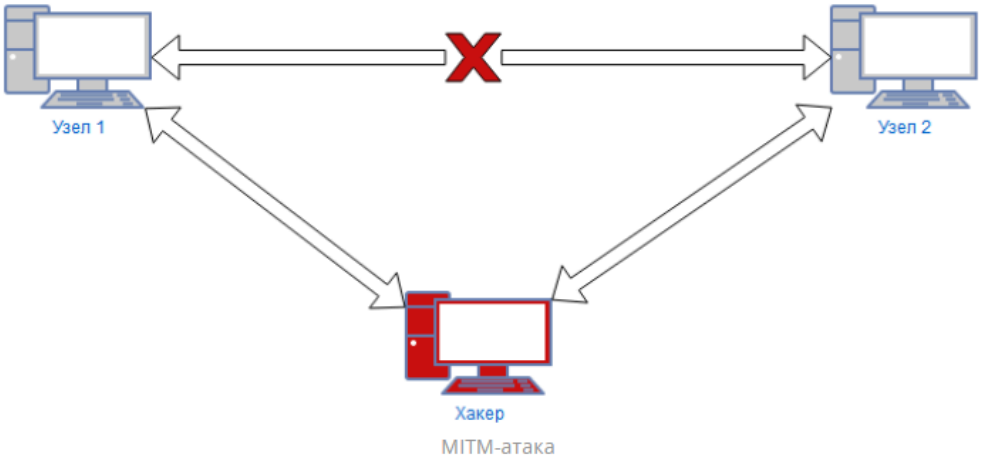
Введение

Схема атаки представлена на рисунке:



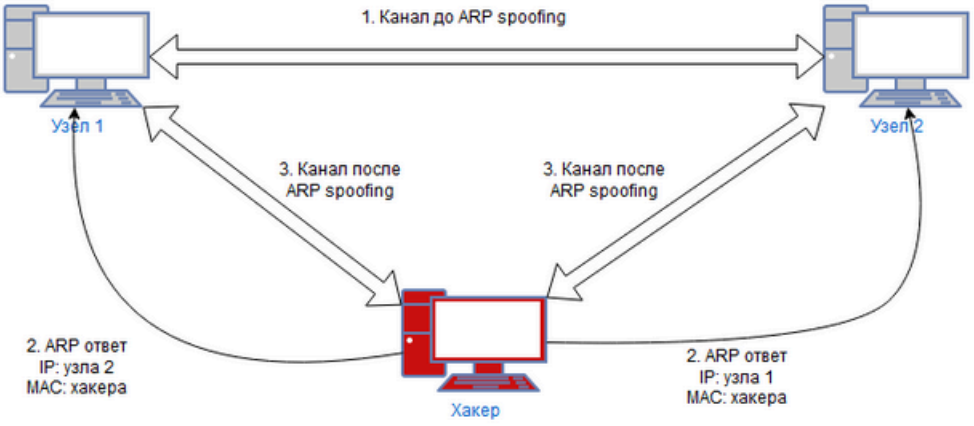
MITM-атака

MITM (Man-in-the-Middle) — атака типа «человек посередине», основной концепцией которой является встраивание в цепочку сообщений между двумя узлами. Хакер имитирует работу сразу двух узлов таким образом, что они уверены в приватности своего диалога.
Основная цель атаки — кража конфиденциальных данных, которые проходят через компьютер хакера. Это могут быть пароли, данные банковских карт, сообщения в социальных
Однако есть ситуации, при которых атака *MITM* будет малоэффективна. Основной враг для хакера в таком случае — шифрование. Мы можем «встать посередине» и даже просматривать трафик, но в этом не будет никакого смысла, если все пакеты будут зашифрованы.
Таким образом, проведение атаки *MITM* на корпоративную сеть, в которой правильно реализовано шифрование (например, протокол *TLS 1.3* с алгоритмом *ECDHE*), не приведет к желаемому результату.
Один из сценариев, когда атака MITM может быть осуществлена, — **подмена сертификата**. Однако этот метод подразумевает установку «хакерского» сертификата в качестве доверенного на устройстве жертвы: браузер жертвы может предупредить её о том, что сертификат не является доверенным, а хакеру нужно учесть время действия сертификата, правильность подписи, его массовость (наличие 1 экземпляра сертификата на весь Интернет вызывает сомнения в его официальности).

Spoofing — класс сетевых атак, при которых хакер маскируется под другое устройство.

Протокол *ARP* отвечает за сопоставление *MAC*-адреса и *IP*-адреса устройства. Таким образом, у каждого устройства в сети есть своя *ARP*-таблица известных ему «соседей». Процесс составления этой таблицы достаточно прост — отправляется широковещательный *ARP*-запрос с требуемым *IP*-адресом. Запрашиваемое устройство при получении данного запроса, отправляет в ответ пакет, содержащий его *MAC*-адрес.
ARP-spoofing (или **ARP-poisoning**) — сетевая атака, которая основана на недостатках протокола ARP. Данный протокол не проверяет подлинность ARP-запросов и ARP-ответов, что позволяет «встать посередине»

в атакуемой сети.

