Intro

MITM-атаки пользуются спросом среди пентестеров. Однако сами MITM-атаки не прощают ошибок, и я встречал многих, которые во время проведения данной атаки наносили урон инфраструктуре, что в свою очередь нарушало бизнес-процессы. В этой статье я поделюсь небольшим бэкграундом, который поможет избежать грубых ошибок во время эксплуатации MITM.

Disclaimer (Дисклеймер)

Статья предназначена исключительно для специалистов по безопасности, проводящих тестирование на проникновение в рамках согласованного контракта. Взлом и разрушение чужих сетей преследуются по закону, поэтому не испытывай судьбу и используй эти знания, чтобы повысить уровень своей компетенции и помочь коллегам по цеху повысить уровень защиты сетевой инфраструктуры.

Ключевые элементы во время MITM-атаки

Firewall Checking

Перед проведением атаки убедись, что на FW твоей системы порядок: нет таких правил, которые бы как-то ограничивали трафик легитимных хостов.

```
Inguz@Inguz:~$ sudo iptables -L
Inguz@Inguz:~$ sudo iptables -t nat -L
Inguz@Inguz:~$ sudo iptables -t raw -L
Inguz@Inguz:~$ sudo iptables -t mangle -L
```

Forwarding Mode

Это режим ОС, в котором компьютер выступает в качестве маршрутизирующего устройства. Во время проведения атаки, ты становишься буквально "человеком по середине" и будет фейл, если на твоей стороне не будет активна маршрутизация, иначе возникнет нарушение работы сети на компьютерах в сети. По умолчанию режим маршрутизации в Linux отключен. Включается в одну команду.

```
Inguz@Inguz:~$ sudo sysctl -w net.ipv4.ip_forward=1
```

Promiscious mode

Это специальный режим сетевого интерфейса, в котором сам интерфейс принимает все сетевые пакеты независимо от того, кому они были адресованы. Обычно этим пользуются, если нужно проанализировать трафик в каком-нибудь сегменте. Promiscious mode позволяет захватить немного больше трафика, что в свою очередь дарит больше импакта атакующему, ему же выгодно знать всё, что происходит в трафике (насколько это возможно)

```
Inguz@Inguz:~$ sudo ip link set ethX promisc on
```

Производительность оборудования

Во время проведения MITM-атаки, трафик других устройств в сети пойдет через твой компьютер. Ты должен быть к этому готов. Поэтому убедись, что у тебя достаточно производительный процессор, также это касается оперативной памяти, её объема. В противном случае, в сети может возникнуть джиттер, задержки и потери пакетов. Также это касается твоего сетевого интерфейса. Ниже я привел примерные технические характеристики, которых бы хватило с аппаратной точки зрения

- CPU: от 4 ядер
- RAM: от 6 GB
- Сетевая карта: Full-Duplex, пропускная способность от 1 GBPS

NAT, Masquerading

Ты спуфишь сеть на канальном уровне (L2). Это нужно сделать и на сетевом (L3). Без NAT ты видишь трафик только в одну сторону. Чтобы видеть не только входящий трафик, но и исходящий - включай маскарадинг

```
Inguz@Inguz:~$ sudo iptables -t nat -A POSTROUTING -o ethX -j MASQUERADE
```

NAT Helper

В сети может быть FTP-сервер, к которому пользователи подключаются удаленно. После настроенного NAT, FTP-трафик может флапать и возникает потеря пакетов. FTP плохо дружит с NAT. Поэтому во избежание проблем стоит настроить NAT helper через сетевой модуль ядра nf_conntrack. Он поможет протащить дальше FTP-трафик и потерь быть не должно

```
Inguz@Inguz:~$ sudo modprobe nf_conntrack
Inguz@Inguz# echo "1" > /proc/sys/net/netfilter/nf_conntrack_helper
```

