило выявить 10 потенциальных уязвимостей. (рис. 14)

Если осуществлять сравнение проведенного сканирования с прошлым вариантом без применения методов защиты, то были выявлены те же уязвимости. В этом случае есть предположение, что предлагаемые процессы и меры не влияют на данные уязвимости,

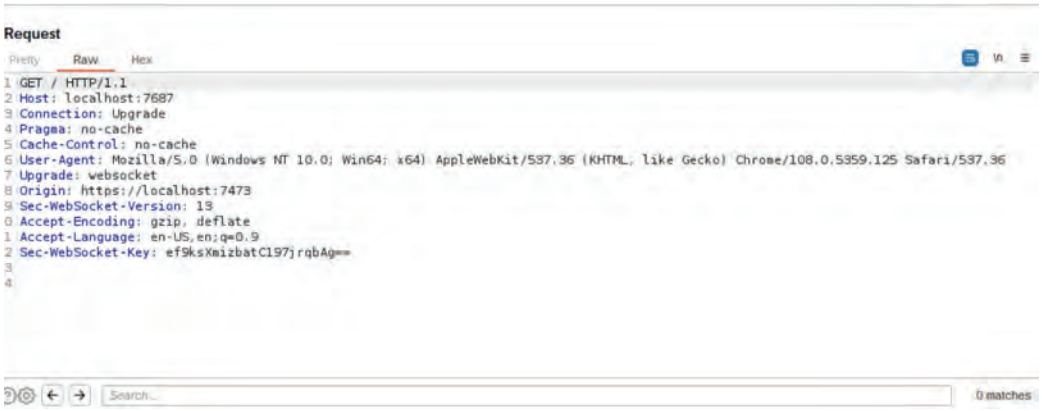

Рис.13. Отслеживание попытки входа в графовую СУБД

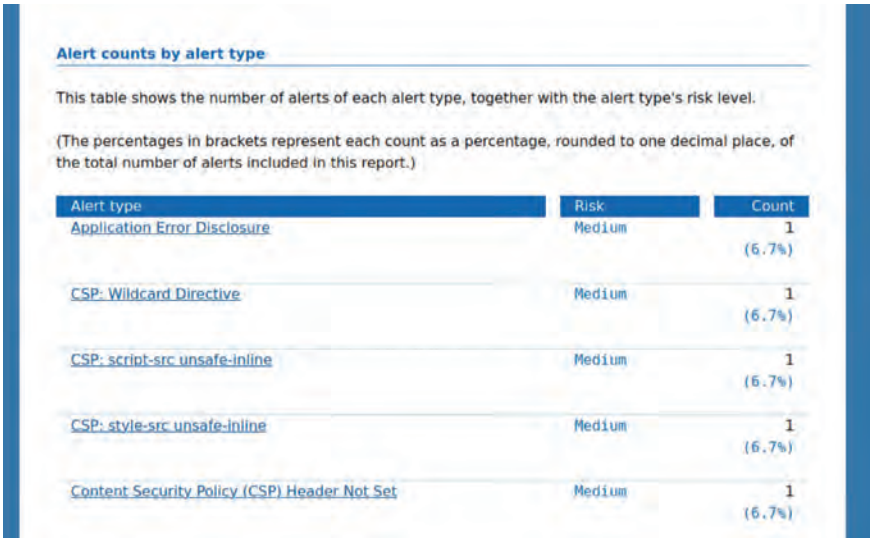

Рис.14. Отчет сканирования ZAP после применения методов защиты

при этом они могут быть связаны с внутренней архитектурой СУБД Neo4j, что необходимо исправлять разработчикам данного ПО.

В случае сканирования с помощью Acunetix необходимо было указать в инструменте в качестве цели тот же локальный хост (127.0.0.1), только другой порт 7473, так как СУБД в данном случае работает на порту протокола https [15]. После запуска сканера были обнаружены следующие результаты с указанием на одну выявленную уязвимость. (рис. 15)

Данная уязвимость заключается в отсутствии заголовка X-Frame-Options, отсутствие которого предоставляет возможность проведения атаки с исправлением поль зовательского интерфейса. При этом данная уязвимости имеет низкую степень тяжести последствий. Если сравнивать сканирования до и после применения методов защиты, то по сравнению с прошлым сканированием удалось избавиться от 4 уязвимостей из 5. Оставшаяся уязвимость, возможно, требует исправления на уровне разработки программного обеспечения.

