Инженерный вестник Дона, №2 (2025)
ivdon.ru/ru/magazine/archive/n2y2025/9868

О реализации информационной защищенности системы распределенного хранения данных малого бизнеса

А. С. Большаков, А. С. Добряков, Р. Р. Туктаров
Московский Технический Университет Связи и Информатики (МТУСИ)

Аннотация: В статье рассматриваются основные аспекты и рекомендации реализации защиты данных системы распределенного хранения данных для малого бизнеса. Рассматриваются методы обеспечения информационной безопасности, включая контроль инцидентов, двухфакторную аутентификацию и шифрование файлов. В ходе работы проведены тесты на отказоустойчивость системы и стойкость аутентификации с имитацией DoS и брутфорс атак с применением фаззинга. Предложены новые методы интеграции платформ для мониторинга инцидентов (MISP, Wazuh) и использование паролей на основе времени для двухфакторной аутентификации. Рассмотрены механизмы шифрования данных и управления доступом с использованием токенов доступа.
Ключевые слова: информационная безопасность, фаззинг, мониторинг, межсетевой экран, система хранения данных, шифрование данных, двухфакторная аутентификация, малый бизнес, отказоустойчивость.

Введение

В современном мире, где информация является одним из ключевых активов любого бизнеса, вопросы информационной безопасности становятся все более актуальными. Для малого бизнеса [1], который часто сталкивается с ограниченными ресурсами и бюджетом, реализация эффективного контроля информационной защищенности согласно принятым политикам информационной безопасности, должна полагаться на собственные силы пользователя/владельца системы распределённого хранения данных (далее СРХД). Система хранения данных является одним из наиболее ценных активов для любого бизнеса, в том числе, для малых предприятий при принятии обоснованных решений, улучшения обслуживания клиентов и оптимизации своей деятельности. Однако хранение данных также сопряжено с определенными обязанностями и рисками, такими, как обеспечение их безопасности, конфиденциальности и доступности. Предприятие может собирать и хранить различные данные, как о своих пользователях, так и внутренние данные о работе бизнеса. Они могут включать в себя статистику,

различную аналитику рынка, использование ресурсов и услуг бизнеса, инвентаризацию, и другие важные документы. В данной работе особое внимание уделено обеспечению безопасности данных, а также реализации аутентификации доступа в СРХД с целью минимизации возможности несанкционированного проникновения злоумышленника в систему.

В статье рассматривается СРХД с подсистемой контроля информационной защищенности для малого бизнеса, которая показана на рис.1.

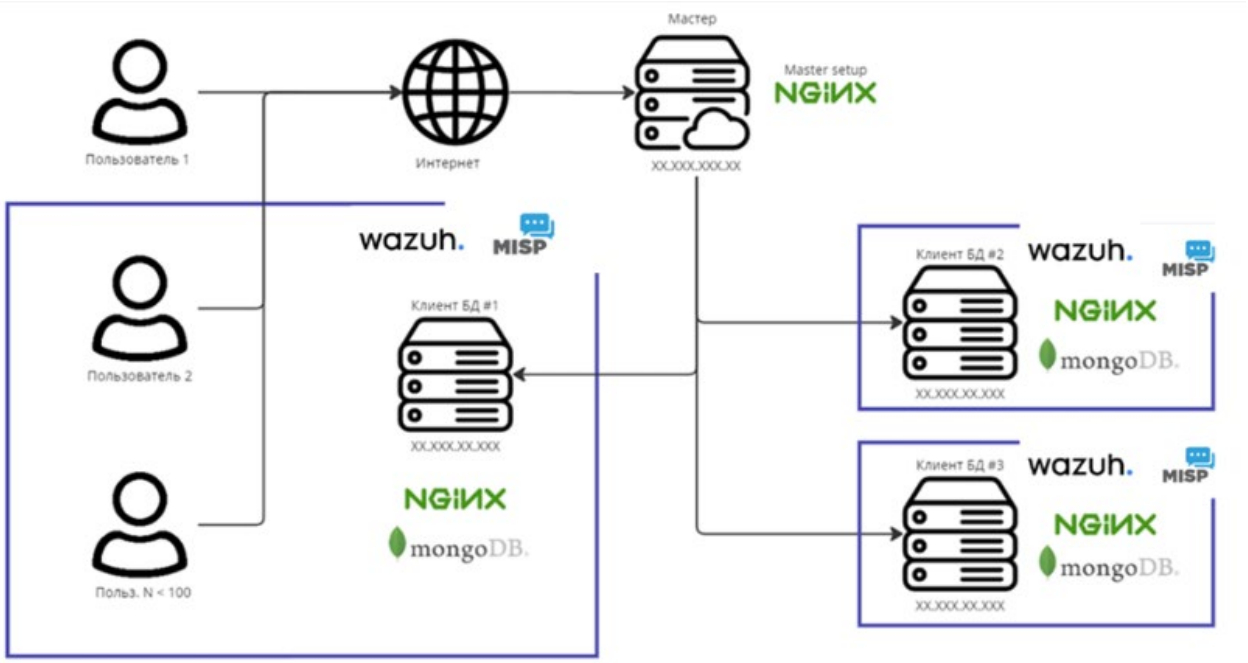

Рис. 1. – Структура СРХД

На данном рисунке изображена система СРХД, состоящая из мастер-сервера и клиентов базы данных (далее БД). На физических серверах установлены Wazuh-агенты и менеджер управления [13], также установлен MISP [9]. В представленной структуре Wazuh-менеджер будет получать события из журнала Sysmon с машин, где установлены Wazuh-агенты, а также считывать индикаторы компрометации и создавать API-запросы к MISP, которые будут содержать считанные индикаторы компрометации для их поиска в базе данных MISP, которые можно подключить внутри платформы

MISP. Если совпадения индикаторов компрометации будут найдены, то Wazuh создаст отчет об инциденте, который будет обогащен из API-ответа платформы MISP с указанием выявленного вредоносного индикатора компрометации. Данные платформы являются решениями с открытым исходным кодом [14].

Данная система позволяет контролировать инциденты на узлах сети, для обеспечения информационной защищенности и возможности расследования инцидентов информационной безопасности [10].

Имитационное моделирование DDoS и брутфорс атак для проверки отказоустойчивости СРХД и стойкости аутентификации

Для проведения брутфорс [12] атаки использовалось программное обеспечение (далее ПО) BurpSuite Professional. В этом программном обеспечении встроены функции для модификации HTTP-запросов, проведения атак и чтения содержимого принятых пакетов. Реализуя методику фаззинга, имеется возможность модифицировать пакеты перед их отправкой. Это дает возможность проверить настройки брандмауэра, Web Application Firewall (далее WAF), а также узнать, проходят ли запросы на сервер и определяется ли тип атаки [15].

Перед началом атаки была проведена тестовая попытка аутентификации со случайными данными HTTP-запроса на сервер. Следующим шагом была его модификация через Intruder в BurpSuite, которая представлена на рис. 2. Во вкладке payload выставлен словарь символов для перебора пароля, а также обозначено само поле с паролем. После отправки HTTP-запроса, сервер отвечает определенным ответом с кодом (HTTP-400 или HTTP-202). По коду ответа и его длине можно определить, что содержится у него внутри (успешная аутентификация, ошибка, и другие виды ответов). После проведения брутфорс атаки нужно отсортировать и найти ответ с кодом 202, с помощью которого можно узнать комбинацию символов пароля. Данная

комбинация и будет являться аутентификационной информацией для пользователя.