ICMP Redirect

Это специальное сообщение протокола ICMP, уведомляющее хост о том, что передача пакета будет воспроизведена по другому маршруту. ICMP Redirect сообщения стоит отключить в твоей ОС во время проведения атаки. Системы безопасности IPS/IDS могут подать сигнал тревоги, если их сенсоры в сети увидят ICMP Redirect(в пользовательской сети это может быть расценено как аномалия), вследствие чего твоя атака перестанет быть секретом. Решается этот вопрос так.

```
Inguz@Inguz:~$ sudo sysctl -w net.ipv4.conf.all.accept_redirects=0
Inguz@Inguz:~$ sudo sysctl -w net.ipv6.conf.all.accept_redirects=0
```

TTL Shifting, Traceroute Evasion

После проведения MITM-атаки, есть достаточно высокая вероятность обнаружения твоей атаки. Сетевые администраторы могут тебя найти с помощью трассировки, где будет обнаружен твой адрес. Также не забывай о значении TTL, которое может быть отлично от TTL легитимного роутера. Я выяснил, что если в цепочке PREROUTING смещать значение TTL с использованием инкремента +1, это позволит спрятаться от трассировки пакетов(вывода команды) со стороны пользователя и, грубо говоря, стать невидимкой во время MITM-атаки. Т.е. в момент проведения атаки, тебя не найдет трассировка пакетов, что дает достаточно большое преимущество, ибо тебя не палит ни пользователь, ни сетевой администратор

```
sudo iptables -t mangle -A PREROUTING -i ethX -j TTL --ttl-inc 1
```

Трассировка пакетов до смещения TTL

Трассировка пакетов после смещения TTL

ARP

Это протокол канального уровня, который был создан для сопоставления IP-адреса и MAC-адреса. Ибо во время передачи данных внутри сети, только IP-адреса знать недостаточно, ибо инкапсуляция. В механизме протокола ARP нет никаких защитных механизмов, что и вызывает проблемы с безопасностью. Да и на основе ARP разработана одна из самых популярных атак - ARP Spoofing.

ARP Spoofing - это сетевая атака, возникающая в результате отравления ARP-таблицы на целевых компьютерах атакующим. Он генерирует специальные ARP IS-AT кадры и изменяет структуру ARP-таблиц таким образом, чтобы трафик легитимных компьютеров шел в твой хост.

Subnet Masks

Во время ARP-спуфинга не спеши брать большие маски. Даже самую распространенную - /24. Данная маска обеспечивает адресами порядка 254 устройства (-2 адреса на шлюз и **BCAST**). Если заспуфить всю /24 маску, твое устройство может не выдержать такую нагрузку, что может вызвать в сети джиттер и задержки, которые в свою очередь нарушают нормальную работу сети. Советую брать маски поменьше, самому разделять, скажем так, этот скоуп. Считай IP-адреса, разбивай подсеть на более маленькие кусочки. Порой это будет являться кропотливой работой, но это поможет избежать побочной DoS-атаки.

Восстановление ARP-таблицы после атаки

Чтобы корректно прекратить этот спуфинг, необходимо после завершения атаки отправить восстанавливающие ARP IS-AT кадры, которые восстанавливают структуру ARP-таблиц. Да, ARP-записи в принципе являются динамическими и хост автоматически генерирует ARP-запрос, они это делают каждые 300 секунд. Однако, в сети есть различное оборудование, оно может зависнуть, повести себя некорректно, так и не отправив запрос ARP, что в конечном счете приводит к разрыву сетевой связности. Такое поведение характерно для различных служебных сервисов внутри сети, Docker-контейнеры, виртуальные машины, SCADA и т.д. Поэтому проследи

этот момент и после прекращения спуфинга - верни состояние таблиц до твоей атаки. Кстати говоря, так умеет **Ettercap** - проверенный временем инструмент.

STP

STP (Spanning Tree Protocol) - протокол канального уровня, позволяющий избежать широковещательного шторма, путем блокировки избыточных связей между коммутаторами. Создан 37 лет назад. Как и ARP, не имеет механизмов аутентификации, однако снабжен функциями безопасности, такими как: Root Guard, Loop Guard, BPDU Guard.