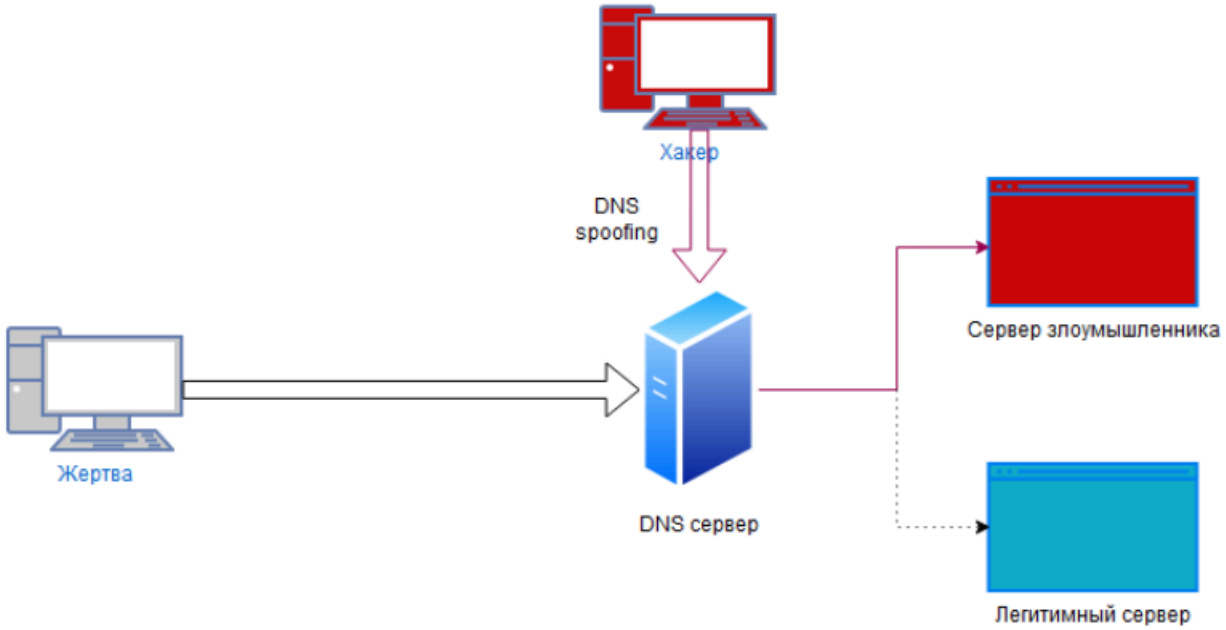


Проведение *ARP-spoofing*

Рассмотрим методы защиты, которые могут быть применены, чтобы избежать данной атаки:

- Одним из самых очевидных вариантов защиты является **отключение динамического обновления ARP-таблицы.** Таким образом хакер не сможет в любой момент отправить *ARP*-ответ устройству жертвы.
- Еще одним способом защиты является **ARP Guard.** Это своеобразная «база данных» неизменяемых *ARP*-записей. Для предотвращения подмены соответствия *IP*-адресов и *MAC*-адресов важных узлов сети (например, маршрутизаторов или коммутаторов), *ARP Guard* при получении *ARP*-ответа с *source*-адресом важного узла сбрасывает данный пакет как вредоносный.

Атака DNS-spoofing (её ещё называют *DNS cache poisoning*) направлена на изменение *DNS*-кеша с целью перенаправления жертв на свой сайт/узел/приложение.



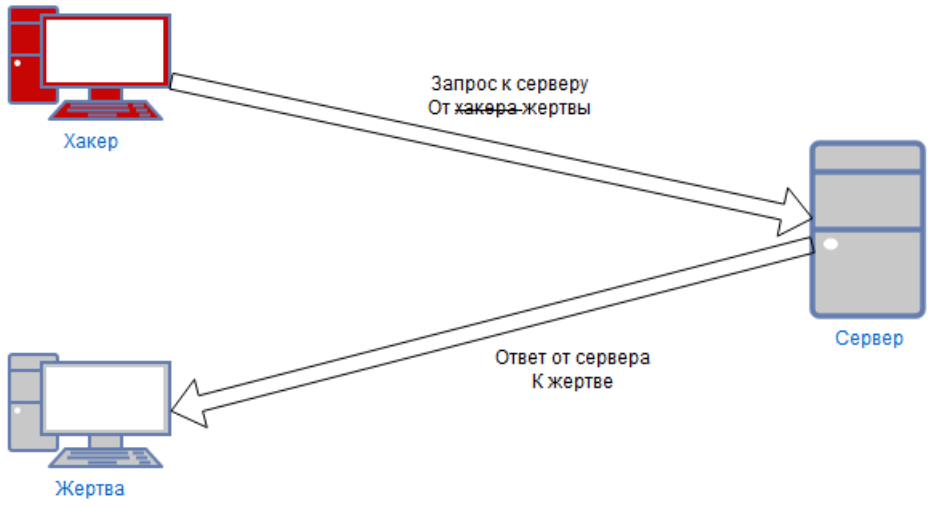
Атака *DNS-spoofing* в классическом понимании

Один из сценариев использования MITM DNS-spoofing:

Хакер «встает посередине», а дальше с помощью *Scapy* в автоматическом режиме перехватывает *DNS*-ответы, направленные жертве, модифицирует их и отправляет к пункту назначения. Таким образом, жертва осуществляет коммуникацию с *DNS*-сервером, но по пути нужный адрес сервера будет подменен адресом злоумышленника.
Скрипт

```
https://www.thepythoncode.com/article/make-dns-spoof-python
```

IP-spoofing-Эта атака зачастую применяется для *DOS*-атак на конкретный узел
Механизм **IP-spoofing** заключается в подмене *source IP*-адреса пакета, так что хакер представляется другим компьютером. Один из вариантов использования — направлять на сервер сообщения от лица жертвы с целью вызвать шквал ответов от сервера, направленных жертве.
Реализовать *IP-spoofing* можно двумя способами: либо как продолжение *MITM*-атаки (перехватываем пакеты и автоматически подменяем им *source*-адрес), либо используя *Scapy* для создания пакета «с нуля» для специфической атаки.