Однако, даже если злоумышленник успешно проведет брутфорс атаку, для получения доступа ему придется пройти двухфакторную аутентификацию в приложении. Правильная парольная политика должна включать следующие требования: минимум 8 символов, обязательное наличие букв верхнего и нижнего регистра [A-z], а также знаки препинания. Это позволит задержать злоумышленника при проведении атаки перебора пароля.

Чтобы обеспечить безопасность от атак перебором целесообразно использовать межсетевые экраны и WAF. Грамотная парольная политика и ограничение количества HTTP-запросов с одного IP-адреса также поможет в обеспечении защиты от данной атаки.

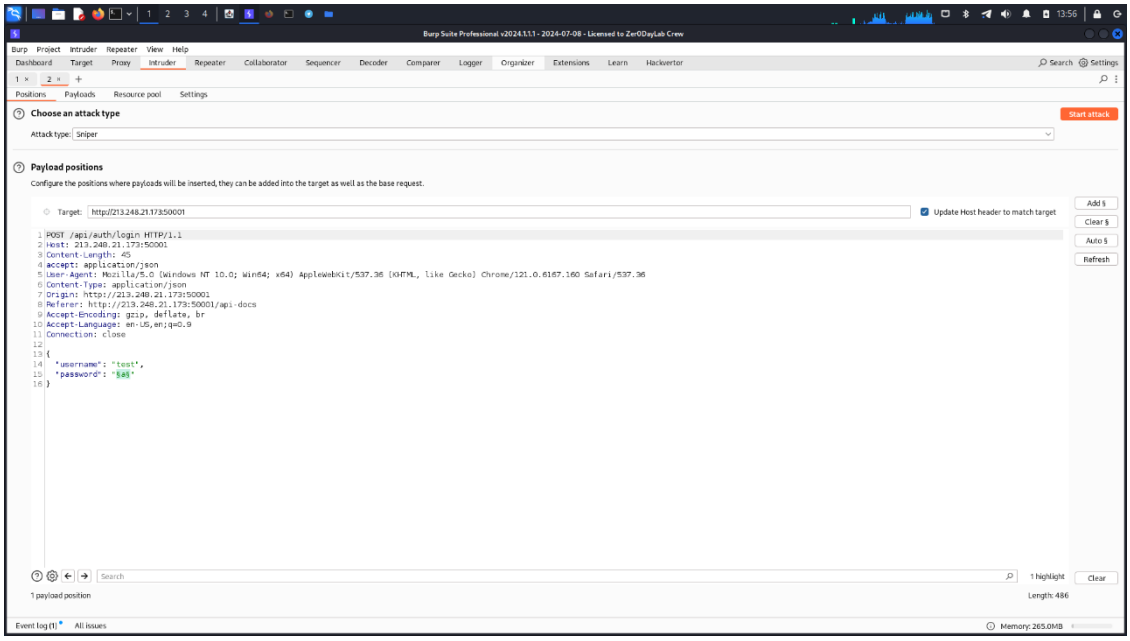

Рис. 2. – Модификация HTTP-запроса

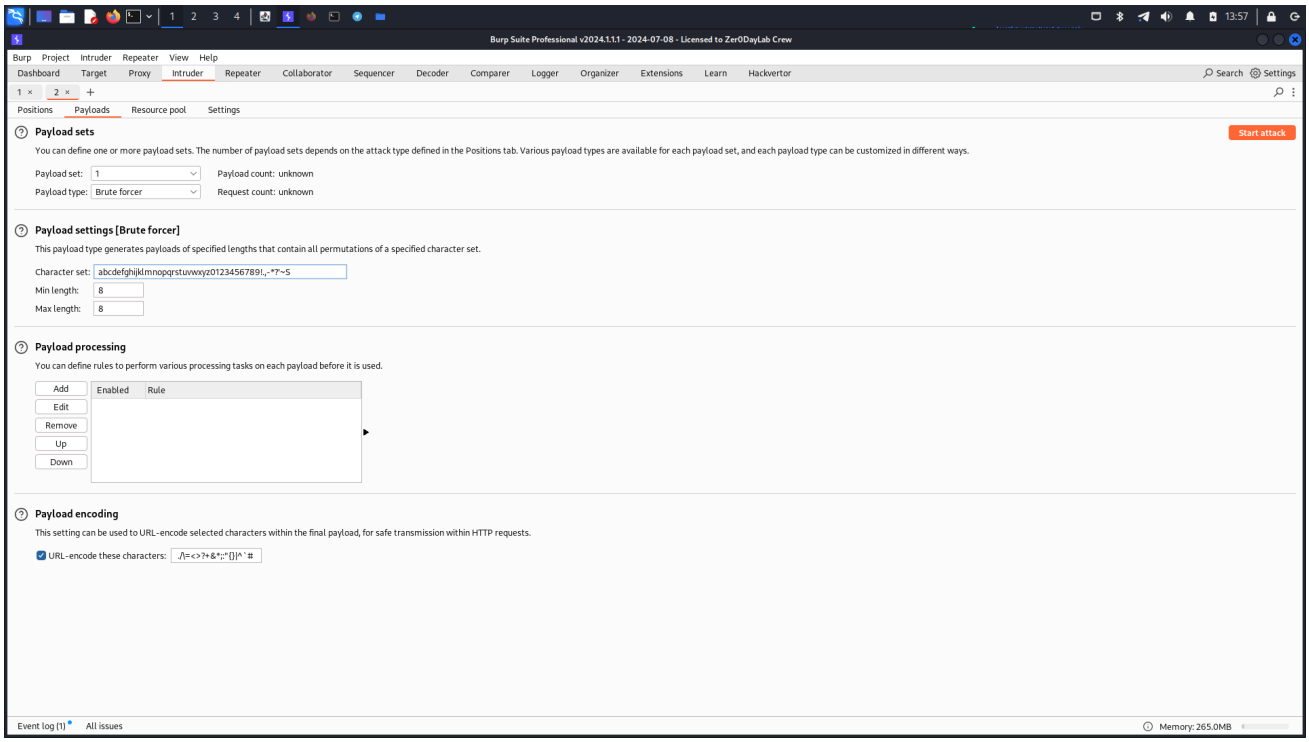

Рис. 3. – Прописывание словаря символов для перебора пароля

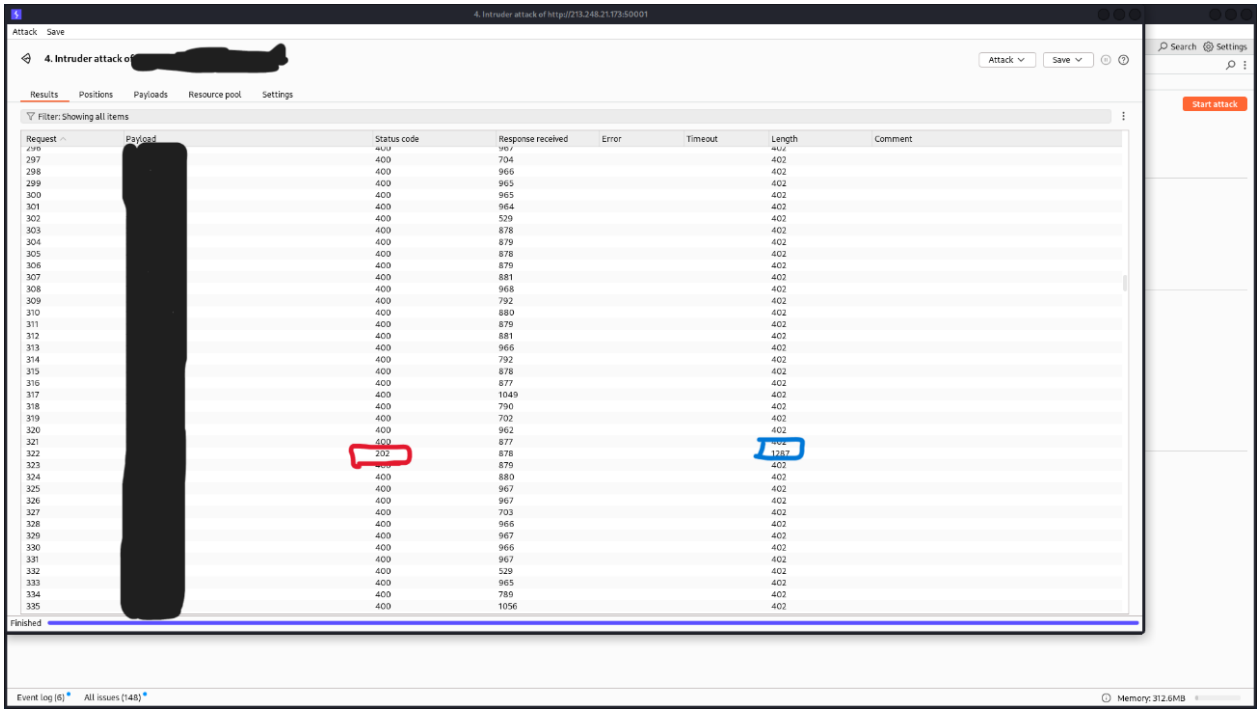

Рис. 4. – Детектирование успешности атаки

Атака типа “отказа в обслуживании” (далее DoS) – это попытка причинить вред, сделав недоступной целевую систему, либо часть

инфраструктуры [12]. Веб-сайт и узел сети становится недоступным для пользователей в следствии перегрузки. Атаки типа “распределенный отказ в обслуживании” (далее DDoS) для своей реализации требует множество взломанных или контролируемых источников. Для реализации атаки DDoS атаки на веб-часть СРХД был использован стенд с несколькими виртуальными машинами. Каждая машина имела Apache и модуль benchmark. С каждой машины отправлялось несколько тысяч запросов на веб-сайт, использовались флаги “k” – keepalive, для осуществления множества запросов в течение одной сессии, “n” -numbers, для указания количества запросов, “c” – количество параллельных соединений, “-h” – передаёт в заголовках информацию о запросах из Google Bot “r” – включение автоматического повтора операции при наличии ошибок. Результат – недоступность сайта из-за перегрузки. Команды, исполняемые в терминале, представлены на рис. 5.

Рис. 5 – Команды для проведения DDoS-атаки на веб-интерфейс РСХД.

Результат проведения DDoS-атаки успешен. Для защиты от DDoS-атак рекомендуется использовать межсетевой экран или WAF. Грамотно настроенные правила помогут снять нагрузку, ограничить нежелательный трафик, а также обеспечат защиту инфраструктуры от атак.

Подобные инциденты нарушения информационной безопасности должны быть задокументированы и детектированы. Именно для этого и будет использоваться решение Wazuh+MISP.

