# Введение

В настоящее время все больше компаний используют цифровые технологии для эффективного функционирования. Например:

* прием и обработка заказов;
* отслеживание загруженности складских помещений;
* отслеживание корпоративного транспорта и логистических цепочек;
* онлайн банкинг
* хранение большого количество различной информации в облачных хранилищах, например, финансовые отчеты, код разрабатываемых программ, данные клиентов, внутреннее общение сотрудников.

Все эти данные являются представляют интерес для злоумышленников, с целью получить выгоду, в связи с этим количество компьютерных атак возрастает.

Атаки проводятся по разным причинам, например, с целью:

* парализовать бизнес конкурента, тем самым переманивая к себе его клиентов.
* кражи конфиденциальных данных. Это могут быть как данные клиентов компании, так и внутренние документы, например отчеты по финансам или схема инфраструктуры компании. Далее такая информация продается на черном рынке, или используется для получения выкупа за неразглашение этой информации.
* промышленного шантажа

Так же компания, подвергшаяся нападению, несет репутационные потери.

Одна из мер уменьшения потерь при потенциальном взломе инфраструктуры компании – своевременное обнаружение злоумышленников в системе и остановка атаки на ранних этапах. Для этих целей есть множество средств, одно из них - Security information and event management (SIEM), оно и будет рассмотрено в данной работе.

# Глава 1 - Анализ

Для начала давайте рассмотрим какие решения существуют для обнаружения компьютерных атак ([взято тут](https://safe-surf.ru/specialists/article/5274/656701/#:~:text=%D0%A1%D0%BE%D0%B2%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D0%BD%D1%8B%D0%B5%20%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%B4%D0%BB%D1%8F%20%D0%BE%D0%B1%D0%BD%D0%B0%D1%80%D1%83%D0%B6%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D1%85,%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B%20%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%20%D1%81%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B3%D0%BE%20%D1%82%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0%20(NTA))):

* Межсетевые экраны нового поколения (NGFW)
* Системы мониторинга событий безопасности (SIEM)
* Xdr
* epp
* Средства обнаружения компьютерных атак на конечных устройствах (EDR)

Рассмотрим некоторые технологии подробнее:

Межсетевой экран следующего поколения (NGFW) - это межсетевой экран с глубокой фильтрацией, интегрированный с системой обнаружения вторжений (IDS) или системой предотвращения вторжений (IPS) и способный контролировать и блокировать трафик на уровне приложений.

NGFW также позволяют осуществлять микросегментацию сети на основе приложений, а не только портов и IP-адресов. Обычно они поставляются в виде отдельных устройств, но существуют межсетевые экраны нового поколения в виде виртуальной машины или облачной службы.

Endpoint Detection & Response (EDR) — класс решений для обнаружения и изучения вредоносной активности на конечных точках: подключенных к сети рабочих станциях, серверах, устройствах Интернета вещей и так далее. В отличие от антивирусов, задача которых — бороться с типовыми и массовыми угрозами, EDR-решения ориентированы на выявление целевых атак и сложных угроз. При этом EDR-решения не могут полностью заменить антивирусы (EPP), поскольку эти две технологии решают разные задачи.

Платформа защиты конечных устройств (EPP) часто включает в себя антивирусы, но при этом предлагает несколько дополнительных ключевых функций. Во-первых, она добавляет машинное обучение для поддержки поведенческого анализа, который расширяет традиционный мониторинг угроз за пределы известных угроз. Эта возможность позволяет EPP предотвращать неизвестные атаки в дополнение к тем, которые встречаются чаще всего. EPP также проверяет индикаторы компрометации (IoC) и мониторит память устройства для выявления нестандартных паттернов потребления памяти.

EPP лучше базовой антивирусной защиты для повсеместного управления конечными точками и предотвращения угроз в крупных компаниях, однако некоторые сложные атаки все еще способны ускользнуть от обнаружения. Стоит также отметить, что хотя EPP полезен для выявления уязвимостей и предотвращения атак, он не способен принять меры по устранению активных угроз, которые проникают на конечные точки.

Extended detection and response (XDR) был разработан как альтернатива точечным решениям безопасности, которые были ограничены только одним уровнем безопасности или могли выполнять только корреляцию событий без реагирования. Это эволюция таких решений, как Endpoint Detection & Response (EDR) и анализ сетевого трафика (NTA).

Несмотря на свою полезность, эти инструменты, ориентированные на конкретные уровни, обычно генерируют большее количество предупреждений, требуют больше времени на расследование и реагирование на события, а также требуют большего обслуживания и управления. В отличие от них, XDR консолидирует инструменты и позволяет командам безопасности работать более эффективно и результативно.

Объединяя security information management (SIM) и security event management (SEM), security information and event management (SIEM) предлагает мониторинг и анализ событий в режиме реального времени, а также отслеживание и регистрацию данных о безопасности в целях соответствия нормативным требованиям.

Проще говоря, SIEM - это решение для обеспечения безопасности, которое помогает организациям распознавать потенциальные угрозы безопасности и уязвимости до того, как они успеют нарушить работу бизнеса. Оно выявляет аномалии в поведении пользователей и использует искусственный интеллект для автоматизации многих ручных процессов, связанных с обнаружением угроз и реагированием на инциденты, и стало основным элементом современных операционных центров безопасности (SOC) для обеспечения безопасности и управления соответствием нормативным требованиям.

С годами SIEM стал более зрелым, чем предшествующие ему инструменты управления журналами.

Есть несколько готовых решений от вендоров, представленных следующими компаниями: HP, RT Solar, PT, IBM, McAfee, КОМРАД. Основным преимуществом готового решения будет быстрое внедрение в вашу инфраструктуру, а также качественная поддержка, вплоть до выделения нескольких людей для полноценной работы в вашей компании. Однако в зависимости от размеров организации, внедрение и поддержка SIEM-системы может стоить десятки миллионов рублей. По этой причине не каждая компания может себе позволить столь большие траты. В таком случае следует обратить внимание на бесплатные варианты, на основе которых, можно построить SIEM. Ниже представлены доступные open-source решения:

* OSSIM
* OSSEC
* Sagan
* Wazuh
* ELK Stack

В данной работе будет рассмотрено решение от компании Elastic. Преимуществом ELK Stack является флагманский продукт компании Elastic - Elasticsearch. Оно используется для хранения событий. Это современная поисковая и аналитическая система, основанная на Apache Lucene, совмещенная с NoSQL базой данных. Преимущество заключается в том, что помимо SIEM системы, компания также получает отличное решение для сбора и обработки логов.

Для представления и визуализации этих событий используется Kibana. Kibana – это инструмент визуализации и изучения данных, который применяется для таких задач, как анализ журналов и временных рядов, мониторинг приложений и текущих процессов. Он предлагает мощные и простые в использовании возможности: гистограммы, линейные графики, круговые диаграммы, тепловые карты и встроенную геопространственную поддержку.

Для сбора информации из различных источников применяется Logstash. В его задачи входит преобразование данных, полученных от источников, в удобный для работы формат и направление их в хранилище для дальнейшей работы. Простота использования и возможность работать с большими объемами данных обеспечивает Logstash ряд преимуществ перед аналогичными проектами. Однако в данной работе Logstash использоваться не будет, так как это демо стенд, соответственно кол-во источников мало, а преобразование/обогащение лог-данных не требуется.

События, которые хранит Elasticsearch, приходят от различных утилит (называемых обобщенно Beats), установленных на клиентских ПК.

Beats - это программы, которые запускаются на клиентских устройствах, после чего отправляют соответствующие данные на сервер.

