**Онлайн-школа Skillfactory**

Мукосов Станислав Дмитриевич

**Выпускная квалификационная работа**

**по направлению ДПО «Информационная безопасность»**

по теме

**«Метод и система сбора поверхности атаки для внешнего периметра организации»**

Студент

Мукосов С.Д.

|  |  |
| --- | --- |
| Рецензент  ученая степень, звание  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Руководитель  ученая степень, звание (при наличии)  С. СЕРОВ  Консультант  ученая степень, звание (при наличии)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Москва 2024 г.

Оглавление

[Введение 3](#_Toc161083636)

[1. Поверхность атаки и внешний периметр организации в информационной безопасности 4](#_Toc161083637)

[1.1 Защищённость внешнего периметра организации 4](#_Toc161083638)

[1.2 Уязвимости в информационной системе 6](#_Toc161083639)

[1.3 Методы поиска уязвимостей 8](#_Toc161083640)

[1.4 Анализ защищенности внешнего периметра ИС 11](#_Toc161083641)

[2. Разработка системы для мониторинга внешнего периметра и сбора поверхности атаки 14](#_Toc161083642)

[2.1 Целесообразность разработки системы мониторинга внешнего периметра и сбора поверхности атаки 14](#_Toc161083643)

[2.2 SIEM-системы. Wazuh. 15](#_Toc161083644)

[2.3 Caldera 17](#_Toc161083645)

[3. Практическая часть. 20](#_Toc161083646)

[3.1 Установка и настройка Caldera 21](#_Toc161083647)

[3.2 Установка и настройка Wazuh 22](#_Toc161083648)

[3.3 Примеры работы совместной работы Wazuh и Caldera 25](#_Toc161083649)

[3.4 Python скрипт для работы с Caldera и Wazuh 29](#_Toc161083650)

[Список использованных источников 33](#_Toc161083651)

[Приложение 1 35](#_Toc161083652)

# **Введение**

Проведение анализа защищенности внешнего периметра организации одна из важнейших процедур для информационной безопасности, так как она обеспечивает нормальное функционирование и беспрерывную работу организации. Важно не только правильно организовать защищенность периметра, но и получать актуальную информацию о состоянии информационной системы. Однако не существует какой-либо методики, которая объяснила и нормировала процесс такой процедуры. В связи с этим часто среди специалистов возникает вопрос о том, как выполнять анализ защищенности. Как итог каждый разрабатывает собственную методику «на ходу», в которой могут допускаться ошибки.

В данной работе мы более детально изучим: какие существуют уязвимости; каким способом можно их отслеживать; какие системы существуют для поиска уязвимостей; как проанализировать защищенность внешнего периметра организации

В связи с этим предлагаются методические рекомендации проведения анализа защищенности информации на этапе внедрения системы защиты и мониторинга в ИС. Методические рекомендации создаются в помощь специалистам по информационной безопасности в малых организациях.

# 

# **Поверхность атаки и внешний периметр организации в информационной безопасности**

Поверхность атаки - это общий термин, обозначающий области системы, устройства или сети, которые содержат уязвимости в системе безопасности, которые могут быть использованы. Поверхность атаки компьютерных систем и устройств организации часто может существенно различаться в зависимости от того, для чего они используются и как они были настроены. Это не следует путать с “поверхностью атаки при этичном взломе”, которая относится к потенциальному риску, создаваемому уязвимостями в данной операционной системе или приложении, установленном на вычислительном устройстве.

На практике этот метод можно использовать для определения того, уязвимо ли устройство к известным и неизвестным формам атак, одним из примеров этого может быть анализ операционной системы, которую может использовать злоумышленник.

Например, если ИТ-инфраструктура организации использует операционную систему Linux для своей платформы электронной почты, то возможно, что она может быть подвержена атаке с использованием уязвимости shell shock. Поверхности атаки также можно проверять на наличие уязвимостей в программном обеспечении не операционной системы, а также в аппаратных устройствах, таких как коммутаторы и маршрутизаторы, хотя их обычно сложнее исправить.

## **1.1 Защищённость внешнего периметра организации**

Анализируя нормативно-правовые акты, следует отметить, что защищенность является основным из показателей эффективности функционирования информационной системы по защите внешнего периметра организации (далее ИС), наряду с такими показателями как надежность, отказоустойчивость, производительность. Под защищенностью ИС обычно понимается степень адекватности реализованных в ней механизмов защиты информации существующим в данной среде функционирования рискам, связанным с осуществлением угроз безопасности, нарушающих основные свойства информации, такие как конфиденциальность, целостность и доступность.

Документы, которые определяют необходимые уровни защищенности информации в ИС – это указы Президента, постановления Правительства РФ, руководящие документы ФСТЭК и ФСБ России, федеральные законы.

Основу нормативной базы составляют в нашей стране руководящие документы и приказы ФСТЭК России в области защиты от НСД к информации.

Защищенность информации достигается при помощи системы защиты информации ИС, нейтрализующей актуальные угрозы. Сама система защиты информации объединяет в себе технические и организационные меры, которые определены по актуальным угрозам безопасности информации в ИС. Защищенность ИС обеспечивает оператор ИС, для которого определен контроль по выполнению требований по защите информации (определение типа ИС, актуальных угроз, уровня защищенности ИС).

После выполнения всех этих требований идет проверка правильной работы механизмов защиты информации по существующим уязвимостям, угрозам и рискам – анализ защищенности ИС. Проведение анализа защищенности ИС позволит узнать актуальную информацию по используемым техническим мерам и средствам защиты информации, а также по итогам выяснить недостатки и повысить уровень безопасности ИС. Но для того, чтобы это сделать, необходимо начать с анализа ИС на уязвимости, которые могут привести к различным угрозам.

## **1.2 Уязвимости в информационной системе**

Основными показателями защищенности являются уязвимости, то есть система будет считаться защищенной в том случае, когда в ней нет известных уязвимостей.

Изучая государственные стандарты, было найдено определение уязвимости – это свойство информационной системы, обусловливающее возможность реализации угроз безопасности обрабатываемой в ней информации.

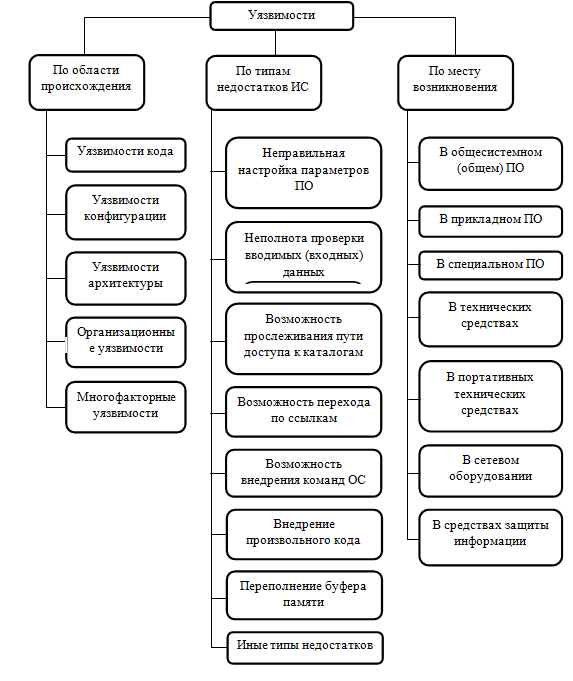


Рисунок 1.1 – Классификация уязвимостей в ИС

Существует множество классификаций уязвимостей. Большинство организаций выделяют собственные классификации и пользуются ими при определение уязвимостей, но с такими классификациями существует возможность пропустить какую-либо уязвимость или неверно её классифицировать, что в дальнейшем приведёт к угрозам , а также хищению, потери и модификации информации.

Большинство уязвимостей находят эксперты в области тестирования информационных систем и информационной безопасности либо инструментальные средства.

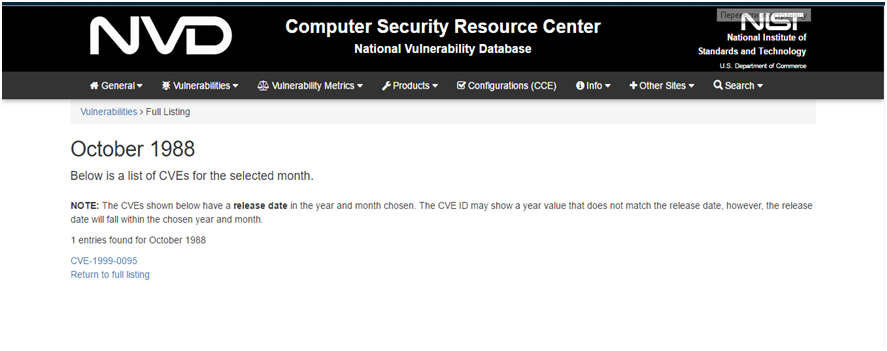


Рисунок 1.2 – Пример описания уязвимостей (NVD)