Управление доступом

При развертывании СРХД назначается root-пользователь, который может присваивать права другим пользователям. Последовательность предоставления доступа к ресурсу выглядит следующим образом:

1. Перед предоставлением доступа к ресурсу необходима аутентификация субъекта. Аутентификация является отдельной функцией, в которой

присваивается JWT-токен, с помощью которого будет осуществляться доступ к информации;

2. Решение об авторизации, разрешающее доступ или отказывающее в доступе к ресурсу, принимается на основе соответствия аутентификационных данных и прохождения двухфакторной аутентификации. Для передачи результата решения выпускается токен доступа и токен обновления;
3. На основе результата решения осуществляется авторизация для доступа к ресурсу и предоставляется доступ к системе.

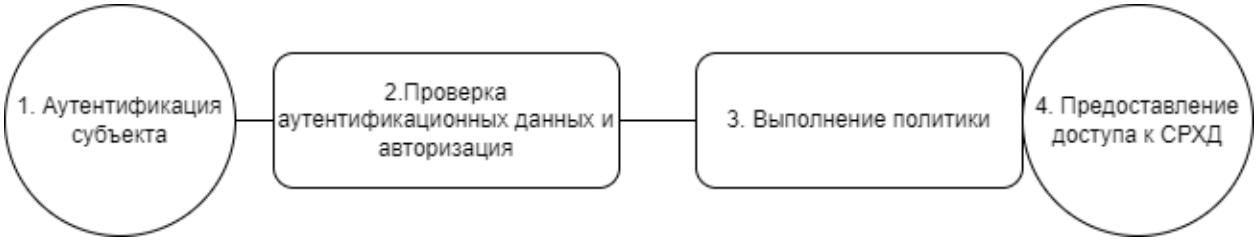

Рис. 6. – Последовательность действий в модели управления доступом

Таблица №1

Матрица доступа в СРХД

		Скачивание файлов	Удаление файлов	Запись файлов в систему	Присвоение ролей пользователям	Доступ к конфигурации СРХД	Смена владельца файла
Пользователи	root	+	+	+	+	+	+
	admin	+	+	+	+	+	+
	user	+	+	+			

Аутентификация

Для веб-приложения предложено установить двухфакторную аутентификацию для повышения уровня защищенности. Двухфакторная аутентификация – это метод проверки подлинности пользователя в сервисе с

помощью данных двух разных типов, например – пароль и секретный код, который приходит на электронную почту, на номер телефона в виде SMS или в приложение на телефоне, которое связано с аккаунтом пользователя в сервисе.

Из широко используемых методов получения одноразового пароля был выбран метод с использованием приложения Яндекс Ключ и общепринятого алгоритма двухфакторной аутентификации Time-based One-time Password Algorithm (далее TOTP).

TOTP – это алгоритм для создания одноразовых паролей для защищенной аутентификации на основе времени, который является улучшенной версией HMAC-Based-One-time Password Algorithm (далее HOTP) [2]. TOTP является алгоритмом односторонней аутентификации, когда сервер убеждается в подлинности клиента. Важно отметить, что TOTP может использовать HMAC-SHA-256, HMAC-SHA-512 и другие HMAC-хеш-функции, в отличии от HOTP, который использует HMAC-SHA-1.