* Auditbeat
* Filebeat
* Functionbeat
* Heartbeat
* Metricbeat
* Packetbeat
* Winlogbeat

Auditbeat - это облегченный сборщик журналов аудита, который может собирать данные из среды аудита Linux и контролировать целостность файлов. Он может объединять связанные сообщения в событие для генерации стандартных структурированных данных для удобного анализа.

Filebeat — это легковесный доставщик лог-сообщений. Принцип его работы состоит в мониторинге и сборе лог-сообщений из лог-файлов и пересылке их в elasticsearch или logstash для индексирования.

Functionbeat - это новое дополнение к набору продуктов Beats, которое может быть легко развернуто как функция в serverless вычислительных платформах, предоставляя все необходимое для настройки, развертывания и преобразования событий, таких как журналы и метрики, получаемые из вашей облачной инфраструктуры.

Heartbeat - это легковесный доставщик информации о доступности сервисов с помощью активного сканирования. Также он передает время отклика на запрос.

Metricbeat собирает и отправляет различные метрики системного уровня для различных систем и платформ. Вы можете настроить частоту, с которой Metricbeat собирает метрики, и какие именно метрики собирать.

Packetbeat - это первый разработанный Beat, он выполняет функцию анализатора сетевых пакетов. Packetbeat захватывает сетевой трафик между серверами и поэтому может использоваться для мониторинга приложений и производительности.

Winlogbeat - это инструмент, разработанный специально для сбора журналов событий Windows. Его назначение - сбор и анализ событий безопасности, установленных обновлений и т.д.

В данной работе будут использованы следующие утилиты: Filebeat, Winlogbeat, Packetbeat.

Изначально ELK Stack позволяет лишь собирать логи/события, просматривать их в текстовом формате или визуализировать их с помощью диаграмм. Однако добавив проверку приходящих данных по определенным правилам, появляется возможность обнаруживать неправомерные действия на устройствах сотрудников или серверах компании. Таким образом мы получаем SIEM систему из бесплатных компонентов[1].

Развертывание SIEM-системы на основе Elastic и Kibana (EK) может происходить как на сервере (виртуальном или физическом) под управлением ОС Linux, так и в Docker контейнерах. Второй вариант может быть проще для деплоя, если запускать уже подготовленные компанией Elastic образы. Это связано с тем, что отпадает надобность в глубоком изучении инструкции по установки системы, и можно сконцентрироваться на ее конфигурировании.

Docker - это платформа контейнеризации с открытым исходным кодом. Он позволяет разработчикам упаковывать приложения в контейнеры - стандартизированные исполняемые компоненты, объединяющие исходный код приложения с библиотеками операционной системы (ОС) и зависимостями, необходимыми для запуска этого кода в любой среде. Контейнеры упрощают доставку распределенных приложений и становятся все более популярными по мере перехода организаций к разработке на основе облачных технологий и гибридным многооблачным средам.

Разработчики могут создавать контейнеры и без Docker, но эта платформа облегчает, упрощает и повышает безопасность создания, развертывания и управления контейнерами. Docker - это набор инструментов, позволяющий разработчикам создавать, развертывать, запускать, обновлять и останавливать контейнеры с помощью простых команд и экономящей работу автоматизации через единый API.

Docker относится к компании Docker, Inc., которая продает коммерческую версию Docker, и к проекту с открытым исходным кодом Docker, в который Docker, Inc. и многие другие организации или частные лица вносят свой вклад.

Контейнеры стали возможны благодаря изоляции процессов и возможностям виртуализации, встроенным в ядро Linux. Эти возможности - такие как группы управления (Cgroups) для распределения ресурсов между процессами, и пространства имен для ограничения доступа или видимости процессов к другим ресурсам или областям системы - позволяют нескольким компонентам приложений совместно использовать ресурсы одного экземпляра операционной системы хоста примерно так же, как гипервизор позволяет нескольким виртуальным машинам (VM) совместно использовать процессор, память и другие ресурсы одного аппаратного сервера.

В результате контейнерная технология предлагает все функциональные возможности и преимущества виртуальных машин - включая изоляцию приложений и экономически эффективную масштабируемость, плюс важные дополнительные преимущества такие как:

* Малый вес: В отличие от ВМ, контейнеры не несут полезной нагрузки в виде целого экземпляра ОС и гипервизора; они включают только процессы ОС и зависимости, необходимые для выполнения кода. Размеры контейнеров измеряются в мегабайтах (по сравнению с гигабайтами для некоторых ВМ), они лучше используют аппаратные мощности и быстрее запускаются.
* Более высокая эффективность использования ресурсов: С помощью контейнеров можно запускать в несколько раз больше копий приложения на том же оборудовании, чем при использовании виртуальных машин. Это позволяет сократить расходы на облако.
* Увеличение удобства разработки: По сравнению с виртуальными машинами контейнеры быстрее и проще развертывать, предоставлять и перезапускать. Это делает их идеальными для использования в системах непрерывной интеграции и непрерывной доставки (CI/CD) и лучше подходит для команд разработчиков, использующих практики Agile и DevOps.

Компании, использующие контейнеры, сообщают о других преимуществах, включая повышение качества приложений, более быстрое реагирование на изменения рынка и многое другое.

Сегодня Docker настолько популярен, что слова "Docker" и "контейнеры" используются как взаимозаменяемые. Но первые технологии, связанные с контейнерами, были доступны в течение многих лет - даже десятилетий - до того, как Docker был выпущен на всеобщее обозрение в 2013 году.

Наиболее примечательно, что в 2008 году в ядро Linux была внедрена технология LinuXContainers (LXC), полностью обеспечивающая виртуализацию для одного экземпляра Linux. Хотя LXC используется и сегодня, существуют и более новые технологии, использующие ядро Linux.

Docker расширил возможности контейнеризации в Linux с помощью технологий, которые позволяют:

* Улучшенную и бесшовную переносимость: В то время как контейнеры LXC часто ссылаются на конфигурации, специфичные для конкретной машины, контейнеры Docker работают без изменений в любой операционной системе, центре обработки данных и облачной среде.
* Еще меньший вес и более точечные обновления: В LXC несколько процессов могут быть объединены в одном контейнере. В контейнерах Docker в каждом контейнере может выполняться только один процесс. Это позволяет создать приложение, которое может продолжать работать, пока одна из его частей отключена для обновления или ремонта.
* Автоматическое создание контейнеров: Docker может автоматически создавать контейнер на основе исходного кода приложения.
* Версионирование контейнера: Docker может отслеживать версии образа контейнера, откатываться к предыдущим версиям и отслеживать, кто и как создал ту или иную версию. Он может загружать только различия между существующей и новой версией.
* Повторное использование контейнеров: Существующие контейнеры можно использовать в качестве базовых образов - по сути, как шаблоны для создания новых контейнеров.
* Общие библиотеки контейнеров: Разработчики могут получить доступ к реестру с открытым исходным кодом, содержащему тысячи контейнеров, созданных пользователями.

Сегодня контейнеризация Docker также работает с сервером Microsoft Windows. А большинство облачных провайдеров предлагают специальные услуги, помогающие разработчикам создавать, доставлять и запускать приложения, контейнеризированные с помощью Docker.

По этим причинам внедрение Docker быстро набрало обороты и продолжает расти. На данный момент компания Docker Inc. сообщает о 11 миллионах разработчиков и 13 миллиардах загрузок образов контейнеров каждый месяц.