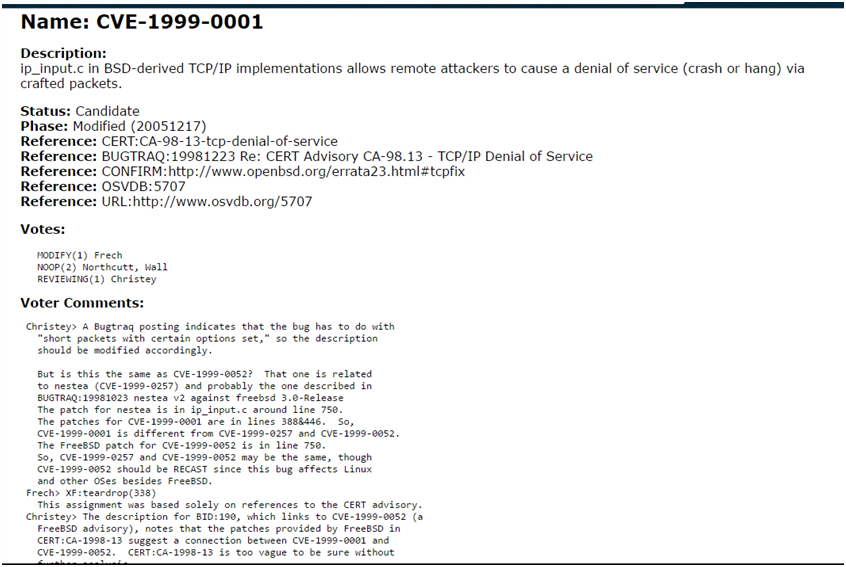


Рисунок 1.3 – Пример описания уязвимостей (СVE)

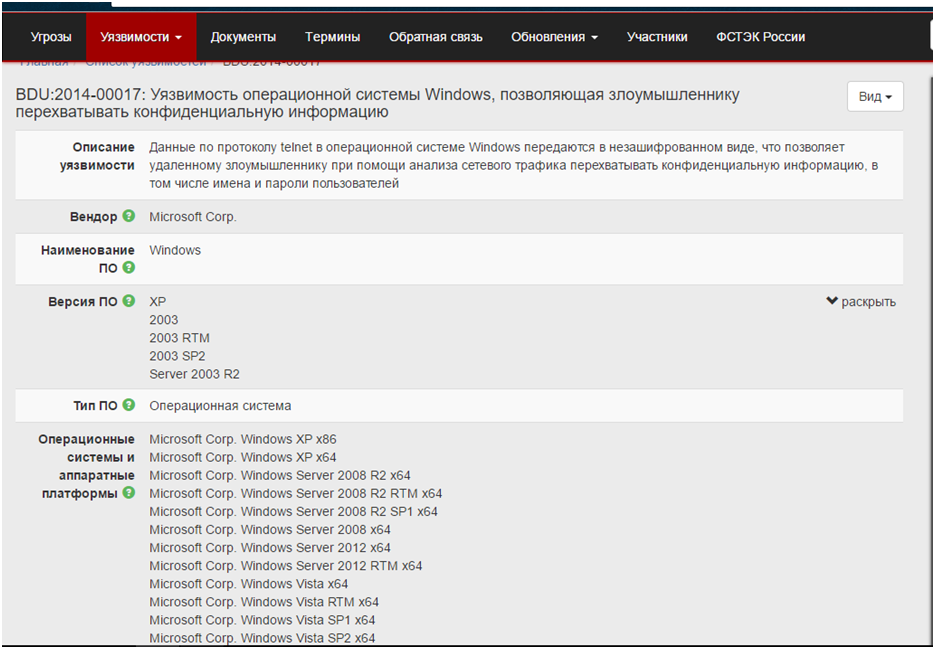


Рисунок 1.4 – Пример описания уязвимостей (ФСТЭК)

## **1.3 Методы поиска уязвимостей**

Существует несколько методов сбора информации о поверхности атаки, включая активные и пассивные подходы. Рассмотрим таблицу 1.

|  |  |  |  |
| --- | --- | --- | --- |
| **Тип метода** | **Метод** | **Преимущества** | **Недостатки** |
| Активные методы | Сканирование портов и служб (Port Scanning): | Позволяет определить, какие порты и службы открыты на сетевых устройствах, что может помочь в выявлении потенциальных уязвимостей. | Может привлечь внимание сетевых администраторов или средств обнаружения вторжений (IDS/IPS), что может вызвать блокировку сканирующего IP-адреса.  Также не всегда точно определяет версии служб и уязвимости. |
| Сбор информации через интернет (Open Source Intelligence, OSINT): | Позволяет собирать информацию о компании или организации из общедоступных источников (социальные сети, форумы, блоги и т.д.), что может помочь злоумышленникам понять структуру сети и возможные слабые места. | Ограничено общедоступной информацией.  Не всегда точно и актуально. |
| Пассивные методы | Сбор информации о доменных именах (Domain Name System, DNS): | Позволяет получить информацию о доменах и их структуре, что может помочь в планировании атаки. | Могут использоваться методы защиты, такие как скрытие деталей регистрации домена или использование анонимных DNS-серверов, чтобы затруднить сбор информации. |
| Сбор информации с помощью утилит сетевого анализа (Network Reconnaissance Tools): | Позволяют осуществлять более глубокий анализ сетевого трафика и выявлять аномалии. | Могут быть обнаружены и заблокированы системами обнаружения вторжений.  Требуют глубоких знаний сетевой архитектуры и протоколов. |
| Использование специализированных информационно-аналитических систем и сервисов: | Предоставляют широкий спектр функциональности для анализа информации о поверхности атаки, включая сканирование уязвимостей, анализ сетевого трафика и т.д. | Могут требовать значительных инвестиций во время и ресурсы для развертывания и обслуживания.  Также могут не быть достаточно гибкими для адаптации к уникальным потребностям организации. |

Таблица 1. Методы сбора информации о поверхности атаки

Пассивные методы обычно менее заметны для целевых систем и могут быть более предпочтительными в ситуациях, когда требуется минимизировать риск обнаружения. Однако активные методы могут предоставить более полную информацию, поскольку они могут инициировать взаимодействие с целевыми системами. Ниже приведены основные методы сбора информации, с описанием их преимуществ и недостатков (Таблица 1)

## **1.4 Анализ защищенности внешнего периметра ИС**

Целью анализа защищенности внешнего периметра корпоративной сети является оценка уровня защищенности ИС всей организации от атак со стороны сети Интернет, оценка степени критичности выявленных уязвимостей и возможностей по осуществлению атак, а также выработка рекомендаций по ликвидации обнаруженных уязвимостей.

Анализ производится путем эмуляции действий потенциального злоумышленника по проникновению в корпоративную сеть с целью нарушения ее функционирования, внедрения вредоносного ПО, кражи конфиденциальной информации и выполнения других деструктивных действий. Производится также анализ конфигурации средств защиты периметра сети.

При выполнении проверок используется богатый арсенал современных инструментальных средств сетевого сканирования, специализированные средства анализа веб сайтов и сетевых приложений, программы, реализующие конкретные методы взлома, средства подбора паролей, а также ручные проверки. Используемые источники информации (банк угроз и уязвимостей ФСТЭК, NVD, CVE), позволяют гарантировать надежную идентификацию всех известных уязвимостей.

Проверочные мероприятия включают в себя:

* проверка на возможность проникновения в локальную сеть компании, похищения и порчи данных;
* обследование доступных из Интернет сетевых сервисов ;
* проверка межсетевых экранов на наличие уязвимостей;
* обследование Web и Почтового серверов.

В случае обнаружения уязвимостей, предоставляются документальные свидетельства возможности компрометации, искажения, уничтожения критичной информации в предоставленных для исследования Интернет-ресурсах.

При анализе конфигурации средств защиты внешнего периметра ЛВС и управления межсетевыми взаимодействиями особое внимание обращается на следующие аспекты, определяемые их конфигурацией:

настройка правил разграничения доступа (правил фильтрации сетевых пакетов) на МЭ и маршрутизаторах;

* используемые схемы и настройка параметров аутентификации;
* настройка параметров системы регистрации событий;
* использование механизмов, обеспечивающих сокрытие топологии защищаемой сети, включающих в себя трансляцию сетевых адресов (NAT), маскарадинг и использование системы split DNS;
* настройка механизмов оповещения об атаках и реагирования;
* наличие и работоспособность средств контроля целостности;
* версии используемого ПО и наличие установленных пакетов программных коррекций.

Отчет по результатам работы содержит общую оценку уровня защищенности корпоративной сети от внешних сетевых атак, подробное описание обнаруженных уязвимостей по каждому IP-адресу, а также рекомендации по ликвидации уязвимостей и совершенствованию защиты.

# **2. Разработка системы для мониторинга внешнего периметра и сбора поверхности атаки**

## **2.1 Целесообразность разработки системы мониторинга внешнего периметра и сбора поверхности атаки**

В РФ в настоящее время идет сильный подъем малого и среднего бизнеса. Очень часто это небольшие организации с малым числом сотрудников и еще меньшим числом рабочих мест. К сожалению, такие небольшие компании становятся жертвами киберпреступлений. Этому сопутствуют ограниченность бюджета на информационную безопасность, нехватка квалифицированных кадров, непонимание важности защиты информации.

Почти всегда деятельность таких организаций тесно связана с информационными технологиями, в том числе передачей данных посредством сети интернет. Обычно автоматизированное рабочее место (АРМ) на таких предприятиях представлено в виде офисного ПК с популярной операционной системой (Windows 7, 10, и даже, иногда, XP); установленным пакетом MS office; браузером, для выхода в интернет; корпоративным мессенджером. Часто компании создают для осуществления своей деятельности интернет-сайты, используя популярные и простые фреймворки (например Wordpress, Tilda). Отдельный почтовый сервер для таких организаций “роскошь” и, поэтому, они пользуются готовыми решениями от популярных компаний. В таких организациях, зачастую, нет штатного системного администратора (не говоря уже о специалистах информационной безопасности), который мог бы правильно организовать работу сотрудников в сети и сделать безопасным внутренний периметр.