Механизм действия IP-spoofing

Защититься от данной атаки достаточно сложно, потому что нет надежного способа проверить подлинность *source*-адреса. Однако несколько способов всё-таки есть:

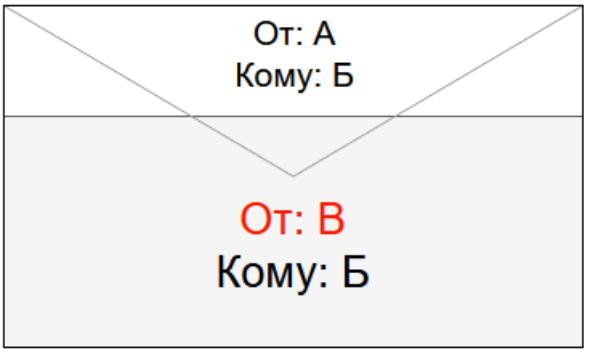
- **Включение фильтрации на маршрутизаторе.** Данная мера позволяет фильтровать пакеты, которые представляют собой аномалии. Например, пакет пришел из внешней сети, но *Source-IP* указан внутренний.
- **Обратная проверка.** Она реализуется отправлением пакета на *Source*-адрес с целью проверки существования в сети устройства с данным *IP* (если ответ на этот пакет не получен, значит, устройства не существует). Если же ответ получен, то сверяются некоторые поля пришедшего пакета с полями изначально полученного сообщения (например, хакер забыл подменить *TTL* и отправил *IP-spoofing* пакет с *TTL 64,* в то время как настоящее устройство с этим *IP* имеет *TTL 128*).

Обнаружить хосты можно nmap -pn либо же netdiscover

Email Hijacking

Данная *spoofing*-атака является примером того, что может произойти при использовании стандартных конфигураций.
Механизм **Email Hijacking** (или *Email Spoofing*) заключается в подмене отправителя электронного письма. Электронные письма содержат два упоминания отправителя: в *SMTP* «конверте» и в содержимом пакета. «Конверт» необходим для сервера, чтобы распознать, куда отправить письмо и от кого оно идет. А содержимое конверта необходимо именно получателю, чтобы увидеть, кто прислал письмо. Таким образом, хакер, изменив

отправителя в содержимом пакета (поле *«From»*), может отправлять письма от лица другого человека.



Подмена поля *«from»* в письме при *Email Hijacking*

- Защита от данной атаки достаточно тривиальна — можно сверять значения в «конверте» со значениями заголовков в содержимом пакета.
- Кроме того, можно реализовать систему аутентификации писем с помощью цифровых подписей. Подробнее об этой системе можно прочитать здесь.

Опасность данной атаки состоит в использовании её для фишинговых атак. Как известно, фишинговые атаки — один из самых актуальных способов проникновения в корпоративные сети, заражения узлов и кражи конфиденциальных данных сотрудников.

Ettercap

Установить **Ettercap** на *Kali Linux* можно одной командой:

```
sudo apt-get install ettercap-graphical
```

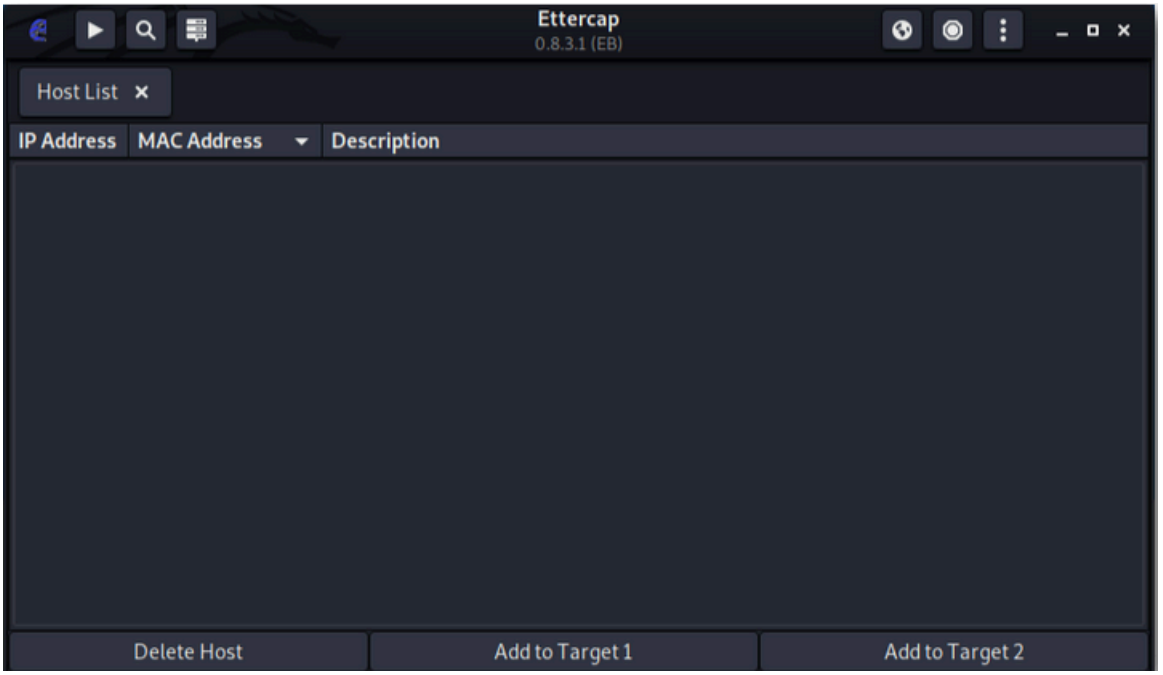
Данный инструмент позволяет проводить множество атак
Перед запуском утилиты желательно перевести устройство в режим форвардинга пакетов (своего рода режим маршрутизатора).
Сделать это можно командой:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