Некоторые инструменты и термины, необходимые при использовании Docker:

* DockerFile  
  Каждый контейнер Docker начинается с простого текстового файла, содержащего инструкции по созданию образа контейнера Docker. DockerFile автоматизирует процесс создания образа Docker. По сути, это список инструкций интерфейса командной строки (CLI), которые Docker Engine будет выполнять для сборки образа.
* Образы Docker  
  Образы Docker содержат исполняемый исходный код приложения, а также все инструменты, библиотеки и зависимости, необходимые для запуска кода приложения в качестве контейнера. Когда вы запускаете образ Docker, он становится одним экземпляром (или несколькими экземплярами) контейнера.  
  Можно создать образ Docker с нуля, но большинство разработчиков пользуются образами из общих репозиториев. Из одного базового образа можно создать несколько образов Docker, и они будут иметь общие черты своего прародителя.  
  Образы Docker состоят из слоев, и каждый слой соответствует версии образа. Каждый раз, когда разработчик вносит изменения в образ, создается новый верхний слой, и этот верхний слой заменяет предыдущий верхний слой в качестве текущей версии образа. Предыдущие слои сохраняются для отката или повторного использования в других проектах.  
  Каждый раз, когда из образа Docker создается контейнер, создается еще один новый слой, называемый слоем контейнера. Изменения, внесенные в контейнер, такие как добавление или удаление файлов, сохраняются только в слое контейнера и существуют только во время работы контейнера. Этот итеративный процесс создания образа позволяет повысить общую эффективность, поскольку несколько экземпляров живых контейнеров могут запускаться только из одного базового образа, и при этом они используют общий стек.
* Контейнеры Docker  
  Контейнеры Docker - это работающие экземпляры образов Docker. Если образы Docker - это файлы, доступные только для чтения, то контейнеры - это исполняемый объект. Пользователи могут взаимодействовать с ними, а администраторы могут регулировать их настройки и условия с помощью команд docker.
* Docker Hub  
  Docker Hub - это публичное хранилище образов Docker, которое называет себя "крупнейшей в мире библиотекой и сообществом для образов контейнеров". Он содержит более 100 000 образов контейнеров, полученных от коммерческих производителей программного обеспечения, проектов с открытым исходным кодом и индивидуальных разработчиков. Сюда входят образы, созданные компанией Docker, Inc, сертифицированные образы, входящие в реестр Docker Trusted Registry, и многие тысячи других образов.  
  Все пользователи Docker Hub могут делиться своими образами по своему усмотрению. Они также могут загружать предопределенные базовые образы из файловой системы Docker, чтобы использовать их в качестве отправной точки для любого проекта контейнеризации.
* Демон Docker  
  Демон Docker - это служба, работающая в вашей операционной системе, например, Microsoft Windows или Apple MacOS или iOS. Эта служба создает и управляет образами Docker, используя команды от клиента, выступая в качестве центра управления реализацией Docker.
* Docker registry  
  Docker registry - это масштабируемая система хранения и распространения образов Docker с открытым исходным кодом. Registry позволяет отслеживать версии образов в репозиториях, используя метки для идентификации. Для этого используется git, инструмент контроля версий.

Если вы используете всего несколько контейнеров, то управлять приложением довольно просто в Docker Engine - среде выполнения де-факто. Но если ваше развертывание состоит из тысяч контейнеров и сотен служб, то управлять этим рабочим процессом практически невозможно без помощи этих специализированных инструментов, таких как Docker Compose.

Приложение может состоять из нескольких контейнеров, запускающих различные службы. Запуск и управление контейнерами вручную может быть утомительным, поэтому компания Docker создала полезный инструмент, который помогает ускорить этот процесс - Docker Compose.

Docker Compose - это программное обеспечение, используемое для определения и запуска многоконтейнерных приложений Docker. Оно может одновременно работать с несколькими контейнерами в production, staging, development, testing и CI-среде. Поэтому необходимо использовать Docker Compose для управления всем жизненным циклом разработки программного обеспечения (SDLC).

Docker Compose работает путем применения правил, определенных в файле docker-compose.yaml. YAML-файл конфигурирует службы приложения и включает правила, определяющие, как вы хотите их запустить. Выполнив этот файл, появляется возможность запускать, останавливать или перестраивать все службы с помощью одной команды. Кроме того, можно проверить состояние служб, отобразить журналы и выполнить разовые команды.

Docker Compose применяется в следующих случаях:

* Среды автоматизированного тестирования:  
  Compose поддерживает автоматизированное тестирование, которое является важной частью CI/CD, поскольку с его помощью можно легко создать и уничтожить необходимую среду тестирования. Разработчики могут определить и настроить среду, необходимую для проведения автоматизированного сквозного тестирования, используя соответствующий файл Docker Compose.
* Развертывание на одном хосте:  
  В Docker Compose контейнеры предназначены для запуска на одном хосте, поскольку они традиционно ориентированы на рабочие процессы разработки и тестирования.
* Среды разработки:  
  Docker Compose - это быстрый и простой способ запуска проектов, поскольку он позволяет легко создавать новые изолированные среды разработки. Программное обеспечение документирует и настраивает все служебные зависимости приложения (включая базы данных, кэши, API веб-служб и т.д.). Оно позволяет создавать и запускать один или несколько контейнеров для каждой зависимости с помощью одной команды.

Преимуществами Docker Compose являются:

* Быстрая и простая конфигурация:  
  Благодаря скриптам YAML и переменным окружения возможно легко настраивать или изменять службы приложения.
* Безопасное внутреннее взаимодействие:  
  Compose создает сеть для совместного использования всеми службами. Это добавляет дополнительный уровень безопасности для приложения, поскольку службы не могут быть доступны извне.
* Переносимость и поддержка CI/CD:  
  Поскольку все сервисы определены в файле docker-compose, разработчики могут легко получить доступ и делиться конфигурационными файлами. Взяв YAML-файл и исходный код, они могут разворачивать приложения в считанные минуты. Это помогает настроить и запустить CI/CD систему.
* Эффективное использование ресурсов:  
  Docker Compose позволяет размещать несколько изолированных сред на одном хосте. Запуск всего на одном аппаратном сервере позволяет экономить много ресурсов. Его функции, позволяющие кэшировать конфигурацию и повторно использовать существующие контейнеры, также способствуют эффективности использования ресурсов.

К особенностям Docker Compose относятся:

* Размещение нескольких изолированных окружений на одном хосте  
  Docker Compose позволяет изолировать окружения друг от друга на одном хосте с помощью имен проектов.  
  По умолчанию имя проекта является базовым именем каталога проекта, а каталог проекта - базовым каталогом файла docker-compose. Однако есть возможность изменить значения по умолчанию:  
  - Задать имя проекта с помощью опции командной строки -p или переменной окружения COMPOSE\_PROJECT\_NAME.  
  - Задать каталог проекта с помощью переменной окружения --project-directory.  
  Можно использовать эту возможность на dev-хосте для запуска стабильных копий каждой ветки проекта, создавая несколько копий среды под разными именами.
* Поддержка переменных среды  
  Вы можете настроить контейнеры для различных сред или пользователей, добавив переменные окружения в файл docker-compose. Это дает большую гибкость при настройке контейнеров с помощью Compose, поскольку значения переменных не закодированы в конфигурации.  
  Значения переменных могут быть заданы в среде оболочки (из которой вы запускаете docker-compose) или в файле .env (хранится в каталоге проекта). По умолчанию Docker compose применяет значения, указанные в файле .env. Однако значения, установленные в среде оболочки, переопределяют значения из файла .env.
* Сохранение данных хранилища  
  Еще одной особенностью Docker Compose является сохранение данных, используемых сервисами. Поэтому вам не нужно беспокоиться о потере данных, созданных в контейнерах.
* Повторное использование существующих контейнеров  
  Docker Compose воссоздает только те контейнеры, которые изменились с момента последнего запуска. Если изменений нет, он повторно использует существующий контейнер.  
  Эта функция опирается на способность программного обеспечения кэшировать конфигурации контейнеров, что позволяет быстрее настраивать службы.