В нашей работе мы попробуем смоделировать такую сеть и постараемся разработать малозатратные по финансовым и ресурсным затратам методы для отслеживания состояния безопасности нашей системы, а также подозрительной активности во внешнем периметре организации. Попробуем предложить простые решения для усиления безопасности нашей сети.

Предположим, что информационная инфраструктура такого предприятия включает в себя несколько автоматизированных рабочих мест (АРМ) на Windows 8, один сервер для хранения общих файлов, на котором установлен Domain Controller. Каждый ПК имеет выход в Интернет. Для обмена файлами сотрудники пользуются общими директориями на сервере и на данных ПК, к которым открывают (закрывают) доступ. Иногда обмен файлами происходит через электронную почту или через desktop версии мессенджеров Whatsapp или Telegram. На постоянной основе общий доступ открыт к отдельным директориям на сервере.

Точек входа для злоумышленника достаточно, но мы сосредоточимся на рабочих ПК, на которых происходит основная деятельность организации. В организации отсутствует отдел ИБ, который мог бы постоянно отслеживать подозрительную активность во внешнем и внутреннем периметре сети, проанализировать защищенность от злоумышленников, просмотреть версии ПО, в котором могли найти критические уязвимости и впоследствии обновить его.

## **2.2 SIEM-системы. Wazuh.**

На рынке существует множество коммерческих решений, которые могут выполнять некоторые функции специалиста отдела ИБ. Например: Alertix (NGR Softlab); Ankey SIEM NG («Газинформсервис»); KUMA («Лаборатория Касперского») и др. К сожалению, небольшие организации часто экономят на информационной безопасности и не хотят тратить финансы на антивирусы, а SIEM-системы тем более отходят далеко на задний план. Но на рынке существуют open-source системы управления событиями безопасности. Предлагаем рассмотреть возможности одной из них - Wazuh.

Wazuh — это бесплатная платформа с открытым исходным кодом, используемая для предотвращения, обнаружения и реагирования на угрозы. Он способен защищать рабочие нагрузки в локальных, виртуализированных, контейнерных и облачных средах. Решение Wazuh состоит из агента безопасности конечных точек, развернутого в контролируемых системах, и сервера управления, который собирает и анализирует данные, собранные агентами. Функции Wazuh включают полную интеграцию с Elastic Stack и OpenSearch, предоставляя поисковую систему и позволяет выполнять сканирование запущенных процессов, проверку уязвимостей CVE, получать отчеты об инцидентах и т. д. Кроме того платформа предлагает мощный инструмент визуализации данных, с помощью которых пользователи могут просматривать предупреждения безопасности. Таким образом, Wazuh — это полноценная SIEM-система (менеджмент информации и событий безопасности), которая хорошо интегрируется с разными платформами.

Преимущества:

* Основана и совместима с популярной SIEM OSSEC;
* Поддерживает различные варианты установки: Docker, Puppet, Chef, Ansible;
* Поддерживает мониторинг облачных сервисов, включая AWS и Azure;
* Включает комплексный набор правил, для обнаружения множества типов атак и позволяет сопоставлять их в соответствие с PCI DSS v3.1 и CIS.
* Интегрируется с системой хранения и анализа логов Splunk визуализации событий и поддержки API.

Недостатки:

* Сложная архитектура

Но что делать после установки и настройки SIEM-системы? Ждать, когда злоумышленники организуют атаку на нашу сеть и после этого делать выводы? Мы при­вык­ли думать, что у нас всег­да все хорошо в пла­не информа­цион­ной безопас­ности. Что­бы удос­товерить­ся в этом, некото­рые ком­пании про­водят пен­тест. Но сущес­тву­ет спо­соб убе­дить­ся в сво­ей готов­ности к кибер­атакам и без пен­теста.

## **2.3 Caldera**

Caldera — это авто­мати­зиро­ван­ная сис­тема эму­ляции дей­ствий зло­умыш­ленни­ков на конеч­ных рабочих стан­циях, соз­данная ком­пани­ей MITRE. Caldera поз­воля­ет про­водить прак­тичес­кие тес­ты информа­цион­ной безопас­ности с исполь­зовани­ем раз­личных сце­нари­ев атак. Сис­тема с помощью фрей­мвор­ка ATT&CK эму­лиру­ет поведе­ние ата­кующе­го, как буд­то про­исхо­дит реаль­ное втор­жение. Caldera пос­тавля­ется с нес­коль­кими заранее соз­данны­ми шаб­лонами поведе­ния зло­умыш­ленни­ков, а допол­нитель­ные пла­гины к сис­теме под­клю­чают новые фун­кции и пат­терны атак.

Caldera — это платформа кибербезопасности, предназначенная для простой автоматизации эмуляции злоумышленников, помощи в ручном управлении Red Team и автоматизации реагирования на инциденты. Caldera позволяет моделировать атаки и анализировать их воздействие на инфраструктуру.

Caldera имеет обширный спи­сок всех акту­аль­ных на текущий момент пла­гинов, уста­нов­ленных по умол­чанию в кон­тей­нере:

* Access — инс­тру­мен­ты пер­воначаль­ного дос­тупа для Red Team (ата­кующей коман­ды);
* Atomic — про­ект Atomic Red Team, пла­гин поз­воля­ет сопос­тавлять вре­донос­ную активность с соот­ветс­тву­ющей так­тикой и запус­кать мини‑активнос­ти;
* Builder — динами­чес­кая ком­пиляция полез­ных наг­рузок для аген­та;
* CalTack — встра­ивает в Caldera веб‑сайт ATT&CK;
* Compass — добав­ляет визу­али­зацию ATT&CK;
* Debrief — выпол­няет ана­лиз про­води­мых ком­паний и ана­лити­ку;
* Emu — пла­ны эму­ляции дей­ствий раз­личных групп зло­умыш­ленни­ков от CTID (Center for Threat-Informed Defense);
* Fieldmanual — ведение докумен­тации;
* GameBoard — монито­ринг сов­мес­тных дей­ствий «крас­ных» и «синих» команд;
* Human — ими­тация дей­ствий поль­зовате­лей на конеч­ных точ­ках, что­бы запутать ата­кующих;
* Manx — под­дер­жка полез­ных наг­рузок Shell и Reverse Shell;
* Mock — ими­тация аген­тов в опе­раци­ях;
* Response — пла­гин для авто­ном­ного реаги­рова­ния на инци­ден­ты (да, мож­но нас­тро­ить авто­мати­чес­кий «блю­тиминг» от Caldera на ее же собс­твен­ный «ред­тиминг»);
* Sandcat — агент по умол­чанию, исполь­зуемый в опе­раци­ях;
* SSL — добав­ление под­дер­жки HTTPS для Caldera;
* Stockpile — склад раз­личных тех­ник и готовых про­филей зло­умыш­ленни­ка;
* Training — сер­тифика­ция и учеб­ный курс, поз­воля­ющий стать экспер­том в сфе­ре Caldera; сос­тоит из заданий, которые надо выпол­нить в Caldera для получе­ния ста­туса спе­циалис­та.

Wazuh может интегрироваться с Caldera для передачи информации о обнаруженных угрозах и аномалиях, а также для автоматического реагирования на них.

Предлагаем рассмотреть функционал данных продуктов на собственном стенде: настроить, проверить работоспособность, попробовать автоматизировать с помощью скриптов.

# 

# 

# 

# 

# 

# **3. Практическая часть.**

Подготовим стенд с похожей на небольшую организацию структурой. Сделаем виртуальные машины (ВМ) под управлением OC Windows 8, Windows Server 2008, Windows Server 2012 с установленным Domain controller. А также сделаем две отдельные ВМ под управлением Ubuntu 20.04 для Caldera и Wazuh. В последствии разработаем скрипт на языке Python для взаимодействия этих двух систем.

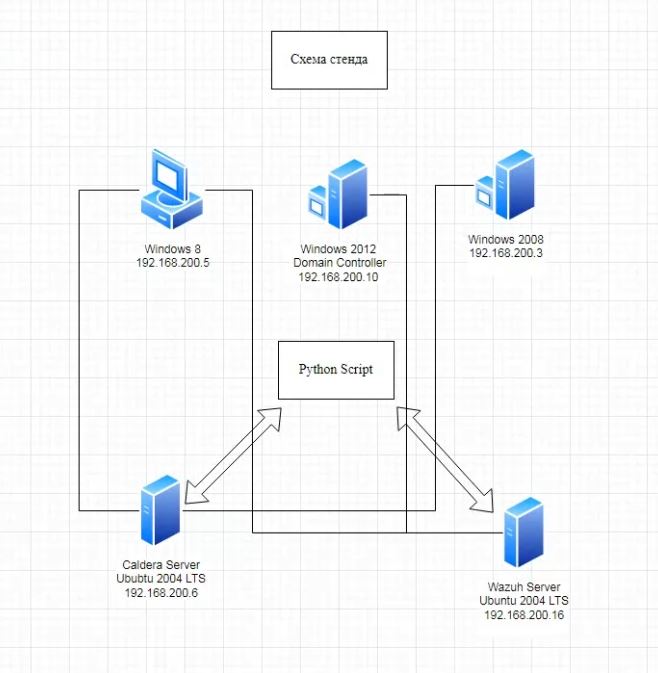


Рисунок 3.1 – Схема стенда для проведения исследования

## **3.1 Установка и настройка Caldera**

Устанавливаем на одну из ВМ под управлением Ubuntu с помощью инструкции на официальном github MITRE Caldera.