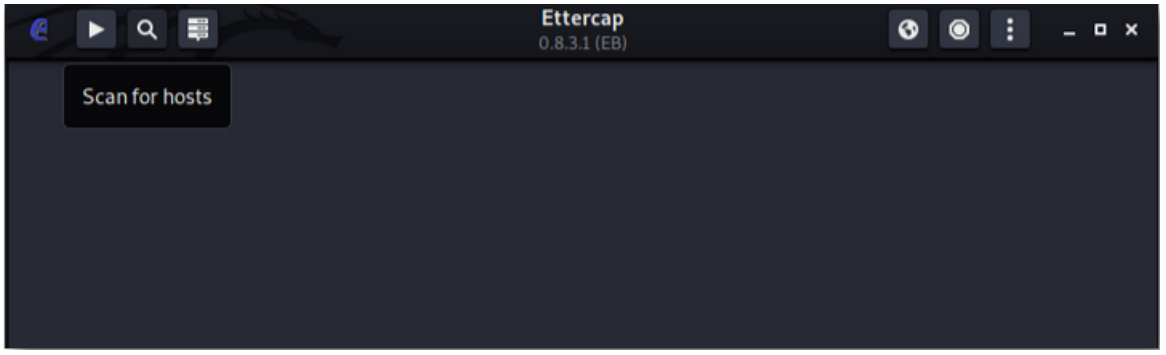
Итак, запускаем утилиту с графическим интерфейсом командой:

```
sudo ettercap -G
```

Первое место, куда нам следует заглянуть — *«Hosts list»*. Это список доступных в данной сети хостов, которые мы можем использовать в качестве жертв.

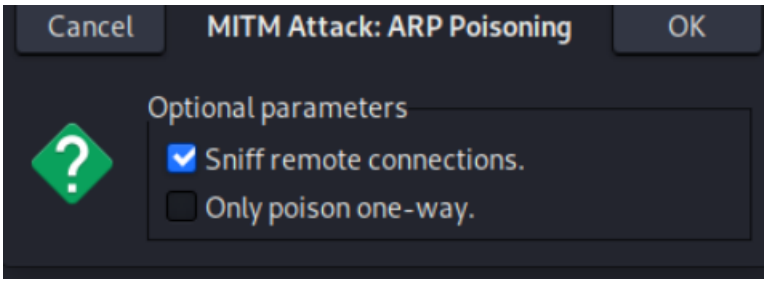


Для того чтобы провести сканирование сети и выявить *IP*-адреса, необходимо нажать на кнопку *«Scan for hosts»:*