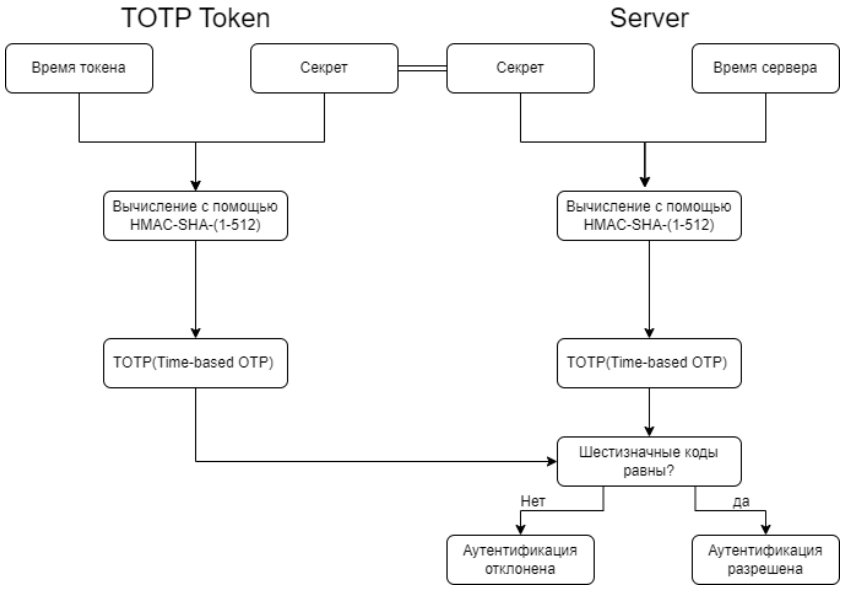

Рис. 7. – Диаграмма последовательности использования TOTP-токена

Для того, чтобы получать шестизначные пароли, в приложении нужно ввести название сервиса, логин и секретный ключ, который выдаст сервер, что показано на рис.8. После ввода данных, приложение будет генерировать коды для учетной записи пользователя, которые он будет вводить при прохождении двухфакторной аутентификации. Также доступна аутентификация с помощью QR-кода, который пользователь будет сканировать с экрана своего монитора, коды представлены на рис.9, рис.10 и рис.11.

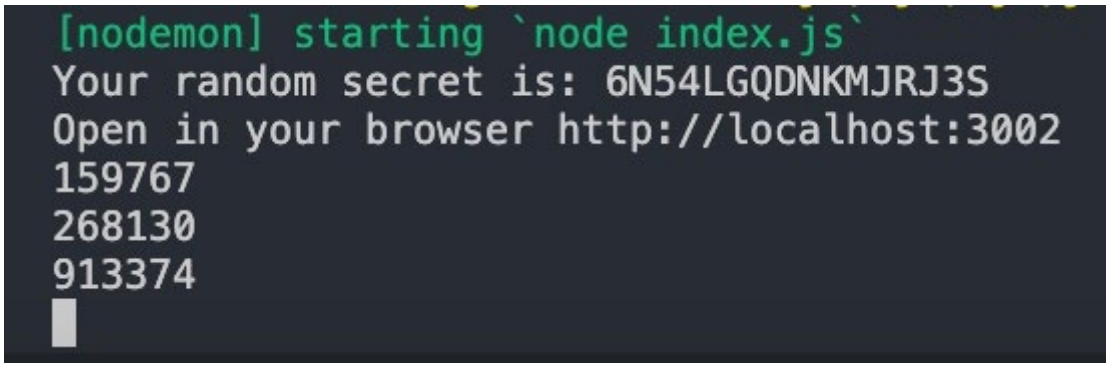

Рис. 8. – Окно консоли с секретным ключом и 6-значными кодами

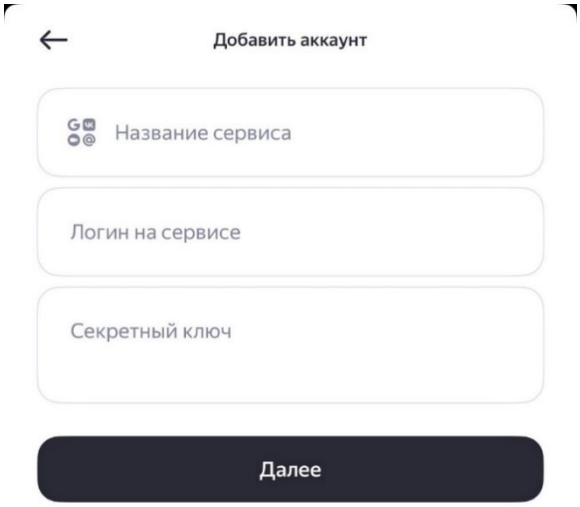

Рис. 9. – Внесение данных в приложение для получения 6-ти значного кода

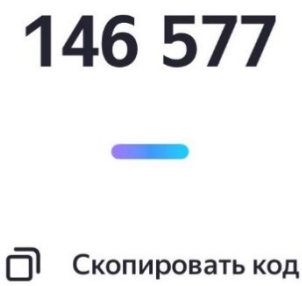

Рис. 10. – Шестизначный код-пароль

Рис. 11. – QR-код для двухфакторной аутентификации

Кода из 6 символов вполне достаточно для обеспечения надлежащей защиты пользовательского аккаунта. Учитывая требования парольной политики, в сочетании с применением кода, который обновляется каждые 30 секунд, подбор пароля методом перебора с минимальной вероятностью даст положительный результат [7].

После прохождения двухфакторной аутентификации пользователь получает JSON Web Token (далее JWT) – стандарт для создания токенов доступа, основанный на формате JSON [3]. Он широко используется для

передачи данных аутентификации в клиент-серверных приложениях. Токены создаются сервером, подписываются секретным ключом и передаются клиенту, который в дальнейшем использует данный токен для подтверждения подлинности запросов от своей учётной записи [4].

В разрабатываемом приложении для подписи JWT используется алгоритм хэширования HMAC-SHA256. Итоговая подпись имеет размер 256 бит, что достаточно для проверки подлинности и целостности данных аутентификации.

Для повышения безопасности, срок действия JWT ограничен, и составляет 15 минут. Это означает, что через 15 минут после выполнения входа, сервер перестанет принимать запросы от пользователя, предоставляющего просроченный токен, и ему нужно будет повторить вход. Чтобы время действия сессии пользователя не зависело от срока действия JWT, применяется токен обновления доступа (refresh token).

Токен обновления доступа – это дополнительный токен, который используется для получения нового JWT после истечения предыдущего. В отличие от JWT, срок действия которого ограничен 15 минутами, токен обновления доступа имеет большее время жизни (от недели до месяца). Применение токена обновления доступа позволяет пользователю не выполнять повторный вход в приложение, и иметь постоянно активную сессию, при этом, сохранив безопасность использования краткосрочных JWT. Когда срок действия JWT истекает, клиент отправляет промежуточный запрос на обновление токена доступа, передав токен обновления доступа. Сервер выполняет необходимые проверки, генерирует новую пару токенов и отправляет их клиенту [6].

Генерация токена обновления доступа происходит по алгоритму UUID v4. Он представляет собой случайный идентификатор длительностью 128 бит. Токен обновления может иметь следующий вид: "840d1728-5cf2-4297-b4e8-

12cd0725fc45”. Алгоритм обновления токена в системе выглядит следующим образом:

1. Пользователь отправляет API-запрос на сервер с токеном доступа, который уже недействителен;
2. Сервер отправит ошибку 401 на клиент пользователя;
3. Клиент пользователя в ответ на ошибку 401/403 отправит свой токен обновления (refresh token);
4. Сервер выдает пользователю новую пару access и refresh токенов;
5. Пользователь повторяет API-запрос из шага один, добавляя к нему новый токен доступа (рис. 12).