Дерево STP возможно атаковать, отправляя специальный BPDU-кадр с наименьшим значением приоритета. В STP сложилось так: чем ниже приоритет коммутатора, тем выше вероятность, что именно он и станет корневым коммутатором (что и хочет сделать атакующий)

Ограничения атаки

- Атака на STP даёт **только частичный** MITM.
- **Тебе помешают** функции безопасности BPDU Guard, Loop Guard, Root Guard, Portfast.

Восстановление дерева STP после атаки

После прекращения атаки, коммутаторы в домене STP сами вернут прежнюю структуру дерева. STP-коммутаторы общаются друг с другом с помощью специальных BPDU "Hello" сообщений, а рассылаются они каждые 2 секунды. Поэтому можешь просто прекратить инжект, дерево STP сделает всю работу.

FHRP

FHRP (First Hop Redundancy Protocol) - протоколы резервирования шлюза, призванные повысить уровень отказоустойчивости на уровне маршрутизации. Два или более роутеров входят в один логический процесс FHRP протокола, далее назначаются MASTER и SLAVE-роли соответственно. Сам же MASTER-маршрутизатор будет обслуживать некоторый виртуальный IP-адрес группы FHRP, который в свою очередь будет являться шлюзом по умолчанию для хостов. Важно понимать, что во время использования FHRP возникает псевдобалансировка, того же Round-Robin здесь не будет (исключение: GLBP)

На FHRP протоколы есть вектор MITM-атаки, когда атакующий выполняет инъекцию пакета с наивысшим значением приоритета, тем самым перехватывая MASTER-роль. **Основной здесь нюанс - делать всё быстро:** настройка статического маршрута, NAT. Любая заминка во время эксплуатации FHRP может нарушить работу сети, а атака перестанет быть секретом.

FHRP Injection

Процесс инъекции может осуществляться разными инструментами: Loki, Scapy, Yersinia. На вкус и цвет, главное чтобы понимал как это работает. Эти инструменты сгенерируют необходимые FHRP-пакеты со значением 255 и отправят их в сеть.

Static Routing & NAT

После инъекции необходимо как можно быстрее решить вопрос с маршрутизацией. Удали предыдущий маршрут по умолчанию во избежание петли маршрутизации и пропиши новый через бывшего MASTER-маршрутизатора. А также NAT (Masquerading)

```
Inguz@Inguz:~$ sudo route del default
Inguz@Inguz:~$ sudo route add -net 0.0.0.0 netmask 0.0.0.0 gw XXX.XXX.XXX.XXX
Inguz@Inguz:~$ sudo iptables -t nat -A POSTROUTING -o ethX -j MASQUERADE
```

Восстановление домена FHRP после атаки

Маршрутизаторы внутри FHRP-домена общаются между собой с помощью служебных Hello-сообщений. Они необходимы для того, чтобы отслеживать состояние между соседями внутри логической группы FHRP. Когда ты становиться ложным FHRP-маршрутизатором с наивысшим значением приоритета, ты также отправляешь эти самые Hello-пакеты. Если внутри логической группы FHRP хотя бы один маршрутизатор перестал отправлять Hello-пакеты - он просто выпадет из этого домена FHRP и MASTER/SLAVE будут перестроены. Поэтому если ты внезапно прекратишь спуфинг, инъекцию пакета с наибольшим приоритетом - домен FHRP сам перестроится в предыдущее состояние. Учитывая даже таймеры тех же HSRP/VRRP/GLBP, сеть перестроится достаточно быстро.

Более подробную информацию о FHRP-протоколах и MITM-атаке с помощью них, ты сможешь узнать из моей статьи FHRP Nightmare. Советую ознакомиться.

DHCP

DHCP (Dynamic Host Configuration Protocol) - с помощью данного протокола, компьютеры в сети могут получать настройки адреса автоматически. По большей части облегчает жизнь сетевому администратору, ибо не приходится обходить каждый компьютер и прописывать адрес вручную.