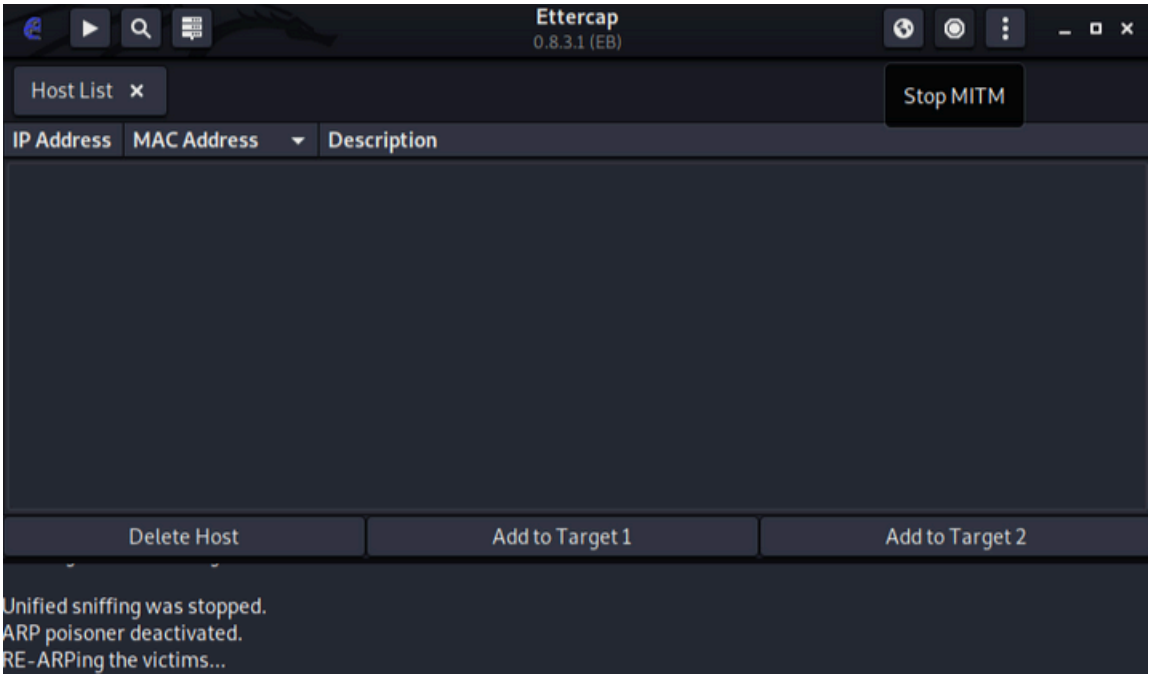


После того как мы просканировали хосты и получили список адресов, мы можем выбрать два узла (две жертвы), которые будут использованы для встраивания посередине и проведения атаки *ARP-spoofing.* Для этого достаточно выбрать один из узлов и нажать на кнопку *«Add to Target 1».* Соответственно, те же действия выполняются для второго хоста и кнопки *«Add to Target 2»*.

Теперь мы можем приступать к атаке.



Чтобы остановить *ARP-spoofing* и вернуть таблицы устройств в исходные состояния, необходимо нажать кнопку *«Stop MITM»:*



Scapy

Scapy — утилита, позволяющая создавать и модифицировать пакеты огромного количества протоколов. Установить *Scapy* можно из командной строки командой:

```
apt install scapy
```

Либо скачав исходники с официального *Git*-репозитория:

```
git clone https://github.com/secdev/scapy.git
```

Данная утилита может быть использована либо как интерактивная оболочка (запущена из командной строки), либо как программная библиотека (можно подключить её как библиотеку к *Python*-скрипту).
Рассмотрим основные команды при работе в интерактивной оболочке *Scapy*:

- `ls()` — список доступных протоколов.
- `help(<object>)` — справка о возможностях объекта.
- `<object>.show()` — текущие параметры объекта.
- `<object name> = <protocol>()` — создание пакета протокола.
  Чтобы изменить поле объекта, используют следующий синтаксис:

```
<object>.<field> = <new value>
```

Пример:

```
ip = IP()
ip.id = 4
```

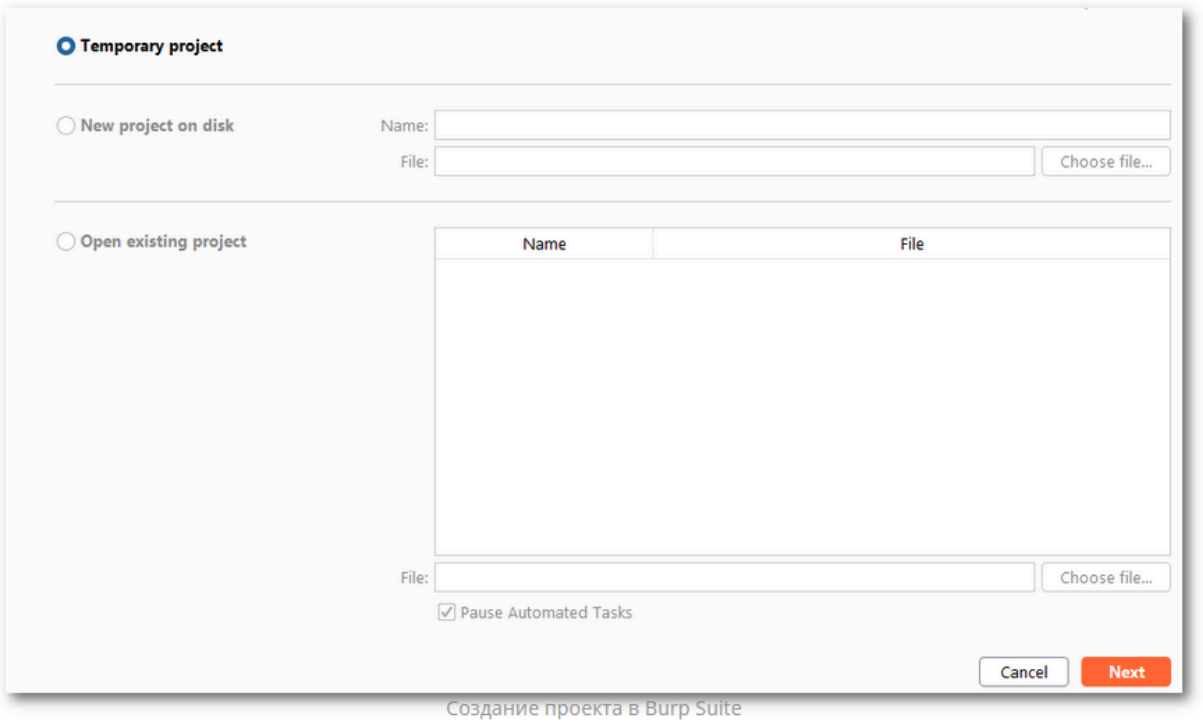
Основной метод отправки сообщения:

```
send(<packet>)
```

Ознакомиться с документацией *Scapy* можно по ссылке.