Рис. 12. – Диаграмма последовательности токена обновления доступа

Шифрование файлов в СРХД

Для реализации шифрования файлов в СРХД используется алгоритм шифрования ГОСТ 28147-89 [5]. Данный алгоритм шифрования является блочным с 256-битным ключом, который оперирует блоками данных по 64

бита. Внутри СРХД используется режим работы “гаммирование с обратной связью” [8]. Алгоритм работы шифра:

1. Исходный файл разбивается на блоки по 64 бита;
2. На каждый блок используется операция XOR с наложением гаммы, тоже длиной 64 бита;
3. Гамма формируется шифрованием 64-битного блока «состояния» с помощью ключа в режиме простой замены;
4. В момент начала шифрования сообщения блок принимается равным синхропосылке или вектору инициализации;
5. В следующей итерации вместо синхропосылки используется зашифрованный блок текста из предыдущей.

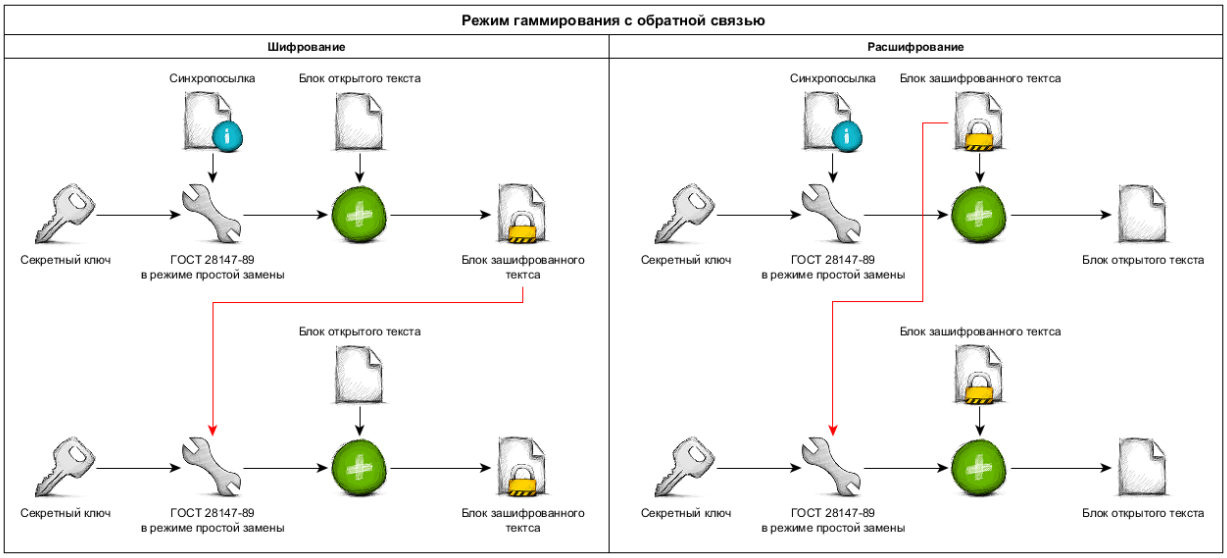

Рис. 13. – Алгоритм гаммирования с обратной связью

Принцип работы разрабатываемой платформы

Применяется система слежения и выявления информационных инцидентов с использованием платформ MISP и Wazuh для обеспечения непрерывного контроля информационной защищенности конечных точек СРХД. Платформу можно использовать на основании аутсорсинга или развернуть свою в зависимости от потребностей малого бизнеса.

Платформа контролирует 4 конечные точки: три сервера и мастер приложение (рис. 14).

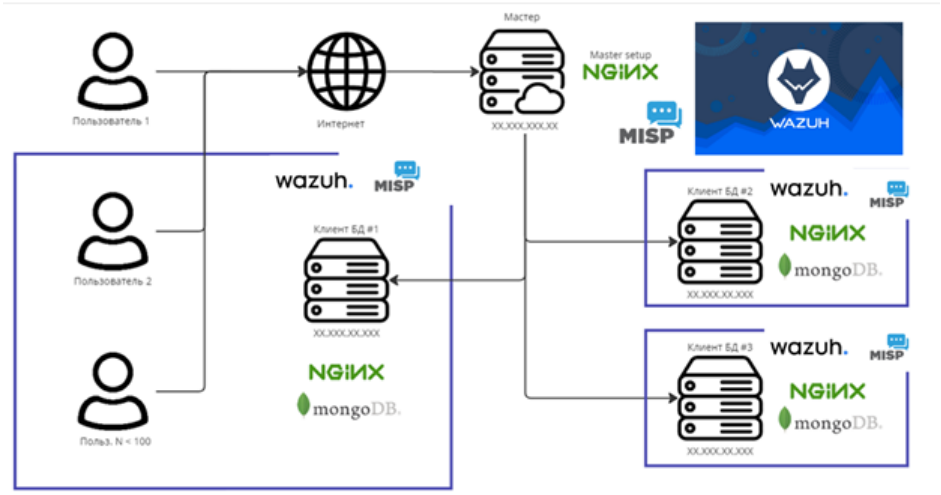

Рис. 14. – Схема сети СРХД

Wazuh устанавливается на мастер-сервер и на клиенты БД в качестве агентов. Для конечных точек СРХД будут использоваться одинаковый набор правил, поэтому администратор будет получать информацию о событиях в едином формате. Для визуализации данных на узел с менеджером управления Wazuh устанавливается утилита для визуализации данных Kibana, которая изображена на рис. 15.

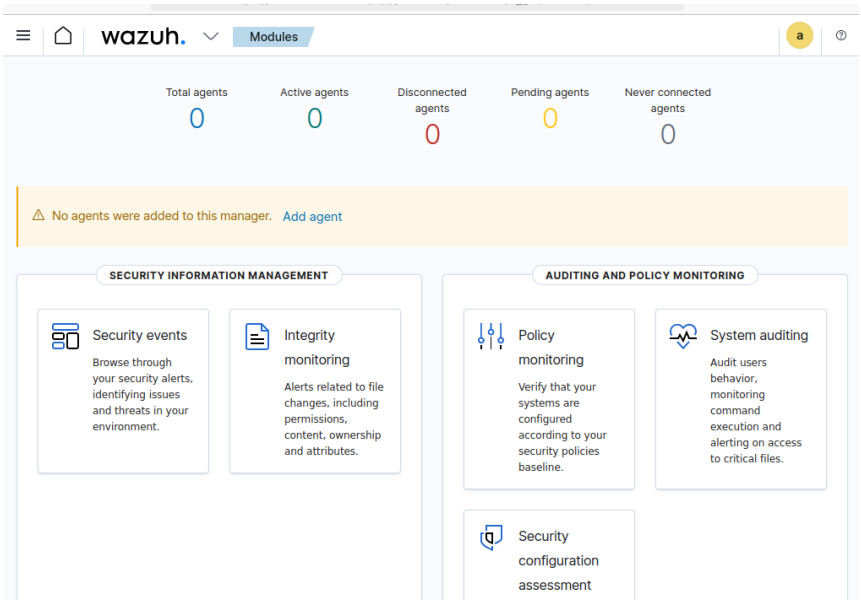

Рис. 15. – Визуализация данных с помощью Kibana

После установки менеджера управления нужно подключить агент и настроить его на передачу событий Sysmon. Подключенный агент появится в графической панели сервера Wazuh вместе с событиями Sysmon, который представлен на рис 16.