Проведя сканирования цели в виде «127.0.0.1» с портом 7473 не удалось выявить уязвимостей у графовой СУБД Neo4j, что говорит о применимости предлагаемых методов защиты. При этом злоумышленник все равно может определить версию запущенной на АРМ СУБД Neo4j, что дает возможность определить актуальный перечень уязвимостей из открытых баз данных CVE.

Как говорилось ранее, для того от атаки типа Brute force, необходимо ограничить количество попыток входа для пользователя. Данная мера позволит снизить риск выявления учетных данных злоумышленником.

Запустив повторно программный код после применения предлагаемых методов можно увидеть (рис. 16), что в случае некорректного ввода пароля выводится сообщение «password incorrect», при этом после прохождения лимита попыток даже наличие правильного пароля не дает пользователю войти в СУБД. Посредством пере-

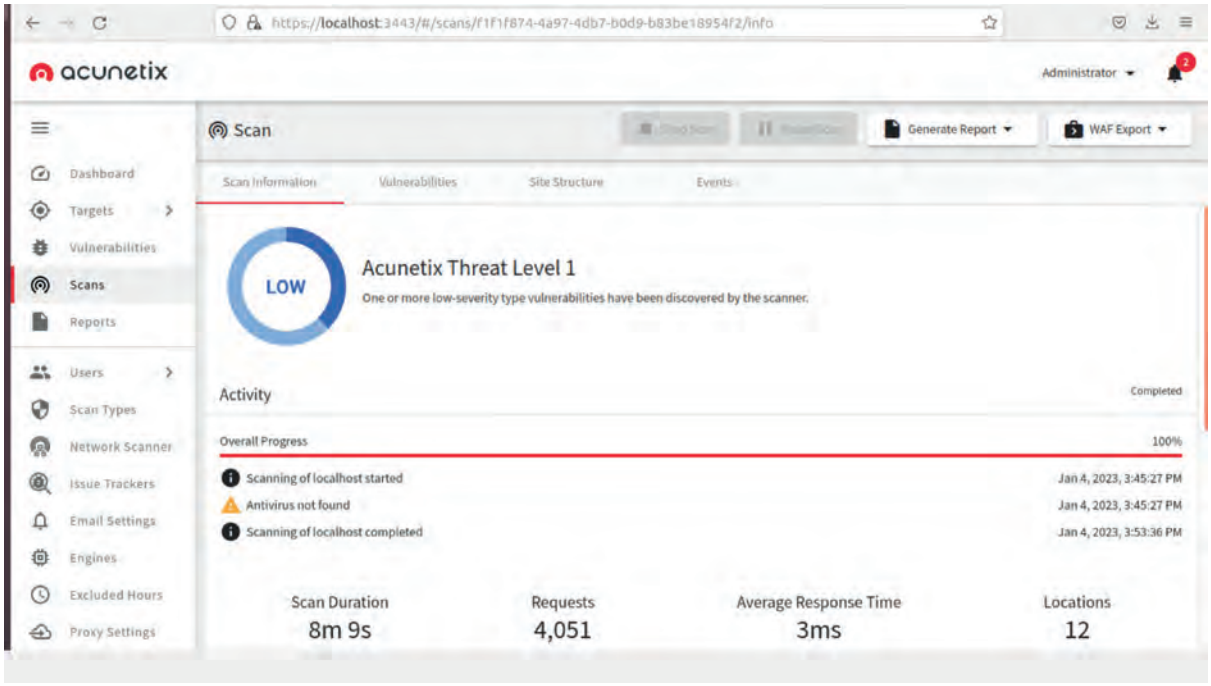

Рис.15. Результат сканирования Accunetix после применения методов защиты

```
ubuntu@ubuntu1804:~/Downloads$ python3 test2.py
Password:  1234
password incorrect
Finally
##################################################
Password:  neo4j
password incorrect
Finally
##################################################
Password:  qwerty
password incorrect
Finally
##################################################
Password:  admin
password incorrect
Finally
##################################################
Password:  amin
password incorrect
Finally
##################################################
Password:  admin
password incorrect
Finally
##################################################
Password:  dsfasdf
password incorrect
Finally
##################################################
Password:  fsadfvcxv
password incorrect
Finally
##################################################
ubuntu@ubuntu1804:~/Downloads$
```

Рис.16. Результат атаки типа Brute force после применения методов защиты

бора было рассмотрено восемь комбинаций, восемь из них вывели сообщение о неправильном пароле. В случае возникновения ошибки при входе, программа будет отображать некорректный пароль пользователя.