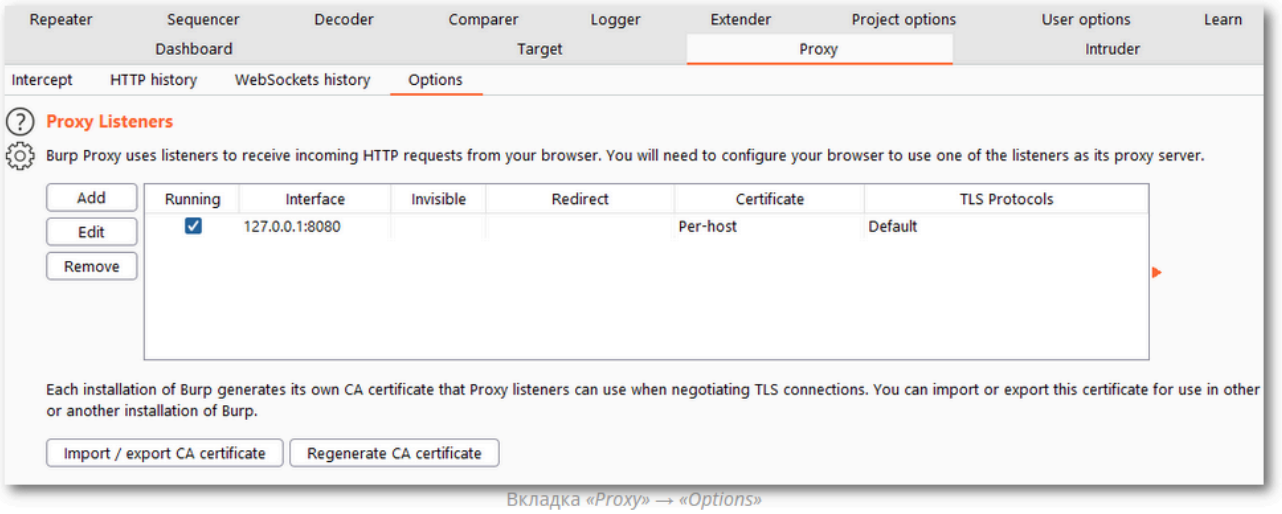
Burp (MITM)

1. При запуске утилиты создаём новый временный проект:



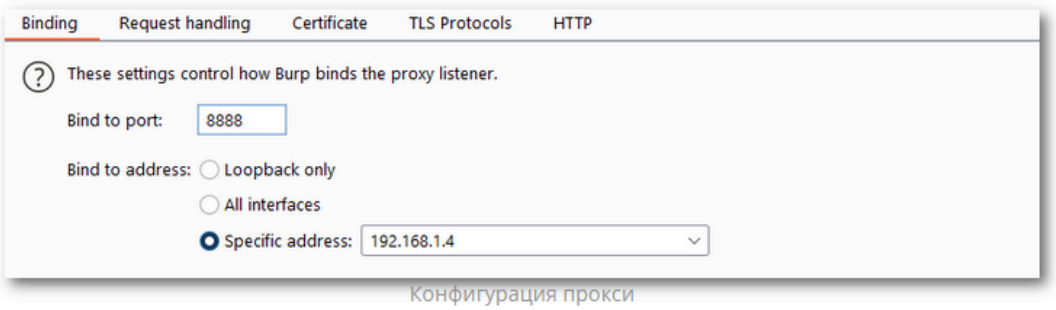
Создание проекта в Burp Suite

2. Нам необходимо настроить прокси, через который будет идти трафик к браузеру. Для этого перейдем во вкладку *«Proxy»* → *«Options»* и нажмем кнопку *«Add»*:



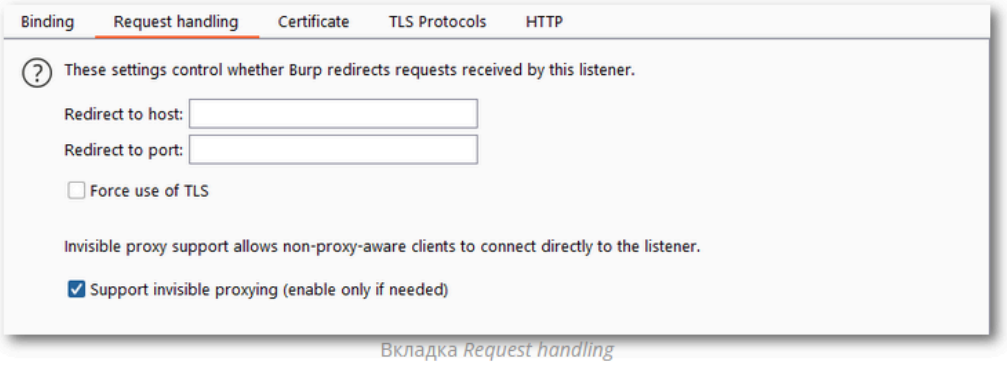
Вкладка *«Proxy»* → *«Options»*

3 Вводим параметры нашего прокси. *IP*-адрес — адрес атакующего (в данном случае нашего компьютера):