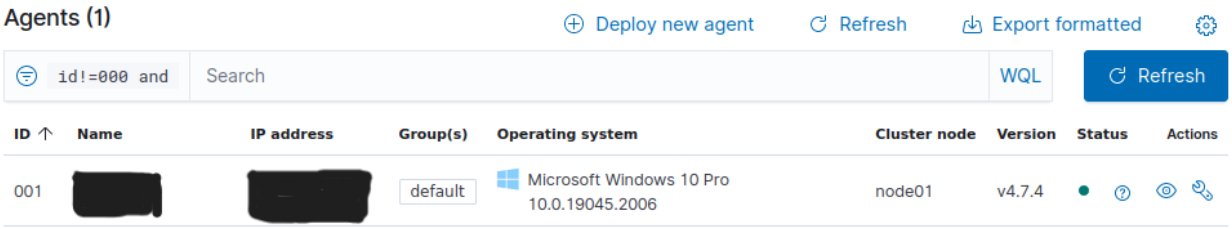

Рис. 16. – Подключенный агент Wazuh

Далее необходимо разработать правила корреляции для отслеживания событий Wazuh. Поскольку к событиям Sysmon еще не применялся анализ на основании данных о киберугрозах, события должны иметь низкий уровень угрозы по умолчанию.

Wazuh позволяет делить события по уровню опасности в 13 балльной системе, где 0 баллов – абсолютно безопасное событие, 13 баллов – явно вредоносное событие. Исходя из этого, в правилах для событий Sysmon был установлен уровень 3, соответствующий уровню опасности «INFO». Уровень «INFO» обычно используется для записи общей информации о нормальном функционировании приложения или системы, которая может быть полезна для мониторинга и диагностики. Примерный набор правил представлен на рис. 17. После активации набора правил, в графическом интерфейсе будут отображаться события нарушения безопасности, которые произошли на агентах.

```
<rule id="61603" level="3">
  <if_sid>61600</if_sid>
  <field name="win.system.eventID">^1$</field>
  <description>Sysmon - Event 1: Запуск процесса $(win.eventdata.description)</description>
  <options>no_full_log</options>
  <group>sysmon_event1,</group>
</rule>

<rule id="61605" level="3">
  <if_sid>61600</if_sid>
  <field name="win.system.eventID">^3$</field>
  <description>Sysmon - Event 3: Сетевое подключение к $(win.eventdata.destinationIp):$(win.eventdata.destinationPort) от $(win.eventdata.image)</description>
  <options>no_full_log</options>
  <group>sysmon_event3,</group>
</rule>

<rule id="61608" level="3">
  <if_sid>61600</if_sid>
  <field name="win.system.eventID">^6$</field>
  <description>Sysmon - Event 6: Загрузка драйвера $(win.eventdata.imageLoaded)</description>
  <options>no_full_log</options>
  <group>sysmon_event6,</group>
</rule>

<rule id="61617" level="3">
  <if_sid>61600</if_sid>
  <field name="win.system.eventID">^15$</field>
  <description>Sysmon - Event 15: $(win.eventdata.targetFilename) запущен именованный файловый поток процессом $(win.eventdata.image)</description>
  <options>no_full_log</options>
  <group>sysmon_event_15,</group>
</rule>
<rule id="61650" level="3">
  <if_sid>61600</if_sid>
  <field name="win.system.eventID">^22$</field>
  <description>Sysmon - Event 22: DNS запрос</description>
  <options>no_full_log</options>
  <group>sysmon_event_22,</group>
</rule>

<rule id="61651" level="3">
  <if_sid>61600</if_sid>
  <field name="win.system.eventID">^23$</field>
  <description>Sysmon - Event 23: Файл удален</description>
  <options>no_full_log</options>
  <group>sysmon_event_23,</group>
</rule>

<rule id="61652" level="3">
  <if_sid>61600</if_sid>
  <field name="win.system.eventID">^24$</field>
  <description>Sysmon - Event 24: Копирование в буфер обмена</description>
  <options>no_full_log</options>
  <group>sysmon_event_24,</group>
</rule>

<rule id="61653" level="3">
  <if_sid>61600</if_sid>
  <field name="win.system.eventID">^25$</field>
  <description>Sysmon - Event 25: Образ процесса изменен</description>
  <options>no_full_log</options>
  <group>sysmon_event_25,</group>
</rule>
```

Рис. 17. – Правила корреляции Wazuh

В рамках разрабатываемой системы для выявления инцидентов используется платформа обмена данными о киберугрозах MISP [15]. Данное ПО с открытым исходным кодом позволяет получать актуальную информацию о киберугрозах от аналитических центров по всему миру. В предоставляемой информации более всего интересны индикаторы компрометации, так как они будут сравниваться с индикаторами из локальных событий, поступающих от конечных точек. Механизм выявления вредоносной активности реализуется следующим образом:

1. Событие, содержащее индикаторы компрометации, собирается на

конечной точке (реализовано в первой главе);

2. Событие передается в платформу управления событиями и инцидентами Wazuh вместе с индикаторами компрометации, событию присваивается информативный статус опасности (реализовано во второй главе);

3. Из события извлекаются индикаторы компрометации и передаются на платформу управления данными о киберугрозах MISP

4. В платформе индикаторы компрометации сравниваются с индикаторами из базы данных, собираемой MISP

5. В случае совпадения, событие возвращается в платформу управления событиями и инцидентами Wazuh и на основании совпадения индикаторов с заведомо вредоносными, ему присваивается статус инцидента и максимальный уровень опасности – 12. Создается отчет о выявленном инциденте.

Для увеличения эффективности анализа событий, необходимо чтобы подключенные источники данных о киберугрозах содержали как можно больше возможных типов индикаторов компрометации: URL адреса, IP-адреса, домены, файловые хеши по алгоритмам MD5 и SHA256. Для работы были выбраны следующие источники данных о киберугрозах:

1. CIRCL.
2. Abuse.ch.
3. Alienvault.com.

Интерфейс управления источниками представлен на рис. 18.