После применения предлагаемых процессов и мер защиты и проведения тестирования безопасности можно сделать вывод, что большая часть обнаруженных уязвимостей была успешно устранена.

Однако остальные уязвимости, которые не были представлены, могут быть связаны с архитектурой Neo4j и требуют вмешательства разработчика программного обеспечения для их решения.

Выводы

В ходе работы были рассмотрены угрозы, уязвимости и распространенные методы защиты данных для графовых СУБД. Определена применимость комбинации процессов и мер обеспечения информационной безопасности на примере Neo4j посредством проведения тестирования безопасности.

Полученные результаты позволят не только обеспечить ИБ графовых СУБД при внедрении в ИС, но и также помогут специалистам информационной безопасности составить общий перечень требований к безопасности СУБД данного типа, что обеспечит полное понимание при внедрении или разработке системы управления базы данных.

Благодаря использованию сочетания различных процессов и мер защиты большая часть выявленных уязвимостей была успешно исправлена, что подтверждает применимость выбранного набора методов защиты для графовой СУБД Neo4j. Это свидетельствует о том, что данные методы обеспечения безопасности могут быть эффективно применены в контексте работы с Neo4j и способны обеспечить необходимый уровень защиты информации, что доказало новизну работы. Однако следует продолжать обращать внимание на оставшиеся уязвимости и осуществлять необходимые меры для их устранения с участием разработчиков ПО.

Начальный перечень программного обеспечения, используемого для тестирования безопасности графовой СУБД, может быть расширен при интеграции в информационную систему, которая имеет свою собственную специфику деятельности. Каждая информационная система имееет свои уникальные требования и особенности, которые могут потребовать использования специализированного программного обеспечения для тестирования безопасности.