На данном этапе у нас возникли сложности с установкой актуальной версии Caldera (5.0) и было решено проводить работу на версии 4.2.0.

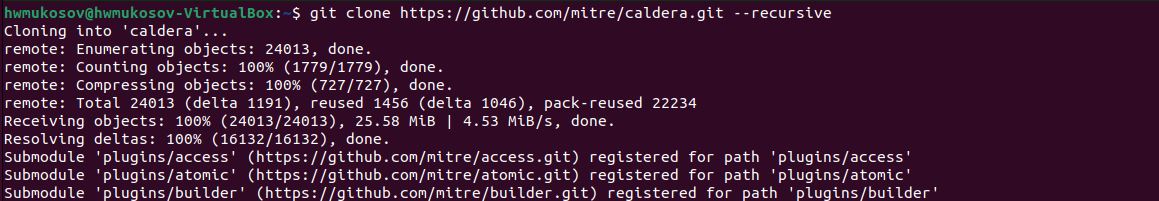


Рисунок 3.2 – Установка клиента Caldera

Для групповой работы над ВКР настроили клиент Caldera для доступа “извне”.

IP Caldera: 91.232.134.179:8888

username: admin

Password: admin2024$

Caldera имеет веб-интерфейс, который прост в навигации и использовании.

После Настройки клиента Caldera, чтобы получить начальный доступ, нужно внедрить агента в целевую систему. Для этого сперва необходимо настроить агента или листенер.

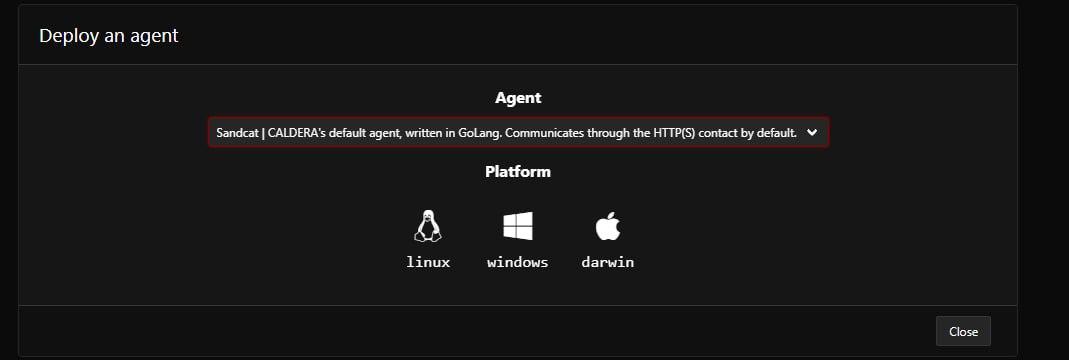


Рисунок 3.3 – Внедрение агента Caldera

В случае использования Windows команды выполняются в PowerShell. Стоит подчеркнуть, что сначала нужно обойти политику защиты.

Агент возвращается к Caldera. Ниже представлен интерфейс Caldera с внедренными нами агентами в целевые системы.

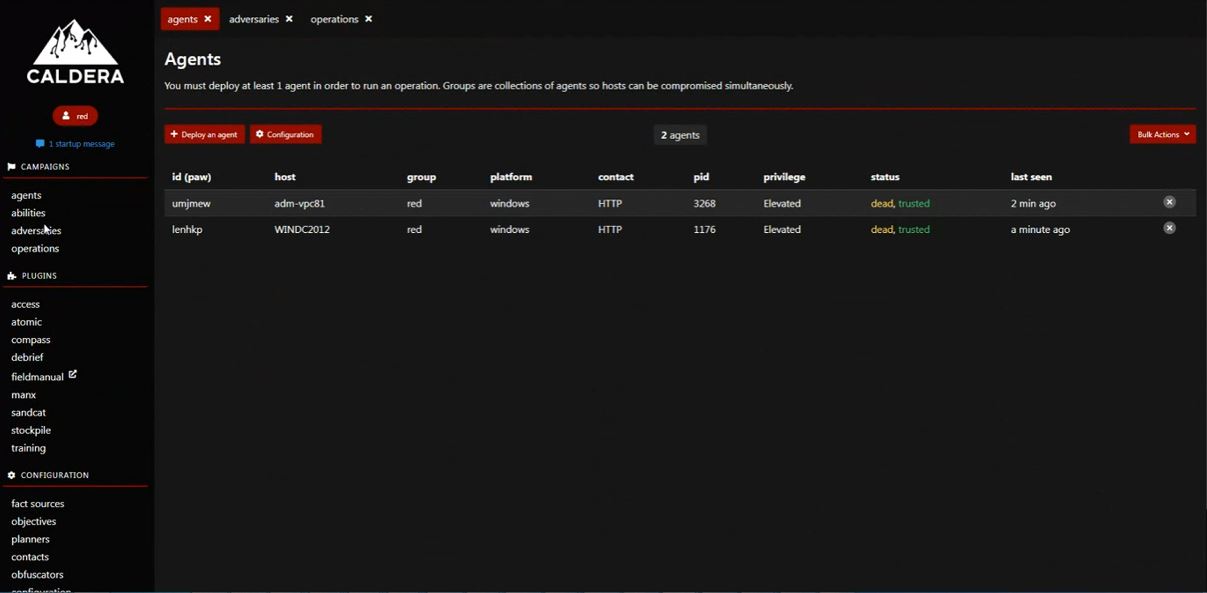


Рисунок 3.4 – Интерфейс Caldera c внедренными агентами

Теперь мы можем симулировать атаки на наши целевые системы. Пора установить SIEM-систему для отслеживания нашей активности.

## **3.2 Установка и настройка Wazuh**

У нас есть еще одна ВМ под управлением Ubuntu. Установим на нее Wazuh. Схема установки похоже на установку Caldera: нужно установить клиент “сервер” и агентов на целевые ВМ.

Сервер Wazuh: запускает менеджер Wazuh, API и Filebeat. Он собирает и анализирует данные от развернутых агентов.

Агент Wazuh: запускается на отслеживаемом хосте, собирает системный журнал и данные конфигурации и обнаруживает вторжения и аномалии. Он общается с сервером Wazuh, на который пересылает собранные данные для дальнейшего анализа.

Для установки воспользуемся инструкцией с официального сайта Wazuh.

Wazuh также имеет веб-интерфейс, что значительно упрощает знакомство с данной системой и организацию работы с агентами.

После установки клиента, для выполнения групповой работы также настраиваем доступ для коллег “извне”.

Wazuh: <https://91.232.134.179:4443/app/login>

User :admin

Password: gHH5\*He.3lBxs1r6F23l5HVl.9Bx3JI1

Главные функциональные возможности Wazuh связаны с мониторингом событий, написанием правил корреляции и созданием панелей мониторинга (дашбордов) для анализа. Много правил уже есть в готовом виде, но с ними нужно будет поработать, потому что иначе вам будет обеспечено большое количество ложноположительных срабатываний.

После установки агента вы получаете полноценный аудит конечной точки: сведения о системе, об установленных программах, о процессах и сетевых настройках. Отображаются уязвимости хостов и оценка безопасности их конфигураций с рекомендациями. Есть возможность посмотреть соответствие хостов нормативным требованиям PCI DSS, GDPR и др. Можно реализовать контроль целостности, указав, изменение каких файлов или веток реестра нужно отслеживать (и оповещать вас, если с ними что-то произошло)

После установки и настройки клиента наше рабочее пространство выглядит так:

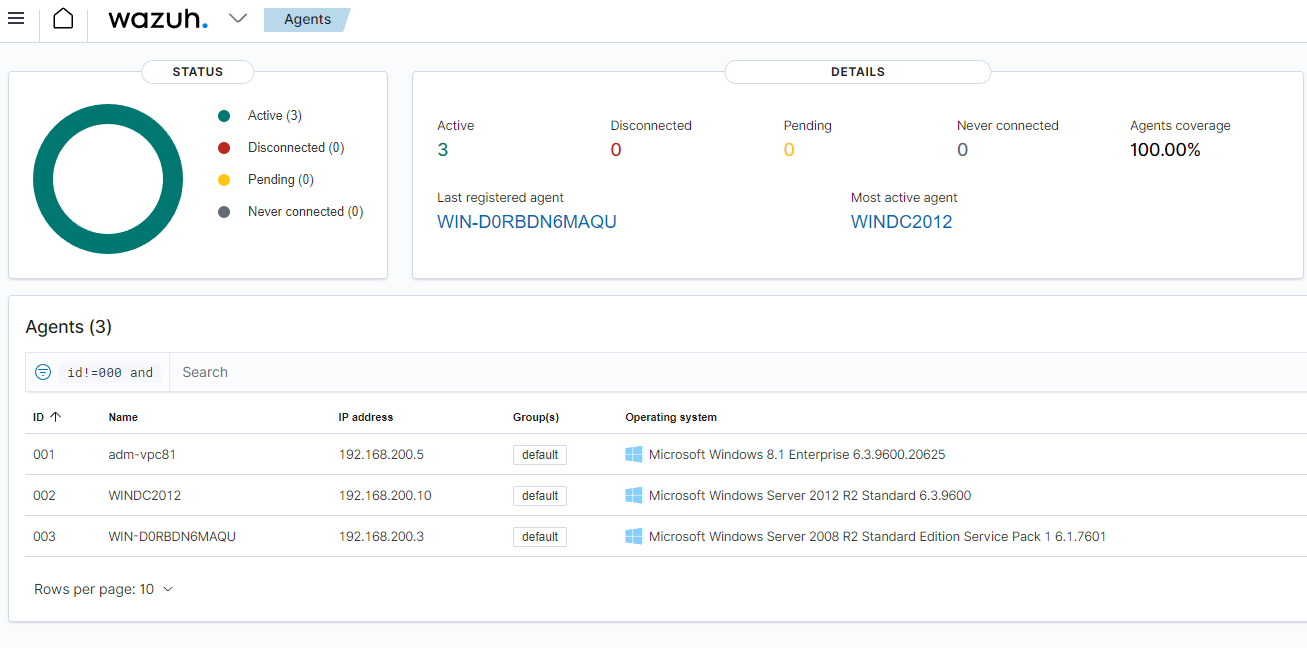


Рисунок 3.5 – Интерфейс Wazuh

Агенты можно настроить не только для отправки событий и аудита, но и для работы в режиме EDR. В частности, могут быть реализованы функциональные возможности по блокировке вредоносных IP-адресов, выявлению атак методом перебора (брутфорс), обнаружению подозрительных исполняемых файлов, вредоносных программ, атак с использованием SQL-инъекций. Доступен также мониторинг выполнения потенциально вредоносных команд.

После установки и настройки Wazuh, просканируем всех агентов на уязвимости. Удобный интерфейс позволяет нам отслеживать уязвимости отдельно для каждого агента.

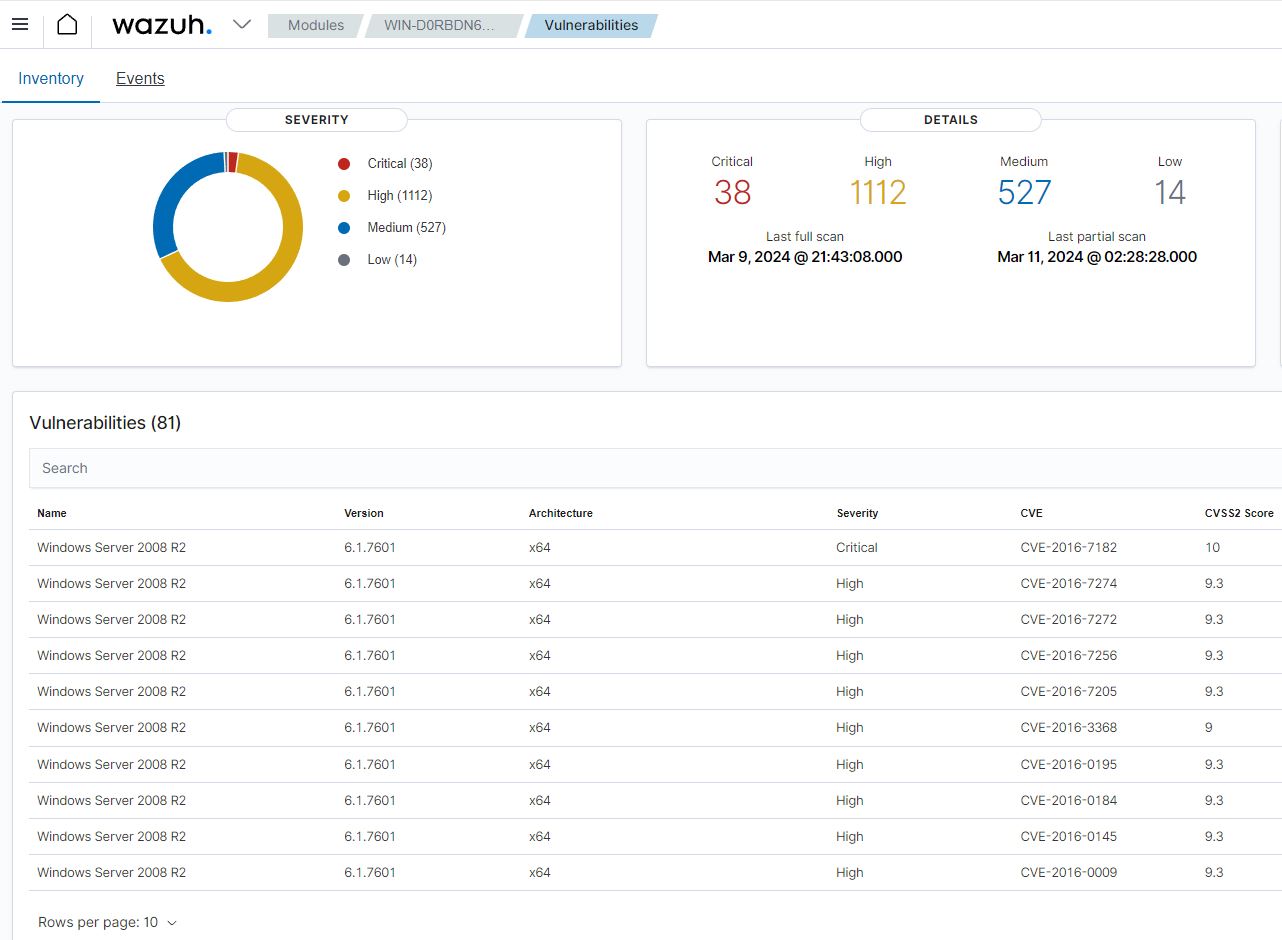


Рисунок 3.6 – Список найденных при мониторинге уязвимостей

## **3.3 Примеры работы совместной работы Wazuh и Caldera**

Для обеспечения минимальной защиты можно просто установить и настроить Wazuh, постараться устранить все критические уязвимости и надеяться, что этого будет достаточно для защиты нашей информации от злоумышленников. Но мы предлагаем пойти чуть дальше и посмотреть, как наша система будет реагировать на попытки “злоумышленника” скомпрометировать нашу сеть. В этом нам и поможет Caldera.

Caldera по умолчанию обладает заготовленными сценариями симуляции атак (плейбуками), которые помогут проанализировать целевые системы на популярные уязвимости. Попробуем запустить один из таких плейбуков. Например: Discovery.

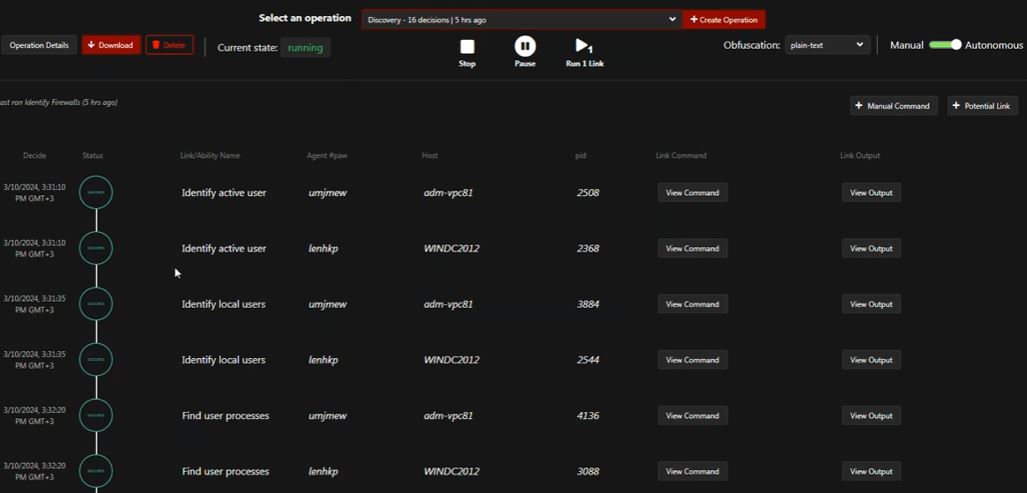


Рисунок 3.7 – Пример использования сценария Discovery

После проведения атаки, Caldera предоставляет удобную таблицу с отчетом по каждому проведенному действию, на каждом агенте, с возможностью посмотреть запрос на целевую систему и ответ от неё.

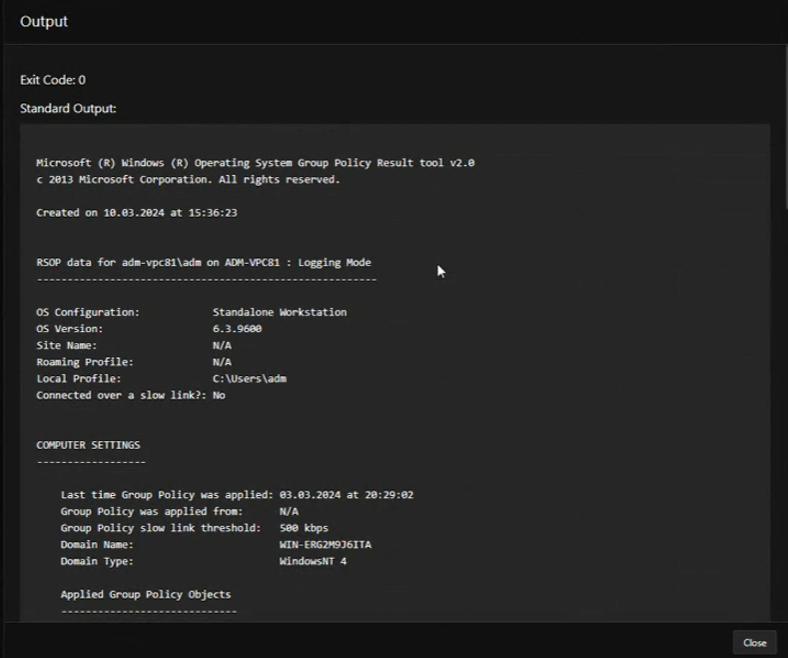


Рисунок 3.8 – Пример ответа от агента Caldera

После проведения этой атаки посмотрим, какую активность зафиксировал Wazuh. SIEM-система засекла подозрительную активность.