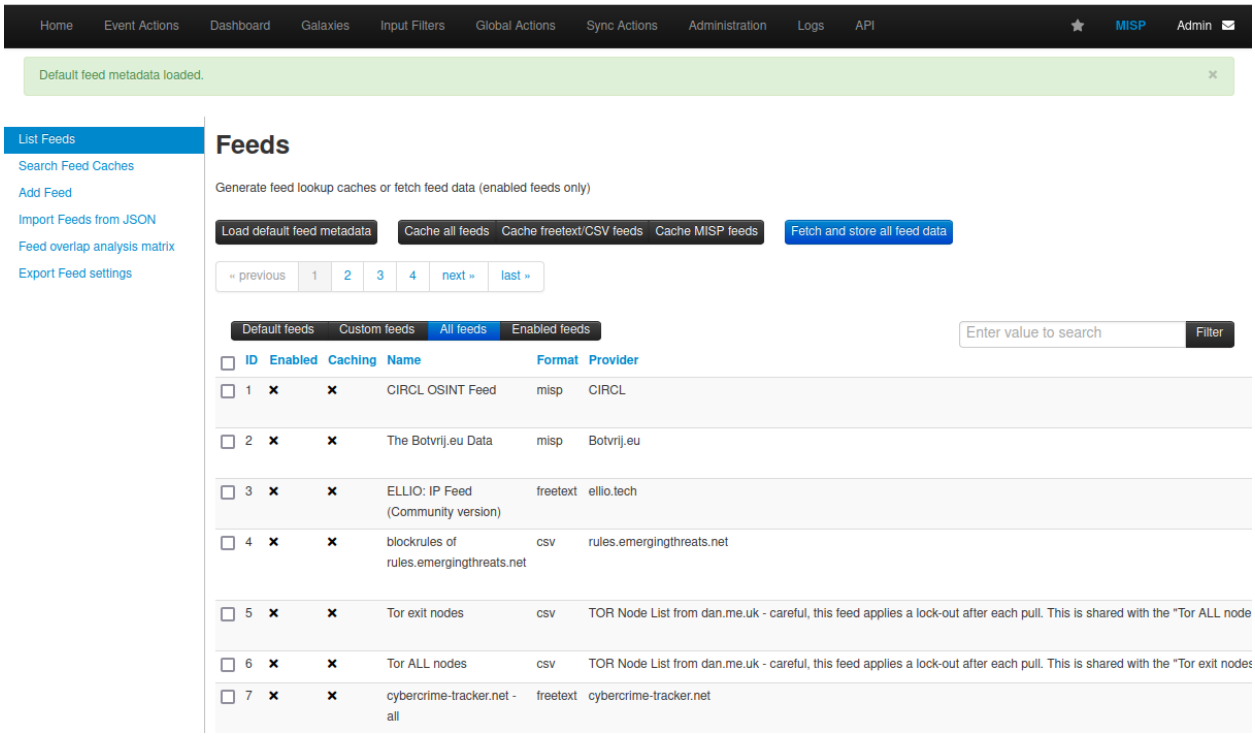

Рис. 18. – Управление источниками данных о киберугрозах

Для активации этих источников необходимо включить кэширование. Кэширование каналов приводит к загрузке всех индикаторов компрометации от этих источников в развернутую локально базу данных MySQL. Все кэшированные данные регулярно обновляются вместе с обновлениями источников. Для запуска процесса выявления инцидентов из событий необходимо организовать обмен данными между платформами MISP и Wazuh. Связь между инструментами должна быть настроена через программный интерфейс API, поскольку они развернуты на сетевых адресах на разных виртуальных машинах. Блок схема скрипта интеграции представлена на рис. 19.

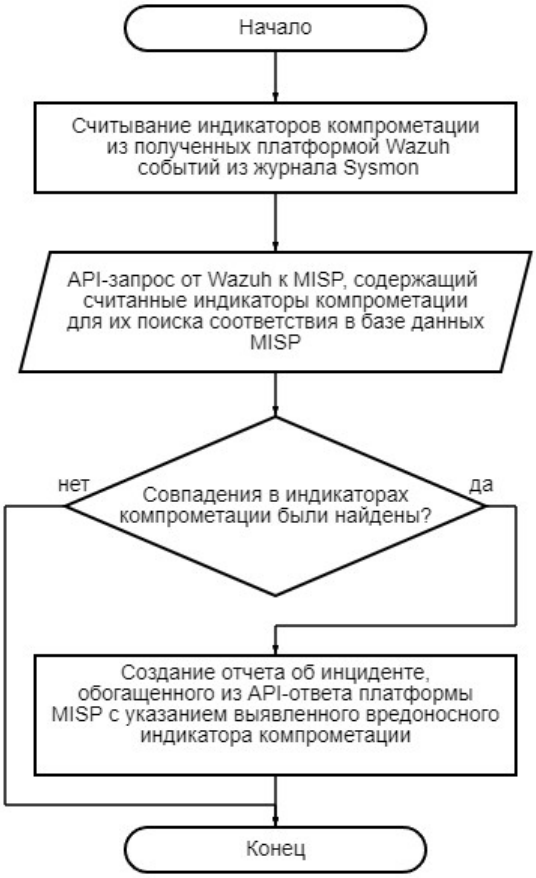

Рис. 19. – Блок-схема программного кода интеграции

Далее необходимо разместить программный код интеграции на сервере Wazuh в директории /var/ossec/integrations. Для работы программного кода необходимо сгенерировать ключ аутентификации API на платформе MISP [11] на рис. 20. После активации скрипта интеграции необходимо написать правила корреляции для того, чтобы Wazuh правильно интерпретировал API-ответы от MISP. Для проверки работоспособности завершенного программного модуля системы, можно осуществить Ping-запрос с защищаемой конечной точки на вредоносный адрес из базы данных MISP [12]. В качестве примера использован адрес psychology.wikia.com, являющийся вредоносным. После выполнения Ping-запроса в графическом интерфейсе Wazuh появляются

события. После выявления инцидента можно открыть по нему подробный отчет, он представлен на рис. 21.