Литература

1. Sicari S., Rizzardi A., Coen-Porisini A. Security&privacy issues and challenges in NoSQL databases //Computer Networks. – 2022. – Т. 206. – С. 108828. DOI: 10.1016/j.comnet.2022.108828.
2. Плаксий К.В., Никифоров А.А., Милославская Н.Г. Исследование графовых СУБД, пригодных для работы с большими данными при обнаружении дел по отмыванию доходов, полученных преступным путем, и финансированию терроризма // Безопасность информационных технологий. – 2019. – Том 26, № 3. – С. 103-116. DOI: 10.26583/bit.2019.3.09.
3. Агафонов А. А. и др. Безопасность систем баз данных //Самара: Изд-во Самар. ун-та. – 2023. – Т. 1.
4. Плаксий К.В., Никифоров А.А., Милославская Н.Г., Кулагина Л.Л. Исследование вопросов обеспечения информационной безопасности графовых СУБД, пригодных для работы с большими данными, при обнаружении дел по отмыванию доходов, полученных преступным путем, и финансированию терроризма // Безопасность информационных технологий. – 2020. Том 27, № 4. – С. 53-64. DOI: 10.26583/bit.2020.4.05
5. Dissanayaka A. M. et al. Security assurance of MongoDB in singularity LXCs: an elastic and convenient testbed using Linux containers to explore vulnerabilities //Cluster Computing. – 2020. – Т. 23. – С. 1955-1971. DOI: 10.1007/s10586-020-03154-7
6. Макаренко С. И., Смирнов Г. Е. Анализ стандартов и методик тестирования на проникновение // Системы управления, связи и безопасности. – 2020. – №. 4. – С. 44-72. DOI: 10.24411/2410-9916-2020-10402
7. Kore A. et al. Burp Suite Extension for Script based Attacks for Web Applications //2022 6th International Conference on Electronics, Communication and Aerospace Technology. – IEEE, 2022. – С. 651-657. DOI: 10.1109/ICECA55336.2022.10009116
8. Abdullah H. S. Evaluation of open source web application vulnerability scanners //Academic Journal of Nawroz University. – 2020. – Т. 9. – №. 1. – С. 47-52. DOI: 10.25007/ajnu.v9n1a532
9. Devi R. S., Kumar M. M. Testing for security weakness of web applications using ethical hacking //2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184). – IEEE, 2020. – С. 354-361. DOI: 10.1109/ICOEI48184.2020.9143018
10. Saputra I. P., Utami E., Muhammad A. H. Comparison of anomaly based and signature based methods in detection of scanning vulnerability //2022 9th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI). – IEEE, 2022. – С. 221-225. DOI: 10.23919/EECSI56542.2022.9946485
11. Putra S. A., Budiono A., Hediyanto U. Y. K. S. Vulnerability Assesment Web Proposal Tugas Akhir Mahasiswa Menggunakan Acunetix dan NMAP //eProceedings of Engineering. – 2023. – Т. 10. – №. 2.
12. Кучкин В. П. Методы защиты баз данных // Проблемы науки. – 2021. – №. 4 (63). – С. 33-35.
13. Fahd K., Venkatraman S., Hammeed F. K. A comparative study of NoSQL system vulnerabilities with big data //Int. J. Manag. Inf. Technol. – 2019. – Т. 11. – №. 4. – С. 1-19. DOI: 10.5121/ijmit.2019.11401
14. Ankomah E. et al. A Comparative Analysis of Security Features and Concerns in NoSQL Databases // International Conference on Frontiers in Cyber Security. – Singapore : Springer Nature Singapore, 2022. – С. 349-364. DOI: 10.1007/978-981-19-8445-7_22
15. Zirwan A. Pengujian dan Analisis Keamanan Website Menggunakan Acunetix Vulnerability Scanner // Jurnal Informasi dan Teknologi. – 2022. – С. 70-75. DOI: 10.37034/jidt.v4i1.190

INVESTIGATION OF PROCESSES AND MEASURES APPLICABLE FOR ENSURING INFORMATION SECURITY FOR SYSTEMS WITH A GRAPHIC DBMS

Karapetyants Mark[13], Plaksiy K.V.[14], Nikiforov A.A.[15]

Purpose of the paper: *research of popular information security processes and measures in information systems with graph DBMS and assessment of their applicability using vulnerability scanning tools and security testing methods.*

Methods: graph theory, system analysis, injection protection, input filtering, Brute force.

Results: *the main threats and vulnerabilities for graph DBMS have been identified. The analysis of information security processes and measures involved in SQL DBMS allowed the authors to determine a list of measures most suitable for use in graph DBMS. During the study the researchers tested Neo4j's security with help of software tools and utilities to identify vulnerabilities which were subsequently eliminated by information security processes and measures. Finally, the investigators checked and assessed security of graph DBMS's security tools combination. The results obtained have practical significance for various information systems that implement graph DBMS in business processes. They can also be used to develop basic criteria needed when creating or improving graph database management systems.*

Scientific novelty: *the novelty of the research lies in proof of processes' and measures' applicability that ensure information security of an information system with a graph DBMS.*

Keywords: *graph DBMS, processes and measures, information security, Acunetix, Nmap, OWASP ZAP proxy, Burp Suite, Neo4j, threats, vulnerabilities, vulnerability scanner*