Многие атаки на устройства под управлением Windows можно выявить с помощью Windows event logs. Ниже представлены критически важные для информационной безопасности

Windows event лог под id 4688

Событие 4688 документирует каждую программу (или процесс), выполняемую системой, вместе с процессом, который запустил программу. Этот идентификатор события интересен тем, что он регистрирует любой процесс, созданный пользователем или даже порожденный скрытым процессом. Например, если в системе Windows присутствует вредоносное ПО, поиск по событию 4688 выявит все процессы, запущенные этой злонамеренной программой. Красные флажки вредоносной активности включают дочерние процессы, ID родительского процесса которых отличается от ID исходного процесса, и процессы, выполняющиеся в другом месте вместо C:Program Files или C:windowssystem32.

Кроме того, есть возможность получить информацию об административных привилегиях пользователя через поле Token Elevation Type. Токен типа 1 означает "полный токен" со всеми привилегиями, предоставленными данной учетной записи пользователя, например, когда UAC (User Access Control) отключен или когда пользователь находится в служебной или встроенной учетной записи администратора. Тип 2 означает то, что повышенный токен был выдан через опцию "Запуск от имени администратора" при включенном UAC. Тип 3 - это ограниченный токен без административных групп или привилегий. Он выдается, когда пользователь не запускает программу с помощью Run as administrative или когда приложение не требует административных привилегий.

Хотя событие 4688 может рассказать о многом, его следует использовать в сочетании с другими журналами событий, чтобы получить полную картину вторжения.

Windows event лог под id 4670

Одним из лучших способов выявления несанкционированного доступа (и, в конечном итоге, утечки данных) является отслеживание изменений разрешений файлового сервера. Именно здесь пригодится событие 4670 - оно срабатывает, когда пользователь изменяет список управления доступом объекта. Известно, что хакеры меняют разрешения, когда пытаются двигаться горизонтально или внедрять в систему программы-выкупы; отслеживание того, кто принимает на себя ответственность за вторжение, является важным шагом в отслеживании источника атаки.

Помимо обнаружения вторжений, вы также можете использовать событие 4670 для получения информации об активности пользователей. С его помощью можно получить информацию о пиковом времени входа в систему, посещаемости пользователей и многом другом.

Windows event лог под id 4672

Это событие информирует специалиста ИБ всякий раз, когда в систему входит эквивалентная учетная запись администратора. Вы можете отследить его для поиска потенциальной атаки Pass-the-Hash (PtH). Если "SubjectSecurity ID" в средстве просмотра событий не содержит "LocalSystem, NetworkService, LocalService", это не эквивалентная администратору учетная запись и требует тщательного анализа.

Но событие 4672 - не единственный идентификатор журнала событий безопасности Windows, указывающий на атаку типа "pass-the-hash". Многие другие события, включая 4648 (была попытка входа в систему с явными учетными данными), 4624 (учетная запись была успешно зарегистрирована) и 4776 (компьютер попытался подтвердить учетные данные учетной записи), могут указывать на коллективный взлом системы.

Обнаружить горизонтальное движение от атак Pass-the-Hash непросто, но SIEM, позволяющая создавать правила корреляции вокруг этого движения, может помочь вам определить другие события, связанные с PtH.

Windows event лог под ID 1125 (Ошибка)

Windows обычно управляет своими параметрами конфигурации на серверах и рабочих станциях с помощью "групповой политики Active Directory". Мониторинг события 1125 поможет вам определить потенциальные сбои, связанные с применением политики или несанкционированными изменениями объектов политики в Active Directory, а не с ошибкой пользователя. Если политика не может быть применена, есть вероятность, что в системе есть проблема безопасности.

Помимо политики AD, также полезно следить за правилами брандмауэра. Поскольку брандмауэр Windows Firewall обеспечивает критически важную линию защиты, злоумышленник может попытаться изменить его правила, чтобы получить доступ к вашей системе. Используйте функцию ведения журнала брандмауэра для проверки динамических и отключенных открытий портов, а также для анализа отброшенных пакетов на маршруте отправки.

В случае обнаружения вредоносной активности, откройте файл журнала в блокноте и используйте DROP для фильтрации всех записей в поле action, анализируя, какие IP-адреса назначения заканчиваются числом, отличным от 255. Если вы обнаружите много таких IP-адресов, запишите IP-адреса назначения пакетов для устранения неполадок.

Windows event логи под ID 1024 и 1102

Следующие два события мы рассмотрим в журнале событий TerminalServices-RDPClient\Operational на исходной машине. В средстве просмотра событий вы можете найти этот журнал в разделе Журналы приложений и служб \ Microsoft \ Windows \ TerminalServices-ClientActiveXCore.

Эти события имеют идентификаторы 1024 и 1102, и каждое из них содержит определенную, потенциально полезную информацию.

Во-первых, событие 1024 обычно появляется в журнале за пару секунд до события 4648, описанного выше. Оно показывает нам, что клиент RDP пытается подключиться к удаленной машине или серверу. В нем будет указано имя машины, если оно доступно.

Второе, 1102, обычно появляется примерно через 10 секунд (плюс-минус несколько) после события 1024. Оно предоставит нам IP-адрес удаленной машины после того, как она инициирует соединение с местом назначения.

Windows event логи под ID 1006 и 1007

Рассмотрите возможность инвестирования уведомлений для выявления, предотвращения и удаления вредоносного ПО в Windows Defender. Да, даже встроенный антивирус может быть использован для осуществления вредоносной деятельности. Начните с просмотра события ID 1006, которое запускается, когда Защитник обнаруживает нежелательное программное обеспечение. Затем просмотрите событие 1007, чтобы узнать, действовал ли антивирус для защиты вашей системы от потенциального проникновения. Все эти события присутствуют в подлоге.

Windows event логи под ID 4624 и 4625

Event ID 4624 (просматривается в Windows Event Viewer) фиксирует каждую успешную попытку входа в систему на локальном компьютере. Это событие генерируется на компьютере, к которому был получен доступ, другими словами, там, где была создана сессия входа. Связанное событие, Event ID 4625, документирует неудачные попытки входа в систему.

Windows event лог под ID 1149

Событие 1149 регистрируется при успешном входе в систему RDP на компьютере. До Windows 7 и Windows Server 2012 событие 1149 регистрировалось при любой инициации RDP-соединения, поэтому оно не было полезным индикатором успешного применения учетных данных пользователя.

Но ситуация изменилась, и все современные версии ОС будут регистрировать 1149 только в том случае, если имя пользователя в событии было успешно аутентифицировано. Он также включает IP-адрес источника соединения, что является полезной информацией.

Если используется учетная запись, которая успешно прошла аутентификацию, но не имеет разрешения на RDP к компьютеру из-за других ограничений, событие 1149 не генерируется.

Наше первое событие, ID 21, регистрируется, когда RDP успешно входит в сеанс.

В этом событии регистрируется как имя пользователя подключения, так и присвоенный идентификационный номер сессии. Имя пользователя здесь включает домен и является учетной записью, используемой для входа в систему, не обязательно учетной записью, зарегистрированной на исходной машине.