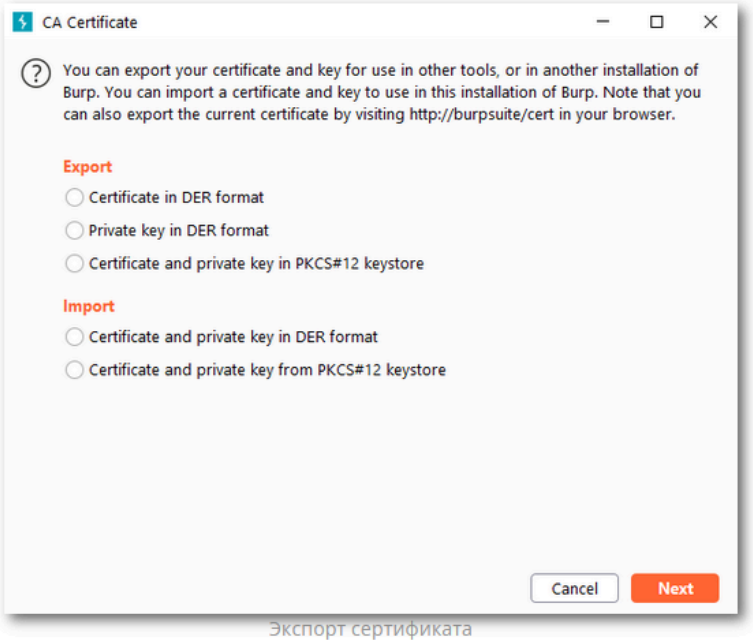
Конфигурация прокси

4 После конфигурации адреса перейдем во вкладку *«Request handling»* и поставим галку на *«Support invisible proxying»:*



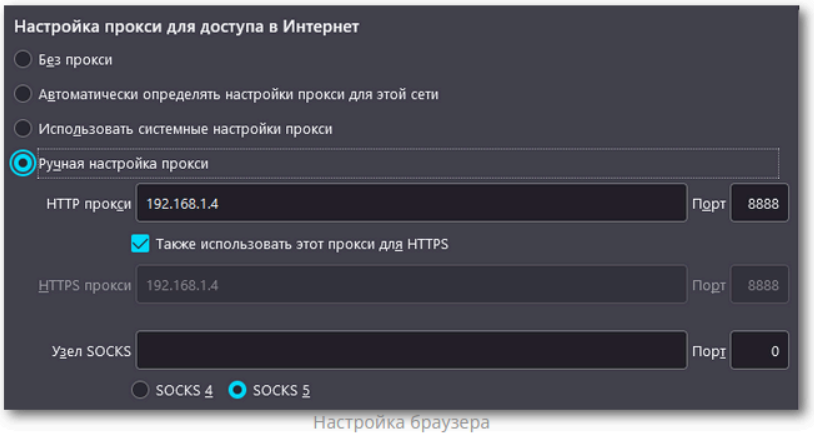
Вкладка *Request handling*

5 После конфигурации прокси нам необходимо экспортировать сертификат и заверить браузер в его подлинности. Для этого вернемся во вкладку *«Proxy»* → *«Options»* и нажмем *«Import /Export CA certificate».* Указываем *«Certificate in DER format»* и жмём *«Next»:*



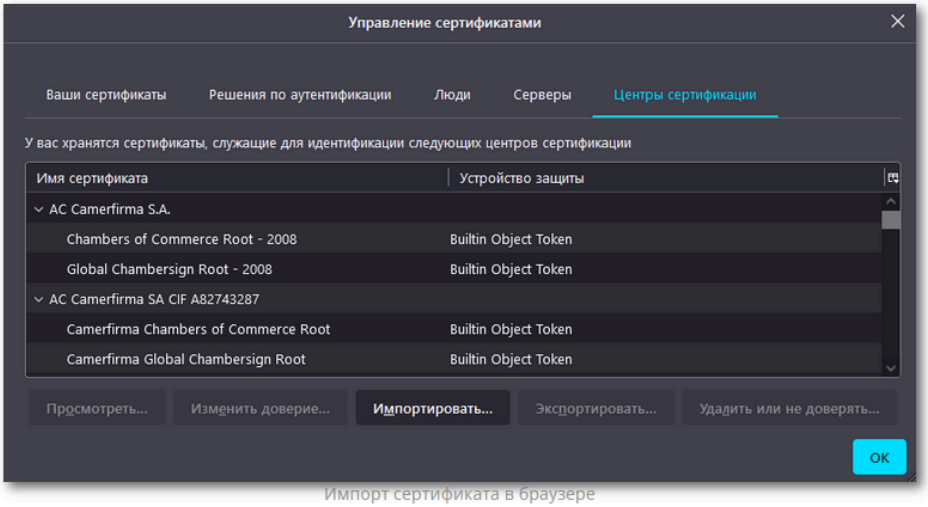
Экспорт сертификата

6 После того как наш сертификат экспортирован, переходим в браузер и идём во вкладку конфигурации сети (как правило, она находится в основных настройках). Здесь выбираем «Ручная настройка прокси» и указываем наш прокси-сервер, который мы конфигурировали в *Burp Suite*.