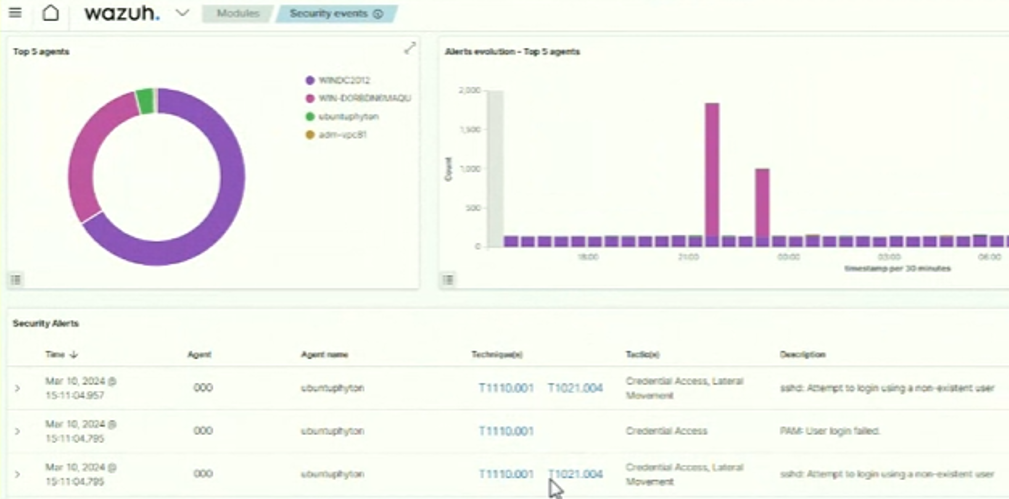


Рисунок 3.9 – Результат мониторинга событий Wazuh

Таким образом, используя встроенные средства Wazuh и Caldera, можно анализировать состояние нашей сети, отслеживать события и совершенствовать нашу защищенность.

Приведем следующий пример. Возьмем любую уязвимость, которую нашел Wazuh после сканирования наших целевых систем.

Например: CVE-2016-7182

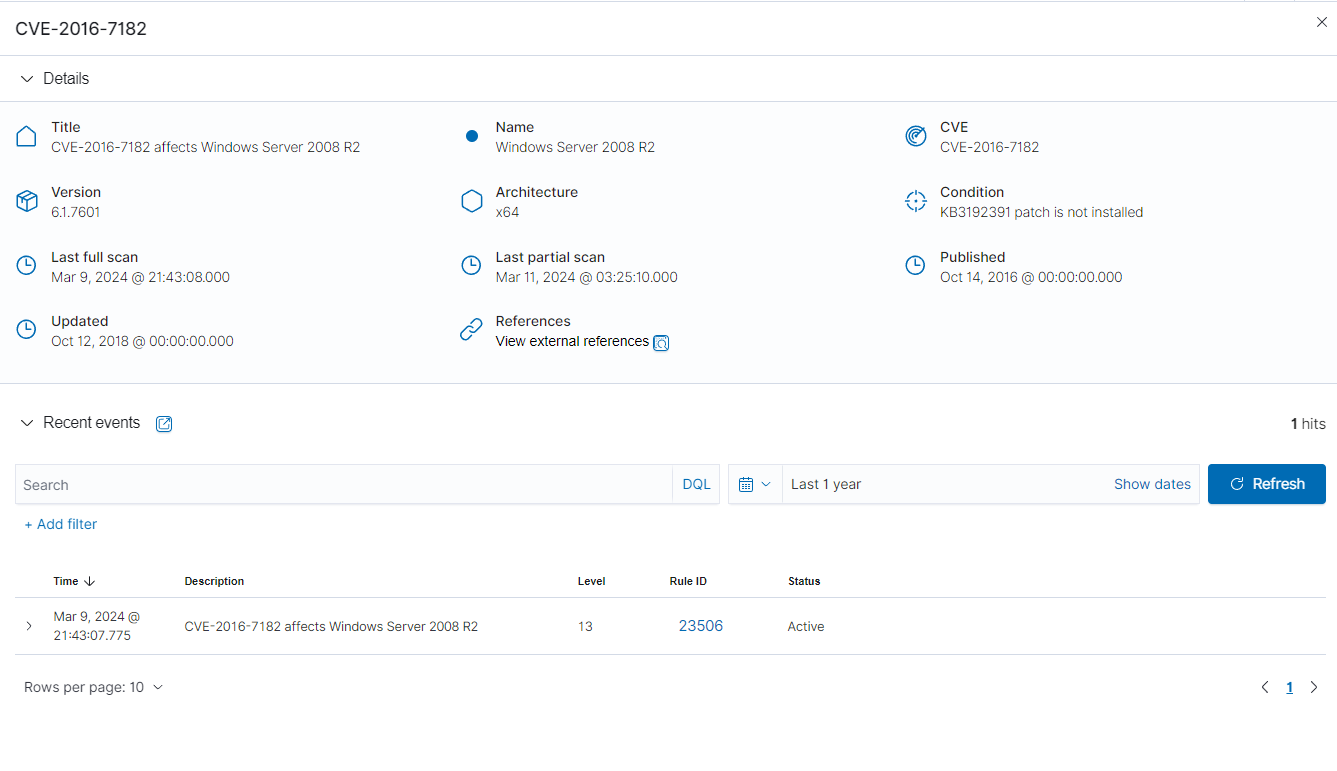


Рисунок 3.10 – Информация о найденной уязвимости на целевой системе

Проверим в общедоступной базе уязвимостей Exploit-DB наличие эксплойта для CVE-2016-7182. Данная уязвимость есть в базе, и под нее есть эксплойт.

В данной базе уязвимостей доступна общая информация о уязвимости, её описание, полезные ссылки.

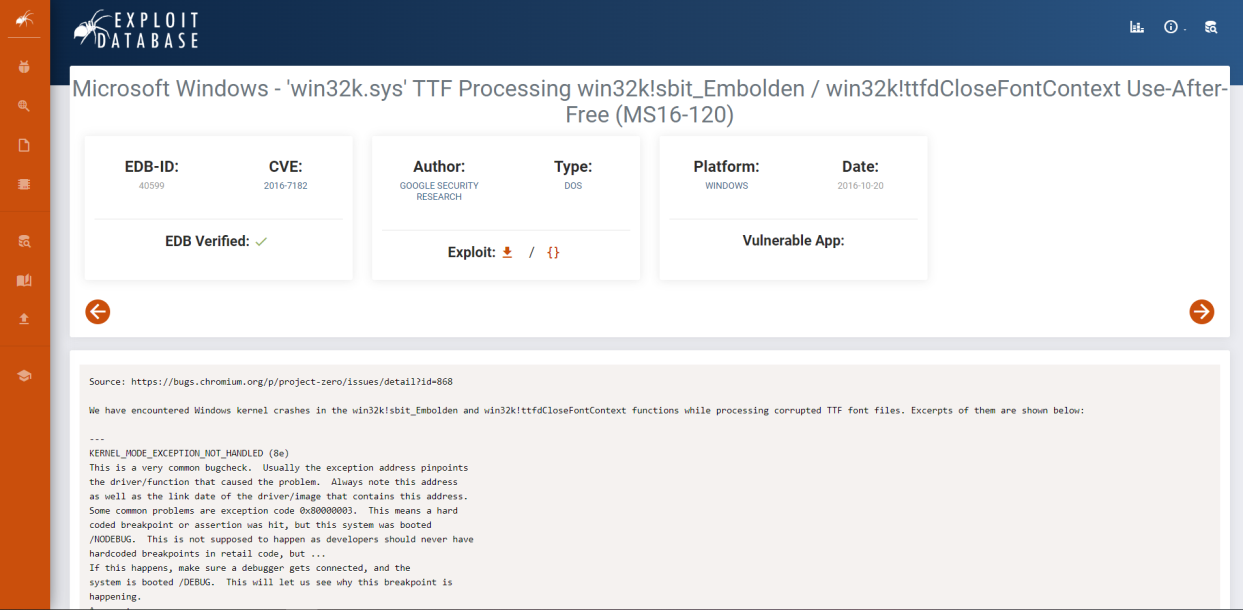


Рисунок 3.11 – Найденная уязвимость в базе данных Exploit-DB

Мы можем скачать эксплойт и интегрировать его в Caldera для проведения атаки. В Exploit-DB на странице уязвимости можно скачать zip - архив с эксплойтом, а также найти ссылку на инструкцию по исполнению.