Событие 22

Следующее событие, на которое следует обратить внимание, - 22. Оно содержит ту же информацию, что и 21, так что ничего особо нового здесь нет. Но оно регистрирует получение уведомления о запуске Shell, что может быть дополнительным свидетельством интерактивного сеанса входа в систему.

Событие 25

Последним из событий входа в систему или подключения, которые мы рассмотрим сегодня, является событие 25. Это событие регистрируется, когда RDP повторно подключается к сеансу, как в случае с сеансом входа типа 7, о котором мы говорили выше. Пользователь уже вошел в сеанс, а затем RDP повторно подключился к нему. Это может быть локальная сессия или предыдущая сессия RDP.

В любом случае, мы бы увидели 4624, созданный при входе в систему типа 7. Это событие также включает IP-адрес источника соединения, который может быть ценным, если он не был доступен в предыдущих событиях.

Теперь, как всегда, нам необходимо рассмотреть полную хронологию действий, которые злоумышленник совершил на исследуемых системах. Поэтому мы должны искать не только события входа в систему и подключения, но и события выхода из системы.

Есть, конечно, два события, которые появляются в журнале безопасности, 4634 и 4647. Они регистрируют события, когда пользователь инициирует выход из системы (4647) и когда пользователь фактически вышел из системы (4634). Вы можете получить некоторые дополнительные сведения о том, была ли система перезагружена, сравнив идентификаторы входа.

По большей части, это то, что они вам скажут, и они не указывают, была ли сессия локальной или удаленной. Но если ваши журналы безопасности не повреждены, они могут быть полезны для пополнения нашего представления о действиях злоумышленника.

Однако, как я уже неоднократно упоминал, журнал безопасности является главной мишенью для манипуляций с журналами и их уничтожения, поэтому нам понадобятся другие места для сбора этой информации. Для этого мы вернемся к журналу TerminalServices-LocalSessionManager/Operational и двум событиям, которые я упоминал ранее: 23 и 40.

Событие 40

Во-первых, давайте рассмотрим событие 40. Это событие регистрируется всякий раз, когда сеанс отключается, то есть это может быть прерывание, отключение или выход пользователя из системы. В тексте события указывается код причины, который дает нам подробную информацию об отключении.

Чаще всего встречаются типы 0, 5, 11 и 12, но существуют определения для всех кодов от 0 до 12.

Код 0 означает, что просто отсутствует дополнительная информация об отключении.

Код 2 аналогичен коду 11; он регистрируется, когда для отключения сеанса от другого сеанса использовался административный инструмент.

Код 5 регистрируется, когда пользователь подключается к машине, вынуждая отключить другое текущее соединение. Это может быть одно и то же используемое имя пользователя или система просто не поддерживает несколько одновременных сеансов.

Код 11 регистрируется, когда отключение было инициировано пользователем, который отключается от сеанса. Это может быть закрытие пользователем окна RDP или использование административного инструмента из той же сессии для принудительного отключения, например, команды logoff в CMD или пакетного файла.

Код 12 регистрируется, если отключение было инициировано выходом пользователя из сеанса на машине, например, выход из системы через меню "Пуск".

Событие 23

Наконец, у нас есть событие 23, которое регистрируется при успешном выходе из сеанса.

Это событие регистрируется после того, как все сценарии выхода из системы или другие задачи, запланированные для завершения при выходе из системы, показывают, что сеанс действительно завершился. Здесь также регистрируется пользователь, который был подключен, и идентификатор сессии, который был с ним связан.

**Тут еще можно налить воды про докер и докер-композ, а также Linux, Windows, CentOS. Рассказать про его +/-. Рассказать про события в винде, какие бывают и что делать с ними. Рассказать про атаки**

# Глава 2 - Синтез

Описать страдания с конфигами, с настройкой битов. Предоставить полную схему инфраструктуры. Рассказать про то, как проводятся атаки. Рассказать как я писал правила, анализ логов.

Одной из задач данной работы было создание рабочей конфигурации Elasticsearch и Kibana. Для выполнения этой задачи необходимо было создать docker-compose файл, позволяющий с помощью одной команды развернуть SIEM-систему на любой инфраструктуре, а также конфигурационные файлы для Elasticsearch и Kibanа.

Ниже представлена структура Docker-compose файла:

* services
  + service name
    - service’s parameters
      * subparameters for parameters

Пример конфигурации одного из сервисов:

services:

kibana:

build: ./kibana/

container\_name: kibana

volumes:

- type: bind

source: ./kibana/config/kibana.yml

target: /usr/share/kibana/config/kibana.yml

read\_only: true

ports:

- "5601:5601"

networks:

- elk

depends\_on:

- elasticsearch

Где параметр build отвечает за местоположение DockerFile для создания контейнера; container\_name - название контейнера для обращения к нему; volumes - сопоставляет папку/файл из родительской системы с папкой/файлом в контейнере, подпараметр type описывает каким образом происходит сопоставление, подпараметр read\_only показывает разрешения на операции с файлами; ports - определяет связку портов контейнера и хостовой системы; networks - указывает внутреннюю сеть между контейнерами; depends\_on - необходим для создания связки между контейнерами. С полным docker-compose файлом можно ознакомиться в Приложении №.

Перейдем к рассмотрению конфигурационных файлов для Elasticsearch и Kibana.

Elasticsearch:

discovery.type: single-node

node.name: "el1"

network.host: 0.0.0.0

xpack.license.self\_generated.type: basic

xpack.security.enabled: true

xpack.security.authc.api\_key.enabled: true

xpack.monitoring.collection.enabled: true

Где discovery.type - выбор количество нод; node.name - название ноды для обращения; network.host - устанавливает адрес этого узла для HTTP и транспортного трафика, при установки значения 0.0.0.0 хост будет доступен из любой сети в интернете; xpack.\* - базовые настройки безопасности, без которых невозможна будет работа ELK Stack.

Kibana:

server.name: kibana

server.host: "0.0.0.0"

elasticsearch.hosts: [ "http://elasticsearch:9200" ]

xpack.monitoring.ui.container.elasticsearch.enabled: true

elasticsearch.username: elastic

elasticsearch.password: 4FLSrCcxthZnsHRYtlnQ

xpack.encryptedSavedObjects.encryptionKey: "akjfldjrntbfhcilel12344nfjslivndlsie"

Где server.name - человекочитаемое отображаемое имя, идентифицирующее данный экземпляр Kibana; server.host - параметр, который разрешает удаленное подключение пользователей; elasticsearch.hosts - URL-адреса Elasticsearch, к которым будет обращаться Kibana для запроса данных; xpack.\* - базовые настройки безопасности, без которых невозможна будет работа ELK Stack; elasticsearch.username и elasticsearch.password - необходимы для получения доступа к данным в Elasticsearch.

После создания вышеописанных файлов, появляется возможность запускать ELK Stack на любой системе, с установленными приложениями Docker и Docker Compose. В данной работе в качестве сервера использовалась виртуальная машина под управлением ОС Centos 7. Эта операционная система популярна в качестве серверной ОС в больших компаниях. Это связано с некоторыми важными факторами:

* Длительность поддержки LTS версий (10 лет)
* Совместимость с пакетами под Red Hat enterprise linux (RHEL), большое кол-во софта для корпораций пишется с заточкой под него
* Стабильность из-за редких обновлений (раз в полгода)

Настройка базового образа Centos 7 в качестве сервера для ELK Stack представлена в приложении № (лаба).