На DHCP-сервер существуют две атаки:

- **DHCP Exhaustion:** Истощение адресного пространства, путем рассылки ложных `DISCOVER`-сообщений от разных MAC-адресов.
- **DHCP Spoofing:** Подмена легитимного DHCP-сервера, с дальнейшей целью MITM-атаки. Это возникает в результате того, что по DHCP передается информация об IP-адресе шлюза по умолчанию. Обычно ложный DHCP-сервер так и настраивается, чтобы шлюзом по умолчанию был компьютер атакующего.

С DHCP Exhaustion есть такой неприятный момент - атака создает огромный шум в эфире из-за рассылки `DHCPDISCOVER` сообщений, достаточно высокая вероятность получить леща от системы Storm-Control. Да и заказчик сети будет не особо рад тому, что его DHCP-сервер в сети больше не сможет обслужить новые устройства.

Однако по своему опыту могу сказать, что часто DHCP-сервер находится в другом канальном сегменте. Это классический дизайн корпоративной сети, когда служебные сервера находятся в другом VLAN-сегменте. Я это всё к тому, что обычно сначала истощают легитимный DHCP-сервер, а потом создают ложный: таким образом повышается вероятность того, что клиенты получат адрес автоматически **именно от нас.**

DHCP Relay

Представляет собой ретранслятор DHCP-запросов. Помогает компьютерам получить адрес от сервера, который находится в другом канальном сегменте. Начало этой идеи берется с того утверждения, что трафик протокола DHCP на канальном уровне - широковещательный. И каждый отдельный VLAN-сегмент - это отдельный широковещательный домен, в рамках которого будут летать эти запросы. Естественно без механизма ретрансляции, DHCP-сервер в другом канальном сегменте не сможет обслуживать другие хосты в других канальных сегментах, для этого и

придумали DHCP Relay. К слову, DHCP Relay настраивается на роутере, с указанием адреса DHCP-сервера.

DHCP Spoofing Impact

Сколько бы я ни хотел себе импакта, о целостности сети и бизнес-процессах - тоже стоит думать. Поэтому, мне кажется самый оптимальный вариант здесь - просто поднять фейковый DHCP-сервер, но при этом, без предварительной атаки истощения на легитимный сервер. Представим, ты находишься в некотором VLAN-сегменте, например - VLAN 100. Этот VLAN 100 является отдельным канальным сегментом и соответственно, отдельным широковещательным доменом. Тебе ничего не мешает здесь поднять свой DHCP-сервер и начать саму атаку. Широковещательный запрос от компьютеров будет лететь в твоём VLAN, и скорее всего, твой фейковый сервер и примет этот запрос. Быстрее, чем легитимный, находящийся в другом канальном сегменте. Грубо говоря, ты будешь здесь быстрее, ибо опять же, ты находишься в самом широковещательном домене. В другом же случае с DHCP Relay, запросу придется подняться на маршрутизатор и под влиянием процесса маршрутизации отправиться до легитимного DHCP-сервера. Согласись, это гораздо медленнее, чем наш DHCP-сервер, где мы уже находимся в этом канальном сегменте, и мы можем триггериться на любые DHCP-запросы.

NAT

Здесь также понадобится NAT

```
Inguz@Inguz:~$ sudo iptables -t nat -A POSTROUTING -o ethX -j MASQUERADE
```

Риск

Данная атака опасна тем, что что пентестер может забыть о таком понятии, как DHCP Lease Time. Если ты раздашь адрес автоматически со слишком большим Lease Time, но при этом прекратишь MITM раньше этого срока - легитимные компьютеры в сети потеряют связность, и они запросят новый адрес только тогда, когда истечёт таймер DHCP Lease Time. Поэтому перед проведением MITM-атаки, грамотно рассчитай Lease Time, в течении какого времени ты хочешь перехватывать трафик.

Outro

В этой статье я немного поделился своим опытом, как правильно проводить MITM-атаки. Я не рассчитываю на открытие Америки этой статьей и ни в коем случае не считай эту статью полным мануалом по MITM-атакам. Сетевые атаки достаточно старые и статичные, их используют многие, но не всегда правильно и безопасно.

Links

https://habr.com/ru/post/685072 - FHRP Nightmare