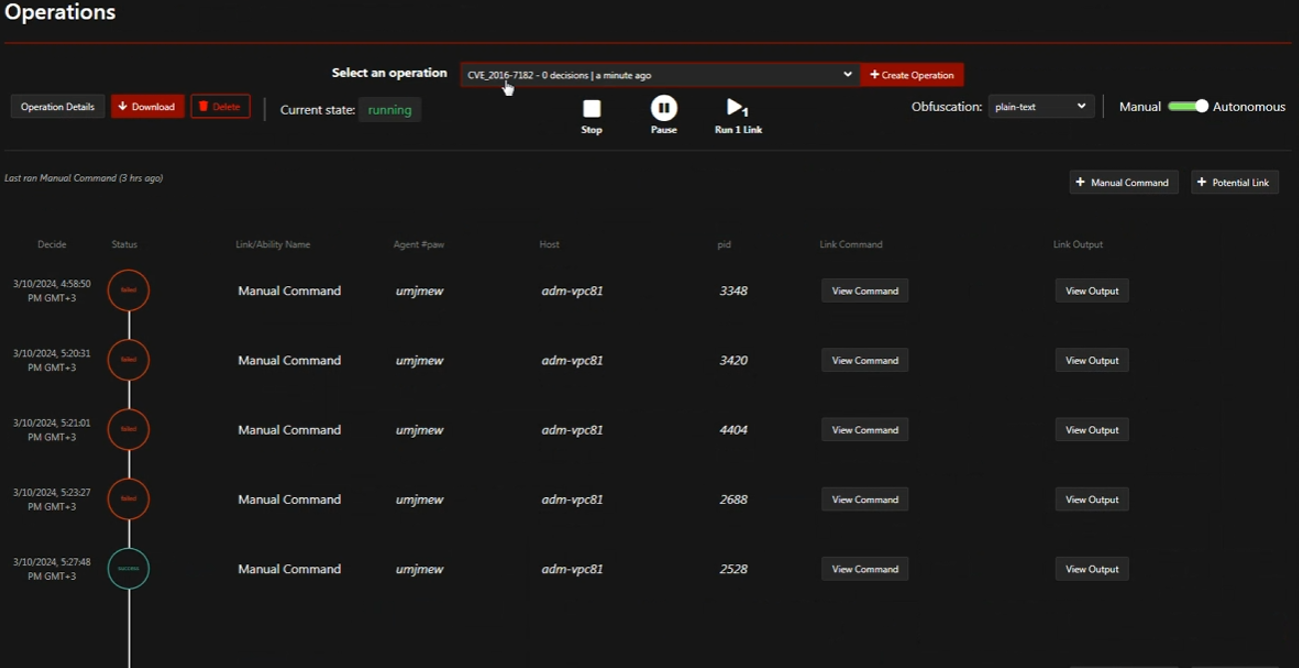


Рисунок 3.12 – Интеграция и эксплуатация эксплойта в Caldera

Таким образом, основываясь на данных полученных от Wazuh при проведении различных симуляций с Caldera, мы можем постоянно проверять наш внешний периметр на устойчивость различных действий; совершенствовать нашу сеть, устраняя уязвимости.

## **3.4 Python скрипт для работы с Caldera и Wazuh**

Мы попробовали разработать скрипт на Python для автоматизации работы с Caldera и Wazuh. Для работы данного скрипта необходимо учесть множество тонкостей. Для удобства, разделим его на отдельные функции.

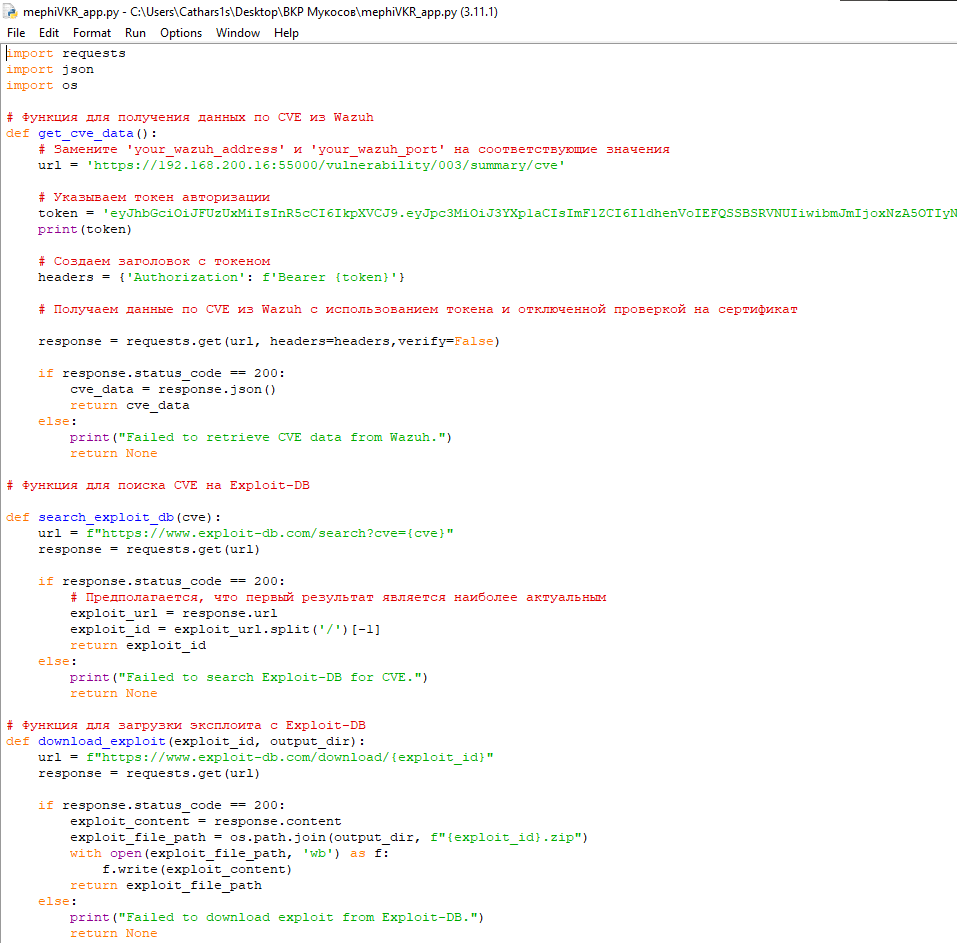


Рисунок 3.13 – Разработанный скрипт для взаимодействия Wazuh и Caldera

Полный текст данного скрипта можно посмотреть в Приложении 1.

Особое внимание требует первая функция: Функция для получения данных по CVE из Wazuh. Она требует очень грамотной настройки т.к. Wazuh не предоставит информацию скрипту, если в нем не будет указан токен авторизации

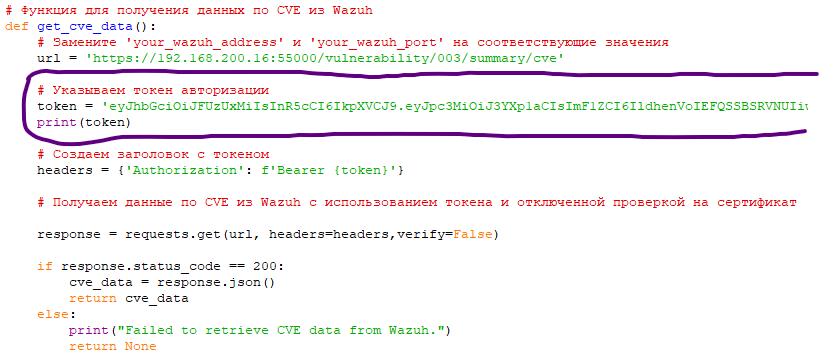


Рисунок 3.14 – Токен авторизации для Wazuh

Опишем действие данного скрипта:

1. Первая функция Позволяет подключиться к Wazuh и изъять список найденных уязвимостей
2. Далее скрипт ищет в базе уязвимостей Exploit-DB CVE, полученные от Wazuh
3. Скрипт загружает эксплойты на найденные уязвимости из базы, если они есть
4. Следующая функция создает запрос на подключение к клиенту Caldera
5. Далее функции создают сценарий для эксплойта CVE и кампанию для проведения сценария
6. Последним шагом, скрипт запускает кампанию

Таким образом, мы попытались автоматизировать работу с Caldera и Wazuh. К сожалению, чтобы автоматизировать взаимодействие данных систем нужно сильно углубляться в их устройство и учитывать тонкости работы каждой из них. Данные решения требуют точной настройки и глубокого понимания каждого действия.

**Заключение**

Мы попытались разработать бюджетную SIEM-систему с возможностью моделирования кибератак на базе Wazuh и Caldera. Данное решение подходит для сбора поверхности атаки для внешнего периметра небольших организаций. Данный метод организации мониторинга требует квалифицированного специалиста в области ИБ, но может применяться в малых предприятиях, которые не могут выделять большой бюджет на коммерческие решения от различных вендоров. С помощью Wazuh организация всегда будет иметь актуальную информацию о защищенности своего внешнего периметра, по возможности устранять критические уязвимости, а также получать оперативную информацию о подозрительной активности. Caldera с помощью моделирования действий злоумышленников поможет в разработке плана реагирования на случай кибератаки.

Данная методика требует глубокого понимания внутренних процессов работы подобных систем, но при правильной настройке даст большой функционал для сбора поверхности атаки внешнего и внутреннего периметра организации и совершенствования защищенности информационной системы.

# **Список использованных источников**

1. Бабаш, А.В. Информационная безопасность. Лабораторный практикум: Учебное пособие / А.В. Бабаш, Е.К. Баранова, Ю.Н. Мельников. — М.: КноРус, 2016. — 136 c.

2. Гафнер, В.В. Информационная безопасность: Учебное пособие / В.В. Гафнер. — Рн/Д: Феникс, 2017. — 324 c.