После установки серверной части ELK Stack перейдем к настройки клиентских устройств и приложений.

Представим, что в инфраструктуре компании присутствует некоторое количество вычислительных серверов также под управлением ОС Centos 7. Для мониторинга данной системы была выбрана утилита FileBeat, которая отслеживает изменения в файлах(логах). Утилита была настроена для работы с ELK Stack с помощью следующего конфигурационного файла - /etc/filebeat/filebeat.yml. Ниже представлены выдержки из файла, подвергшиеся изменению. Полный файл находится в приложении №.

paths:

- /var/log/\*.log

setup.ilm.enabaled: true

setup.ilm.overwrite: true

hosts: ["192.168.128.50:9200"]

username: "elastic"

password: "4FLSrCcxthZnsHRYtlnQ"

Где paths - пути к каталогам, которые будут проверяться на изменения; setup.ilm.enabaled - включает или выключает управление “жизненным циклом” индекса для любых новых индексов, созданных Filebeat; setup.ilm.overwrite - если установлено значение “true”, политика “жизненного цикла” индекса перезаписывается при запуске; hosts - список нод Elasticsearch, к которым необходимо подключиться; username и password - необходимы для получения доступа к данным в Elasticsearch.

На клиентском ПК с ОС Windows установлены Winlogbeat, Packetbeat. Winlogbeat - собирает события, записанные системой Windows, Packetbeat - отправляет данные о сетевой активности устройства. Также созданы конфигурационные файлы для этих программ.

Ниже представлен конфигурационный файл для Packetbeat:

setup.kibana:

host: "192.168.128.50:5601"

setup.kibana.ssl.enabled: false

output.elasticsearch:

# Array of hosts to connect to.

hosts: ["192.168.128.50:9200"]

username: "elastic"

password: "4FLSrCcxthZnsHRYtlnQ"

Где в пункте setup.kibana - указан IP-адрес Kibana, и флаг, указывающий на отсутствие шифрования; в пункте output.elasticsearch подпункты означают: hosts - список нод Elasticsearch, к которым необходимо подключиться; username и password - используются для получения доступа к данным в Elasticsearch.

Далее показаны изменения в файле конфигурации для Winlogbeat:

winlogbeat.event\_logs:

- name:Microsoft-Windows-Authentication User Interface/Operational

- name: Microsoft-Windows-WMI-Activity/Operational

- name: Microsoft-Windows-TerminalServices-RemoteConnectionManager/Operational

setup.kibana:

host: "192.168.128.50:5601"

setup.kibana.ssl.enabled: false

output.elasticsearch:

# Array of hosts to connect to.

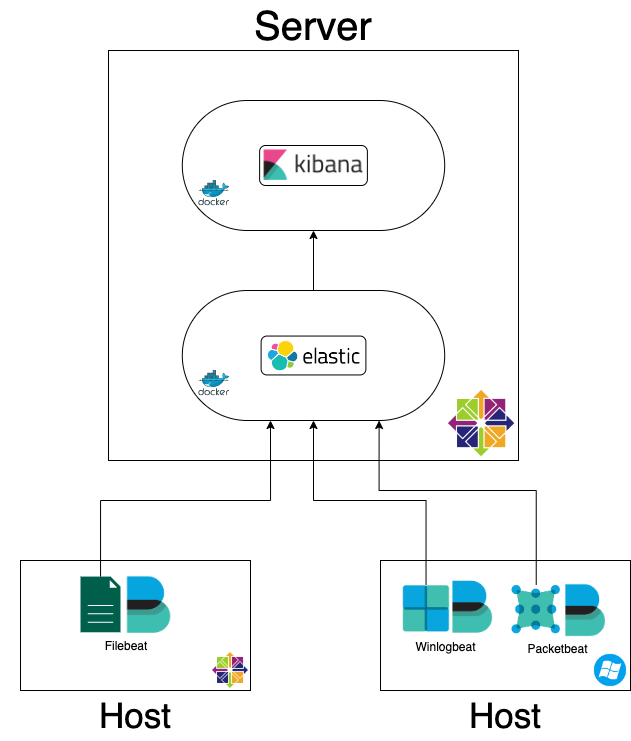
hosts: ["192.168.128.50:9200"]

username: "elastic"

password: "4FLSrCcxthZnsHRYtlnQ"

Где в пункте winlogbeat.event\_logs добавлены логи, которые теперь будут также отслеживаться; в пункте setup.kibana - указан IP-адрес Kibana, и флаг, указывающий на отсутствие шифрования; в пункте output.elasticsearch подпункты означают: hosts - список нод Elasticsearch, к которым необходимо подключиться; username и password - используются для получения доступа к данным в Elasticsearch.

На рисунке 1 представлена полная схема стенда для проведения атак, а также их обнаружения.

Рис. 1. Схема инфраструктуры

Для демонстрации возможностей данной системы было выбрано 2 популярных типа атак: bruteforce - грубый подбор пары логин - пароль, например, для доступа по протоколу Remote Desktop Protocol (RDP)[2] или Windows Remote Management (WinRM)[3], и атаку использующую уязвимость Eternal Blue[4].

Определение атаки методом перебора

Атака перебором - это метод взлома, использующий метод проб и ошибок для взлома паролей, учетных данных и ключей шифрования. Это простая, но надежная тактика получения несанкционированного доступа к индивидуальным учетным записям и системам и сетям организаций. Хакер пробует несколько имен пользователей и паролей, часто используя компьютер для проверки широкого спектра комбинаций, пока не найдет правильную информацию для входа.

Название " brute force" происходит от того, что злоумышленники используют чрезмерно интенсивные попытки получить доступ к учетным записям пользователей. Несмотря на то, что это старый метод кибератак, атаки грубой силы проверены и остаются популярной тактикой среди хакеров.

Типы атак методом перебора

Существуют различные виды атак методом перебора, которые позволяют злоумышленникам получить несанкционированный доступ и украсть данные пользователя.

Простые атаки методом перебора

Простая атака грубой силы происходит, когда хакер пытается угадать учетные данные пользователя вручную, без использования программного обеспечения. Обычно для этого используются стандартные комбинации паролей или коды персонального идентификационного номера (PIN).

Эти атаки просты, потому что многие люди до сих пор используют слабые пароли, такие как "password123" или "1234", или практикуют плохой парольный этикет, например, используют один и тот же пароль для нескольких веб-сайтов. Пароли также могут быть угаданы хакерами, которые проводят минимальную исследовательскую работу, чтобы взломать потенциальный пароль человека, например, название его любимой спортивной команды.

Атаки по словарю

Атака по словарю - это базовая форма взлома методом перебора, при котором атакующий выбирает цель, а затем проверяет возможные пароли на соответствие имени пользователя этого человека. Сам метод атаки технически не считается перебором, но он может играть важную роль в процессе взлома пароля злоумышленником.

Название "атака по словарю" происходит от того, что хакеры просматривают словари и дополняют слова специальными символами и цифрами. Этот тип атаки обычно занимает много времени и имеет низкие шансы на успех по сравнению с новыми, более эффективными методами атаки.

Гибридные атаки методом грубой силы

Гибридная атака методом грубой силы - это когда хакер сочетает метод атаки по словарю с простой атакой методом грубой силы. Она начинается с того, что хакер узнает имя пользователя, затем проводит атаку по словарю и простой перебор, чтобы узнать комбинацию входа в учетную запись.