References

1. Sicari S., Rizzardi A., Coen-Porisini A. Security&privacy issues and challenges in NoSQL databases //Computer Networks. – 2022. – Vol. 206. – pp. 108828. DOI: 10.1016/j.comnet.2022.108828.
2. K.V. Plaksij, A.A. Nikiforov, N.G. Miloslavskaya. Issledovanie grafovyh SUBD, prigodnyh dlya raboty s bol'shimi dannymi pri obnaruzhenii del po otmyvaniyu dohodov, poluchennyh prestupnym putem, i finansirovaniyu terrorizma // Bezopasnost informacionnyh tekhnologij. – 2019. – Vol. 26, № 3. – pp. 103-116. DOI: 10.26583/bit.2019.3.09.
3. Agafonov A. A. i dr. Bezopasnost sistem baz dannyh //Samara: Izd-vo Samar. un-ta. – 2023. – Vol. 1.
4. K.V. Plaksij, A.A. Nikiforov, N.G. Miloslavskaya, L. L. Kulagina. Issledovanie voprosov obespecheniya informacionnoj bezopasnosti grafovyh SUBD, prigodnyh dlya raboty s bol'shimi dannymi, pri obnaruzhenii del po otmyvaniyu dohodov, poluchennyh prestupnym putem, i finansirovaniyu terrorizma. // Bezopasnost informacionnyh tekhnologij. – 2020. Vol. 27, № 4. – pp. 53-64. DOI: 10.26583/bit.2020.4.05
5. Dissanayaka A. M. et al. Security assurance of MongoDB in singularity LXCs: an elastic and convenient testbed using Linux containers to explore vulnerabilities //Cluster Computing. – 2020. – T. 23. – C. 1955-1971. DOI: 10.1007/s10586-020-03154-7
6. Makarenko S. I., Smirnov G. E. Analiz standartov i metodik testirovaniya na proniknovenie //Sistemy upravleniya, svyazi i bezopasnosti. – 2020. – №. 4. – pp. 44-72. DOI: 10.24411/2410-9916-2020-10402
7. Kore A. et al. Burp Suite Extension for Script based Attacks for Web Applications //2022 6th International Conference on Electronics, Communication and Aerospace Technology. – IEEE, 2022. – C. 651-657. DOI: 10.1109/ICECA55336.2022.10009116
8. Abdullah H. S. Evaluation of open source web application vulnerability scanners //Academic Journal of Nawroz University. – 2020. – T. 9. – №. 1. – C. 47-52. DOI: 10.25007/ajnu.v9n1a532

13 Mark Karapetyants, Ph.D. student, National Research Nuclear University MEPhI, Moscow, Russia. E-mail: Mkarapetyants@mephi.ru, ORCID: 0009-0002-3262-1138.

14 Kirill V. Plaksiy, Senior Lecturer, National Research Nuclear University MEPhI, Moscow, Russia. E-mail: KVPlaksii@mephi.ru, ORCID: 0000-0002-8949-6772.

15 Andrey A. Nikiforov, Senior Lecturer, National Research Nuclear University MEPhI, Moscow,Россия. E-mail: andreinikiforov993@gmail.com,http://orcid.org/0000-0002-2726-0000

9. Devi R. S., Kumar M. M. Testing for security weakness of web applications using ethical hacking //2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184). – IEEE, 2020. – С. 354-361. DOI: 10.1109/ICOEI48184.2020.9143018
10. Saputra I. P., Utami E., Muhammad A. H. Comparison of anomaly based and signature based methods in detection of scanning vulnerability //2022 9th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI). – IEEE, 2022. – С. 221-225. DOI: 10.23919/EECSI56542.2022.9946485
11. Putra S. A., Budiono A., Hediyanto U. Y. K. S. Vulnerability Assesment Web Proposal Tugas Akhir Mahasiswa Menggunakan Acunetix dan NMAP //eProceedings of Engineering. – 2023. – Т. 10. – №. 2.
12. Kuchkin V. P. Metody zashchity baz dannyh //Problemy nauki. – 2021. – №. 4 (63). – pp. 33-35.
13. Fahd K., Venkatraman S., Hammeed F. K. A comparative study of NoSQL system vulnerabilities with big data //Int. J. Manag. Inf. Technol. – 2019. – Т. 11. – №. 4. – С. 1-19. DOI: 10.5121/ijmit.2019.11401
14. Ankomah E. et al. A Comparative Analysis of Security Features and Concerns in NoSQL Databases //International Conference on Frontiers in Cyber Security. – Singapore : Springer Nature Singapore, 2022. – С. 349-364. DOI: 10.1007/978-981-19-8445-7_22
15. Zirwan A. Pengujian dan Analisis Keamanan Website Menggunakan Acunetix Vulnerability Scanner //Jurnal Informasi dan Teknologi. – 2022. – С. 70-75. DOI: 10.37034/jidt.v4i1.190