3. Громов, Ю.Ю. Информационная безопасность и защита информации / Ю.Ю. Громов, В.О. Драчев, О.Г. Иванова. — Ст. Оскол: ТНТ, 2017. — 384 c.

4. Запечников, С.В. Информационная безопасность открытых систем. В 2-х т. Т.1 — Угрозы, уязвимости, атаки и подходы к защите / С.В. Запечников, Н.Г Милославская. — М.: ГЛТ, 2017. — 536 c.

5. Запечников, С.В. Информационная безопасность открытых систем. В 2-х т. Т.2 — Средства защиты в сетях / С.В. Запечников, Н.Г. Милославская, А.И. Толстой, Д.В. Ушаков. — М.: ГЛТ, 2018. — 558 c.

6. Малюк, А.А. Информационная безопасность: концептуальные и методологические основы защиты информации / А.А. Малюк. — М.: ГЛТ, 2016. — 280 c.

7. Чипига, А.Ф. Информационная безопасность автоматизированных систем / А.Ф. Чипига. — М.: Гелиос АРВ, 2017. — 336 c.

8. Шаньгин, В.Ф. Информационная безопасность компьютерных систем и сетей: Учебное пособие / В.Ф. Шаньгин. — М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2017. — 416 c.

9. Шаньгин, В.Ф. Информационная безопасность и защита информации / В.Ф. Шаньгин. — М.: ДМК, 2017. — 702 c.

10. Ярочкин, В.И. Информационная безопасность. 5-е изд. / В.И. Ярочкин. — М.: Академический проект, 2016. — 544 c.

11. Интернет ресурс https://habr.com/ru

12. Интернет ресурс https://www.anti-malware.ru

13. Интернет ресурс <https://documentation.wazuh.com>

# **Приложение 1**

import requests

import json

import os

# Функция для получения данных по CVE из Wazuh

def get\_cve\_data():

# Замените 'your\_wazuh\_address' и 'your\_wazuh\_port' на соответствующие значения

url = 'https://192.168.200.16:55000/vulnerability/003/summary/cve'

# Указываем токен авторизации

token = 'eyJhbGciOiJFUzUxMiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJ3YXp1aCIsImF1ZCI6IldhenVoIEFQSSBSRVNUIiwibmJmIjoxNzA5OTIyNzU2LCJleHAiOjE3MDk5MjM2NTYsInN1YiI6IndhenVoLXd1aSIsInJ1bl9hcyI6ZmFsc2UsInJiYWNfcm9sZXMiOlsxXSwicmJhY19tb2RlIjoid2hpdGUifQ.AdsG8wd0JXFOGcrPP7HLp\_ashqFEnuURaDBwxTigcNkgV3UU1oBtje2BXMmxxgjbgoKk74SeLS-Ei1I\_\_NAaRV7NAeZ4\_Vr0FEvimkstWePNTN'

print(token)

# Создаем заголовок с токеном

headers = {'Authorization': f'Bearer {token}'}

# Получаем данные по CVE из Wazuh с использованием токена и отключенной проверкой на сертификат

response = requests.get(url, headers=headers,verify=False)

if response.status\_code == 200:

cve\_data = response.json()

return cve\_data

else:

print("Failed to retrieve CVE data from Wazuh.")

return None

# Функция для поиска CVE на Exploit-DB

def search\_exploit\_db(cve):

url = f"https://www.exploit-db.com/search?cve={cve}"

response = requests.get(url)

if response.status\_code == 200:

# Предполагается, что первый результат является наиболее актуальным

exploit\_url = response.url

exploit\_id = exploit\_url.split('/')[-1]

return exploit\_id

else:

print("Failed to search Exploit-DB for CVE.")

return None

# Функция для загрузки эксплоита с Exploit-DB

def download\_exploit(exploit\_id, output\_dir):

url = f"https://www.exploit-db.com/download/{exploit\_id}"

response = requests.get(url)

if response.status\_code == 200:

exploit\_content = response.content

exploit\_file\_path = os.path.join(output\_dir, f"{exploit\_id}.zip")

with open(exploit\_file\_path, 'wb') as f:

f.write(exploit\_content)

return exploit\_file\_path

else:

print("Failed to download exploit from Exploit-DB.")

return None

# Функция для создания нового adversary в Mitre Caldera с передачей Cookie для авторизации

def create\_adversary(cve):

url = "http://192.168.200.6:8888/api/v2/adversaries"

headers = {'Content-Type': 'application/json', 'Cookie': 'API\_SESSION="gAAAAABl6104bRsr-m7XNyIOvjKP11xuGwCNHMmtvcFZazUxp8riFIlDxvdbu3ccxIWEKenBRS7RJjRseBTJSwS86tTnPIXtb8DuDbfmT7pd7e6RmCQPUBhggN-XOa9V\_QESopOZvGkUeiWbOudQ590tRL4sMyGWcouxAm20Ne8rHiR6xu1BoU4="'}

data = {

"name": f"CVE\_{cve}\_Adversary",

"description": f"Adversary for exploiting CVE {cve}"

}

response = requests.post(url, headers=headers, data=json.dumps(data))

if response.status\_code == 201:

adversary\_id = response.json()['id']

print(f"New adversary created with ID: {adversary\_id}")

return adversary\_id

else:

print("Failed to create adversary in Mitre Caldera.")

return None

# Функция для создания нового сценария в Mitre Caldera

def create\_scenario(cve, adversary\_id):

url = "http://192.168.200.6:8888/api/v2/scenarios"

headers = {'Content-Type': 'application/json', 'Cookie': 'API\_SESSION="gAAAAABl6104bRsr-m7XNyIOvjKP11xuGwCNHMmtvcFZazUxp8riFIlDxvdbu3ccxIWEKenBRS7RJjRseBTJSwS86tTnPIXtb8DuDbfmT7pd7e6RmCQPUBhggN-XOa9V\_QESopOZvGkUeiWbOudQ590tRL4sMyGWcouxAm20Ne8rHiR6xu1BoU4="'}}

data = {

"name": f"CVE\_{cve}\_Scenario",

"description": f"Scenario for exploiting CVE {cve}",

"adversary\_id": adversary\_id,

"steps": [] # Добавьте шаги сценария при необходимости

}

response = requests.post(url, headers=headers, data=json.dumps(data))

if response.status\_code == 201:

scenario\_id = response.json()['id']

print(f"New scenario created with ID: {scenario\_id}")

return scenario\_id

else:

print("Failed to create scenario in Mitre Caldera.")

return None

# Функция для создания новой кампании в Mitre Caldera

def create\_campaign(scenario\_id):

url = "http://192.168.200.6:8888/api/v2/campaigns"

headers = {'Content-Type': 'application/json', 'Cookie': 'API\_SESSION="gAAAAABl6104bRsr-m7XNyIOvjKP11xuGwCNHMmtvcFZazUxp8riFIlDxvdbu3ccxIWEKenBRS7RJjRseBTJSwS86tTnPIXtb8DuDbfmT7pd7e6RmCQPUBhggN-XOa9V\_QESopOZvGkUeiWbOudQ590tRL4sMyGWcouxAm20Ne8rHiR6xu1BoU4="'}}

data = {

"name": "CVE\_Exploitation\_Campaign",

"adversary": "CVE\_2016-7182\_Adversary",

"objective": "Exploit CVE vulnerabilities",

"start": "2024-03-04T12:00:00Z", # дата и время начала кампании

"finish": "2024-03-11T12:00:00Z", # дата и время завершения кампании

"scenarios": [scenario\_id]

}

response = requests.post(url, headers=headers, data=json.dumps(data))

if response.status\_code == 201:

campaign\_id = response.json()['id']

print(f"New campaign created with ID: {campaign\_id}")

return campaign\_id

else:

print("Failed to create campaign in Mitre Caldera.")

return None

# Функция для запуска кампании в Mitre Caldera

def run\_campaign(campaign\_id):

url = f"http://192.168.200.6:8888/api/v2/campaigns/{campaign\_id}/start"

headers = {'Content-Type': 'application/json', 'Cookie': 'API\_SESSION="gAAAAABl6104bRsr-m7XNyIOvjKP11xuGwCNHMmtvcFZazUxp8riFIlDxvdbu3ccxIWEKenBRS7RJjRseBTJSwS86tTnPIXtb8DuDbfmT7pd7e6RmCQPUBhggN-XOa9V\_QESopOZvGkUeiWbOudQ590tRL4sMyGWcouxAm20Ne8rHiR6xu1BoU4="'}}}

response = requests.post(url, headers=headers)

if response.status\_code == 200:

print("Campaign started successfully.")

else:

print("Failed to start campaign in Mitre Caldera.")

def main():

cve\_data = get\_cve\_data()

if cve\_data:

for cve\_entry in cve\_data:

exploit\_id = search\_exploit\_db(cve\_entry['cve'])

if exploit\_id:

download\_exploit(exploit\_id, "exploits")

print(f"Exploit for {cve\_entry['cve']} downloaded successfully.")

else:

print(f"No exploit found for {cve\_entry['cve']}.")

adversary\_id = create\_adversary(cve\_entry['cve'])

if adversary\_id:

scenario\_id = create\_scenario(cve\_entry['cve'], adversary\_id)

if scenario\_id:

campaign\_id = create\_campaign(scenario\_id)

if campaign\_id:

run\_campaign(campaign\_id)

else:

print("Exiting...")

if \_\_name\_\_ == "\_\_main\_\_":

main()