Злоумышленник начинает со списка потенциальных слов, а затем экспериментирует с комбинациями символов, букв и цифр, чтобы найти правильный пароль. Такой подход позволяет злоумышленникам обнаружить пароли, сочетающие распространенные или популярные слова с цифрами, годами или случайными символами, например, "SanDiego123" или "Rover2020".

Атаки обратного перебора

При атаке обратным перебором злоумышленник начинает процесс с известного пароля, который обычно обнаруживается в результате взлома сети. С помощью этого пароля они ищут подходящие учетные данные для входа в систему, используя списки из миллионов имен пользователей. Злоумышленники также могут использовать часто используемый слабый пароль, например "Password123", для поиска совпадения в базе данных имен пользователей.

Вброс учетных данных

Использование слабых паролей - это способ получения доступа к учетным данным. Злоумышленники собирают украденные ими комбинации имен пользователей и паролей, которые затем проверяют на других сайтах, чтобы узнать, смогут ли они получить доступ к дополнительным учетным записям пользователей. Этот подход приносит результат, если люди используют одну и ту же комбинацию имени пользователя и пароля или повторно используют пароли для различных учетных записей и профилей социальных сетей.

Что стоит за атаками методом перебора?

Взлом методом грубой силы требует большого терпения, поскольку для успешного взлома пароля или ключа шифрования злоумышленнику могут потребоваться месяцы или даже годы. Однако потенциальное вознаграждение огромно.

Эксплуатация рекламы или данных о действиях

Хакер может провести атаку грубой силы на веб-сайт или несколько веб-сайтов, чтобы получить финансовую прибыль от рекламных комиссионных. К распространенным методам относятся:

Размещение спам-объявлений на популярных сайтах, что позволяет злоумышленнику зарабатывать деньги каждый раз, когда на объявление нажимает или просматривает посетитель.

Перенаправление трафика с законного сайта на незаконные рекламные сайты.

Заражение веб-сайта и посетителей сайта вредоносными программами, например, шпионскими, которые отслеживают их активность. Собранные данные затем продаются рекламодателям без согласия пользователя.

Кража личных данных

Взлом личных аккаунтов пользователей может стать сокровищницей данных - от финансовых сведений и банковских счетов до конфиденциальной медицинской информации. Доступ к аккаунту позволяет злоумышленнику подделать личность человека, украсть его деньги, продать учетные данные третьим лицам или использовать информацию для более масштабных атак.

Личные данные и учетные данные также могут быть украдены в результате утечки корпоративных данных, когда злоумышленники получают доступ к конфиденциальным базам данных организаций.

Распространение вредоносного ПО

Атаки методом грубой силы часто не носят личного характера. Хакер может просто хотеть посеять хаос и продемонстрировать свои способности к вредоносным действиям. Для этого они могут распространять вредоносное ПО по электронной почте или через SMS-сообщения, скрывать его на поддельных сайтах, выглядящих как законные, или перенаправлять посетителей сайтов на вредоносные сайты.

Заразив компьютер пользователя вредоносным ПО, злоумышленник может проникнуть в подключенные системы и сети и начать более масштабные кибератаки на организации.

Захват систем для вредоносной деятельности

Атаки методом перебора могут сыграть свою роль в организации злоумышленниками более масштабных атак с использованием множества устройств, называемых ботнетами. Как правило, это распределенная атака типа "отказ в обслуживании" (DDoS), цель которой - преодолеть защиту и системы безопасности объекта атаки.

Подрыв репутации компании или веб-сайта

Атаки методом перебора часто осуществляются с целью кражи данных у организации, что не только приводит к финансовым потерям, но и наносит огромный репутационный ущерб. Веб-сайты также могут подвергаться атакам, заражающим их непристойными или оскорбительными текстами и изображениями, тем самым порочащими их репутацию, что может привести к их закрытию.

Инструменты атаки методом перебора

Подбор пароля к электронной почте или сайту социальной сети может занять много времени, особенно если у аккаунта сложный пароль. Чтобы упростить этот процесс, хакеры разработали программное обеспечение и инструменты для взлома паролей.

К инструментам атаки грубой силы относятся приложения для взлома паролей, которые взламывают комбинации имен пользователей и паролей, которые человеку было бы крайне сложно взломать самостоятельно. К наиболее распространенным инструментам для перебора относятся:

Aircrack-ng: Набор инструментов для оценки безопасности сети Wi-Fi с целью мониторинга и экспорта данных и атаки на организацию с помощью таких методов, как поддельные точки доступа и инъекция пакетов.

John the Ripper: Инструмент для восстановления паролей с открытым исходным кодом, который поддерживает сотни типов шифров и хэшей, включая пароли пользователей macOS, Unix и Windows, серверов баз данных, веб-приложений, сетевого трафика, зашифрованных закрытых ключей и файлов документов.

L0phtCrack - инструмент для взлома паролей Windows. В нем используются радужные таблицы, словари и многопроцессорные алгоритмы.

Hashcat - работает в Windows, Linux и Mac OS. Может выполнять простой перебор, атаки на основе правил и гибридные атаки.

DaveGrohl - инструмент с открытым исходным кодом для взлома Mac OS. Может быть распределен по нескольким компьютерам.

Ncrack - инструмент для взлома сетевой аутентификации. Его можно использовать в Windows, Linux и BSD.

Эти типы программ могут быстро угадывать комбинации, определяющие слабые пароли, и взламывать многочисленные компьютерные протоколы, беспроводные модемы и зашифрованные устройства хранения данных.

Атака методом грубой силы может потребовать огромных вычислительных мощностей. Для решения этой проблемы хакеры разработали аппаратные решения, которые упрощают процесс, например, объединение центрального процессора (CPU) и графического процессора (GPU) устройства. Добавление вычислительного ядра GPU позволяет системе обрабатывать несколько задач одновременно, а хакерам - значительно быстрее взламывать пароли.

Что такое Eternalblue?

CVE-2017-0143 - CVE-2017-0148 - это семейство критических уязвимостей в сервере Microsoft SMBv1, используемом в Windows 7, Windows Server 2008, Windows XP и даже Windows 10, работающем через порт 445. Закодированные строки в оригинальном исполняемом файле Eternalblue раскрывают целевые версии Windows.

Однако уязвимость относится не только к Microsoft Windows; фактически, все, что использует протокол сервера Microsoft SMBv1, например, ультразвуковое медицинское оборудование Siemens, потенциально уязвимо.

Сам Eternalblue касается CVE-2017-0144, недостатка, который позволяет удаленным злоумышленникам выполнить произвольный код на целевой системе, отправив специально созданные сообщения на сервер SMBv1. Другие связанные эксплойты были обозначены как Eternalchampion, Eternalromance и Eternalsynergy группой Equation Group - прозвище хакерской группы APT, которая, как теперь предполагается, является Агентством национальной безопасности США.

Насколько широко распространен Eternalblue?

Недостатки в протоколе SMBv1 были исправлены компанией Microsoft в марте 2017 года с помощью обновления безопасности MS17-010. К сожалению, несмотря на то, что патч был доступен уже более 5 лет, по имеющимся данным, около миллиона машин, подключенных к Интернету, все еще остаются уязвимыми.

Во всем мире наиболее нуждающимися в обновлении версиями Windows являются Windows Server 2008 и 2012 R2.

Из более чем 400 000 машин, уязвимых к Eternalblue, расположенных в США, более четверти из них, около 100 000, находятся в Калифорнии, сердце американской технологической индустрии.

И не только программы-вымогатели пользуются широким распространением Eternalblue. Эксплойт может найти применение в любом вредоносном ПО, требующем червеобразных возможностей. С января 2019 года криптоджекеры атакуют предприятия в Китае с помощью Eternalblue и вредоносной программы Beapy.