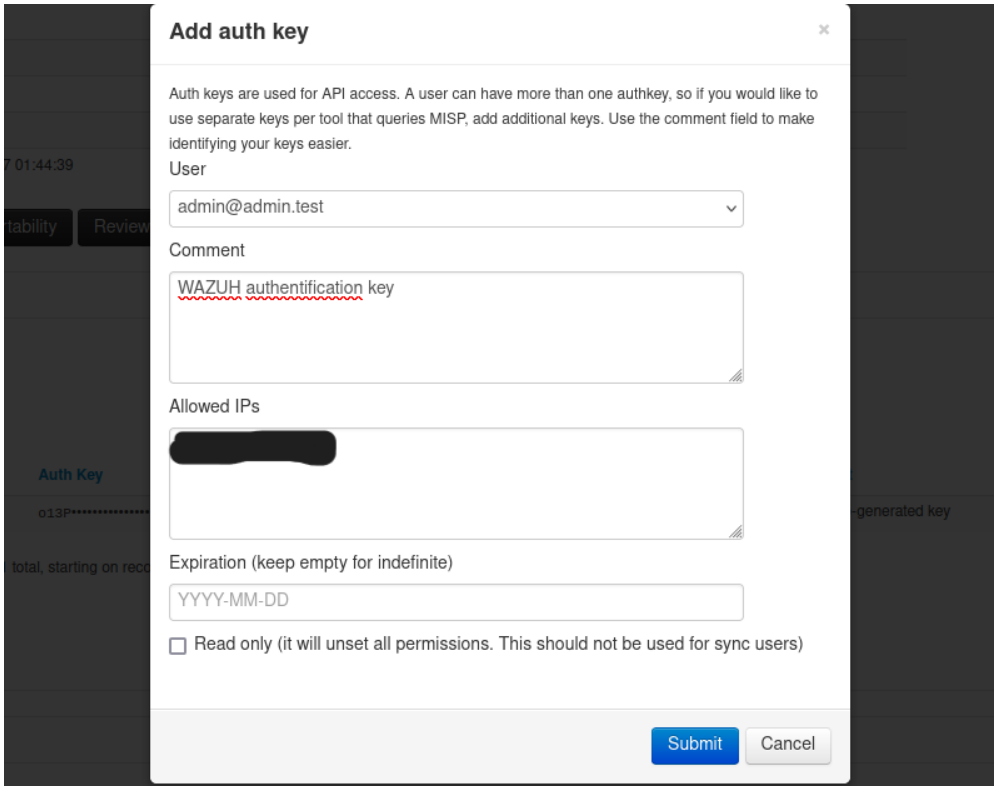

Рис. 20. - Создание ключа аутентификации для адреса сервера Wazuh

t	data.misp.category	Network activity
t	data.misp.event_id	31413
t	data.misp.source.description	MISP - Индикатор компрометации найден в базе - Категория: Network activity, Attribute: psychology.wikia.com
t	data.misp.type	domain
t	data.misp.value	psychology.wikia.com
t	decoder.name	json
t	description	MISP - Индикатор компрометации найден в базе - Категория: Network activity, Attribute: psychology.wikia.com
t	id	1884763.2415
t	input.type	log
t	location	misp
t	manager.name	Wazuh
t	rule.description	MISP - Индикатор компрометации найден в базе - Категория: Network activity, Attribute: psychology.wikia.com
#	rule.firedtimes	2
t	rule.groups	misp, misp_alert
t	rule.id	100622
#	rule.level	12

Рис. 21. – Подробный отчет по инциденту

Выводы

В результате исследования была разработана система распределенного хранения данных. Основное внимание уделено обеспечению информационной безопасности с использованием современных технологий защиты данных, мониторинга инцидентов и контроля доступа.

Проведенные тесты, включая имитацию атак DoS и брутфорс, подтвердили высокую отказоустойчивость и надежность предложенной системы. Для защиты от атак грубой силы используется межсетевой экран, WAF и двухфакторная аутентификация, реализованная на основе алгоритма TOTP. Грамотная парольная политика и ограничения на количество запросов с одного IP-адреса дополнительно усиливают защиту от несанкционированного доступа.

Для обеспечения конфиденциальности данных предложено использование отечественного алгоритма шифрования ГОСТ 28147-89 с режимом гаммирования, который защищает файлы от несанкционированного доступа. Управление доступом реализовано с использованием токенов JWT, что обеспечивает надежную аутентификацию пользователей. Интеграция платформ MISP и Wazuh позволила автоматизировать процесс выявления и анализа угроз, используя индикаторы компрометации и базы данных актуальной информации о киберугрозах.

Литература

1. Лопастейская, Л.Г., Хасянова Ю.Р. Малый бизнес: общая характеристика и критерии отнесения организаций к малому бизнесу // Научные революции: сущность и роль в развитии науки и техники. Сборник статей по итогам Международной научно-практической конференции, Магнитогорск, 2019.

2. TOTP- стандарт. datatracker.ietf.org/doc/html/rfc6238.

3. JWT-token стандарт. datatracker.ietf.org/doc/html/rfc7519.

4. Колесников, А.О. Идентификация пользователей клиент-серверных приложений с помощью JWT-токена // EurasiaScience : Сборник статей XXXVI международной научно-практической конференции, Москва, 2021.

5. ГОСТ 28147-89 docs.cntd.ru/document/1200007350

6. Бетелин А.Б., Егорычев И.Б., Прилипко А.А. О некоторых особенностях JWT аутентификации в веб-приложениях // Труды научно-исследовательского института системных исследований Российской академии наук. 2021. Т. 11, № 1.

7. Moretto N. Two-factor authentication with TOTP - 2018 [электронный ресурс]. - URL: medium.com/@nicola88/two-factor-authentication-with-totp-ccc5f828b6df.

8. Зуев, М.С., Баранов П.А. Шифрование данных. Алгоритм ГОСТ 28147-89 // Психолого-педагогический журнал Гаудеамус. 2010. Т. 2. № 16. С. 208-210.

9. Колесников, А.А., Меньков И.И. Возможности платформы MISP для обнаружения инцидентов информационной безопасности // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2023). Сборник научных статей XII Международной научно-технической и научно-методической конференции. Санкт-Петербург. 2023. Том 2. С. 719-722.

10. Ammi, M. Cyber Threat Hunting Case Study using MISP // Journal of Internet Services and Information Security. 2023. Vol. 13, No. 2. P. 1-29.

11. Mitre-attack Denial of Service attack.mitre.org/techniques/T1499.

12. Mitre-attack Brute Force attack.mitre.org/techniques/T1110.

13. Wazuh-cluster documentation.wazuh.com/current/user-manual/wazuh-server-cluster.html.

14. MISP-documentation github.com/MISP/misp-book.

15. Большаков, А.С., Раковский Д.И. Программное обеспечение моделирования угроз безопасности информации в информационных системах. Правовая информатика. 2020. № 1. С. 26-39.

References

1. Lopasteyskaya L.G., Khasyanova Ju.R. Nauchnye revolyutsii: sushchnost' i rol' v razvitii nauki i tekhniki. Sbornik statey po itogam Mezhdunarodnoy nauchno-prakticheskoy konferentsii, Magnitogorsk, 2019.

2. TOTP standard. URL: datatracker.ietf.org/doc/html/rfc6238.

3. JWT-token standard. URL:datatracker.ietf.org/doc/html/rfc7519.

4. Kolesnikov A.O. EurasiaScience: Sbornik statey XXXVI mezhdunarodnoy nauchno-prakticheskoy konferentsii, Moscow, 2021.

5. GOST 28147-89. URL:docs.cntd.ru/document/1200007350.

6. Betelin A.B., Egorychev I.B., Prilipko A.A. Trudy nauchno-issledovatel'skogo instituta sistemnykh issledovaniy Rossiyskoy akademii nauk. 2021. Vol. 11, No. 1.

7. Moretto N. Two-factor authentication with TOTP. 2018. URL: medium.com/@nicola88/two-factor-authentication-with-totp-ccc5f828b6df.

8. Zuev M.S., Baranov P.A. Shifrovanie dannykh. Psikhologo-pedagogicheskiy zhurnal Gaudeamus. 2010. Vol. 2, No. 16, pp. 208-210.

9. Kolesnikov A.A., Menkov I.I. Aktual'nye problemy infotelekommunikatsiy v nauke i obrazovanii (APINO 2023). Sbornik nauchnykh statey XII Mezhdunarodnoy

nauchno-tekhnicheskoy i nauchno-metodicheskoy konferentsii, Saint Petersburg, 2023. Vol. 2, pp. 719-722.

10. Ammi M. Journal of Internet Services and Information Security. 2023. Vol. 13, No. 2, pp. 1-29.

11. MITRE ATT&CK: Denial of Service. URL: attack.mitre.org/techniques/T1499.

12. MITRE ATT&CK: Brute Force. URL: attack.mitre.org/techniques/T1110.

13. Wazuh-cluster. URL: documentation.wazuh.com/current/user-manual/wazuh-server-cluster.html.

14. MISP documentation. URL: github.com/MISP/misp-book.

15. Bolshakov A.S., Rakovsky D.I. Pravovaya informatika. 2020. No. 1, pp. 26-39.

Дата поступления: 13.12.2024

Дата публикации: 4.02.2025