Кроме того, сейчас, когда программы-вымогатели снова входят в моду после короткого перерыва в 2018 году, Eternalblue снова попадает в заголовки газет и в США, хотя в некоторых случаях авторство кажется неуместным. Вопреки некоторым сообщениям, RobinHood Ransomware, которая нанесла ущерб Baltimore, не обладает способностью распространяться и, скорее всего, попадает на каждую машину в отдельности.

Как используется Eternalblue?

Как уже упоминалось выше, эксплуатация CVE-2017-0144 с помощью Eternalblue - это метод, предположительно разработанный АНБ и ставший известным всему миру после утечки в интернет их инструментария.

Чтобы воспользоваться уязвимостью, неаутентифицированному злоумышленнику достаточно отправить на сервер вредоносно составленный пакет, именно так и распространялись программы WannaCry и NotPetya ransomware. По сути, Eternalblue позволил программе-вымогателю получить доступ к другим машинам в сети. Злоумышленники могут использовать DoublePulsar, также разработанный Equation Group и ставший известным благодаря Shadow Brokers, в качестве полезной нагрузки для установки и запуска копии ransomware на любой уязвимой цели.

Как работает Eternalblue?

Eternalblue полагается на функцию Windows под названием srv!SrvOS2FeaListSizeToNt. Чтобы понять, как это приводит к удаленному выполнению кода, давайте вкратце рассмотрим, как работает SMB.

В первую очередь, SMB (Server Message Block) - это протокол, используемый для запроса служб файлов и печати от серверных систем по сети. Среди спецификаций протокола есть структуры, которые позволяют протоколу передавать информацию о расширенных атрибутах файла, по сути, метаданные о свойствах файла в файловой системе.

Eternalblue использует преимущества трех различных ошибок. Первая - это математическая ошибка, когда протокол пытается привести структуру списка OS/2 FileExtended Attribute (FEA) к структуре NT FEA, чтобы определить, сколько памяти нужно выделить. Неверный расчет приводит к целочисленному переполнению, в результате которого выделяется меньше памяти, чем ожидалось, что в свою очередь приводит к переполнению буфера. Когда записывается больше данных, чем ожидалось, лишние данные могут переполниться в соседнее пространство памяти.

Переполнение буфера происходит благодаря второй ошибке, которая возникает из-за разницы в определении протокола SMB двух связанных подкоманд: SMB\_COM\_TRANSACTION2 и SMB\_COM\_NT\_TRANSACT. Обе команды имеют команду \_SECONDARY, которая используется, когда слишком много данных для включения в один пакет. Решающее различие между TRANSACTION2 и NT\_TRANSACT заключается в том, что последняя вызывает пакет данных вдвое большего размера, чем первая. Это важно, поскольку ошибка в проверке возникает, если клиент отправляет подделанное сообщение с помощью подкоманды NT\_TRANSACT непосредственно перед командой TRANSACTION2. Хотя протокол распознает, что были получены две отдельные подкоманды, он назначает тип и размер обоих пакетов (и соответствующим образом выделяет память), основываясь только на типе последнего полученного пакета. Поскольку последний пакет меньше, первый пакет займет больше места, чем ему выделено.

Как только злоумышленники достигнут этого начального переполнения, они могут воспользоваться третьей ошибкой в SMBv1, которая позволяет распыление кучи (heap spraying) - технику, которая приводит к выделению куска памяти по заданному адресу. Отсюда злоумышленник может написать и выполнить шеллкод, чтобы взять систему под контроль.

Тестировщики на проникновение и "красные команды" вскоре увидели ценность использования этих связанных эксплойтов, и вскоре они были усовершенствованы и включены во фреймворк Metasploit.

Довольно простой Ruby-скрипт, написанный Шоном Диллоном и доступный из Metasploit, может как просканировать цель на предмет отсутствия исправлений, так и использовать все соответствующие уязвимости.

Атаки проводятся с помощью специализированной ОС для тестирований на проникновение - Kali Linux, с использованием скриптов на языке программировании Python, и таких утилит, как Crowbar и Metasploit Framework[5]. Все атаки будут происходить на ПК под управлением Windows 7.

Любые действия, направленные на взлом инфраструктуры, оставляют следы. Благодаря этому, у сотрудников отдела информационной безопасности появляется возможность обнаруживать атаки и пресекать их до утечки конфиденциальной информации.

Первостепенно будем пытаться подобрать пароль к учетной записи пользователя для подключения по RDP. Отследить такие действия не составляет труда, это связано с тем, что при подборе пароля появляется множество событий с Windows event ID 4625, которые в операционной системе означают ложную авторизацию.

Таким образом для выявления успешной атаки подбора пароля для RDP мы будем отслеживать следующую цепочку событий:

1. Большое количество событий с event ID 4625,
2. Одно событие с event ID 4624, которое означает успешный вход в систему
3. Событие с event ID 1149 - подключение по протоколу RDP.

Добавляя условие множественной ложной авторизации удается минимизировать количество ложных срабатываний, например, когда пользователь случайно ввел неверный пароль несколько раз. Данное правило будет выявлять критическую ситуацию, так как если оно сработало, то значит через злоумышленники уже обошли защиту. По этой причине предлагается дополнительное правило, которое будет отслеживать подбор пароля, путем выявления большого количества логов с event ID 4625. Благодаря такому подходу у сотрудников SOC будет время предпринять меры для усиленной защиты, а если атакующие окажутся быстрее, применить крайние меры по защите.

При попытке подбора пароля для входа по WinRM в событии event ID 4625 указывается имя процесса - svchost.exe. Таким образом мы можем создать фильтр, в котором ищем события с event ID 4625, но при этом чтобы в записи присутствовал процесс svchost.exe. Далее задаем планку срабатывания правила, например, если неудачных попыток входа было больше 100 шт. Данное правило позволяет определить, что происходит подбор пароля. Для выявления успешного подбора пароля необходимо еще проверять не появлялось ли событие 4624.

Эксплуатацию уязвимости Eternal Blue обнаружить сложнее. Проблема заключается в том, что после атаки с использованием данной уязвимости остается только одно событие в журналах Windows - это событие с event ID 4625 (неуспешная попытка авторизации). Благодаря Winlogbeat данное событие передается на сервер. Однако этого не достаточно для точного выявления данного вектора атаки. Неверный ввод пароля пользователем будет восприниматься правилом, как атака. Как говорилось ранее, данный эксплойт использует уязвимость в сетевом протоколе SMB. По этой причине с помощью утилиты Packetbeat будет происходить мониторинг подключений по 445 порту, который является портом SMB протокола. Таким образом собирая события из разных источников удается выявить атаку.

Ссылки

<https://www.ibm.com/topics/siem>

[**https://www.esecurityplanet.com/endpoint/antivirus-vs-epp-vs-edr/**](https://www.esecurityplanet.com/endpoint/antivirus-vs-epp-vs-edr/)

[**https://www.cynet.com/xdr-security/understanding-xdr-security-concepts-features-and-use-cases/**](https://www.cynet.com/xdr-security/understanding-xdr-security-concepts-features-and-use-cases/)

[**https://encyclopedia.kaspersky.com/glossary/next-generation-firewall-ngfw/**](https://encyclopedia.kaspersky.com/glossary/next-generation-firewall-ngfw/)

**https://www.lightnetics.com/topic/16190/functionbeat-serverless-ingestion-for-elasticsearch**