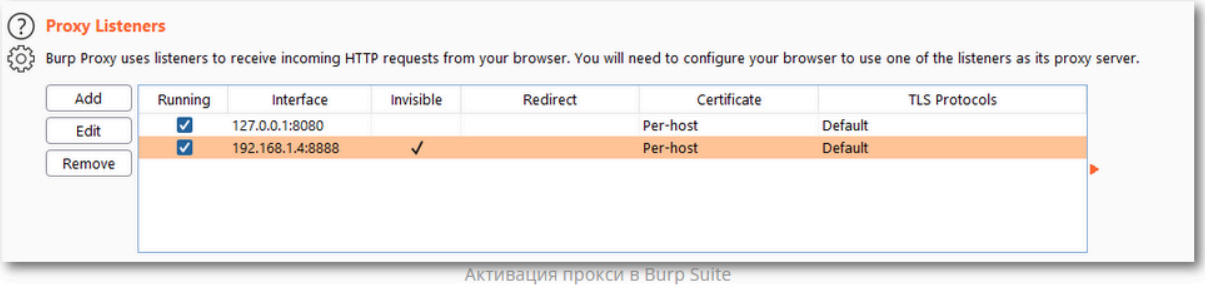
Настройка браузера

7 Далее импортируем свежесозданный сертификат. Как правило, управление сертификатами находится во вкладке «Приватность и защита». Нажимаем «Импортировать» и выбираем сертификат *Burp Suite:*



Импорт сертификата в браузере

8 Удостоверьтесь в том, что напротив прокси стоит галка в колонке *«Running»:*



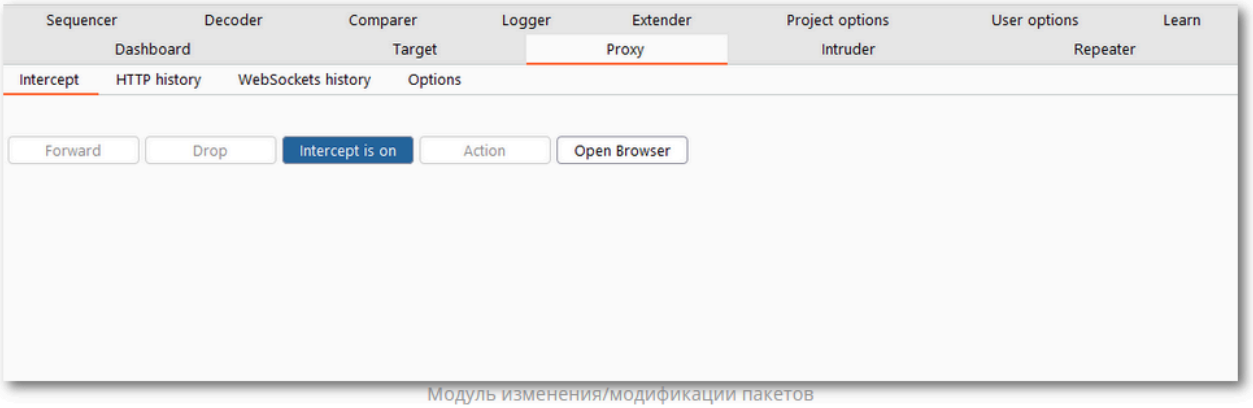
Активация прокси в Burp Suite

9 После всех вышеописанных действий во вкладке *«Proxy»* → *«HTTP history»* должны появиться перехваченные запросы:

https://yandex.ru	POST	/clck/jclck/text=hello%20world/pos=1...	✓		200	449	script	ru/	
https://yandex.ru	GET	/search/?lr=213&msid=1636663203.8...	✓		200	332859	HTML		Ð¯Ð½Ð´ÐµÐºÑ
https://yastatic.net	GET	/s3/web4static/_/v2/tXMJyG2-k94nqB...			200	9478	script	js	
https://yastatic.net	GET	/nearest.js			304	209	script	js	
https://yandex.ru	POST	/ick/r	✓		200	347	JSON		
https://yastatic.net	GET	/s3/web4static/_/v2/1VPXEK1EuSvY_re_...			200	860	XML	svg	
https://yastatic.net	GET	/s3/web4static/_/v2/static/media/Arro...			200	935	XML	svg	
https://yastatic.net	GET	/s3/web4static/_/v2/static/media/Arro...			200	932	XML	svg	
https://mc.yandex.ru	GET	/watch/731962?wmode=7&page-url=...	✓		200	848	JSON		
https://yastatic.net	GET	/s3/web4static/_/v2/static/chunks/56fd...			200	929	script	js	
https://yastatic.net	GET	/s3/web4static/_/v2/static/chunks/5e91...			200	865	script	js	
https://yastatic.net	GET	/s3/web4static/_/v2/ECE2HC-YfbB_OH...			200	32749	script	js	
https://yastatic.net	GET	/s3/web4static/_/v2/static/chunks/18d5...			200	1732	script	js	
https://yastatic.net	GET	/s3/web4static/_/v2/static/chunks/7d61...			200	1075	script	js	
https://yastatic.net	GET	/s3/web4static/_/v2/static/chunks/3dd...			200	1614	script	js	
https://yastatic.net	GET	/s3/web4static/_/v2/static/chunks/abb...			200	1141	script	js	
https://yastatic.net	GET	/s3/web4static/_/v2/static/chunks/9b65...			200	3963	script	js	

Перехваченные пакеты

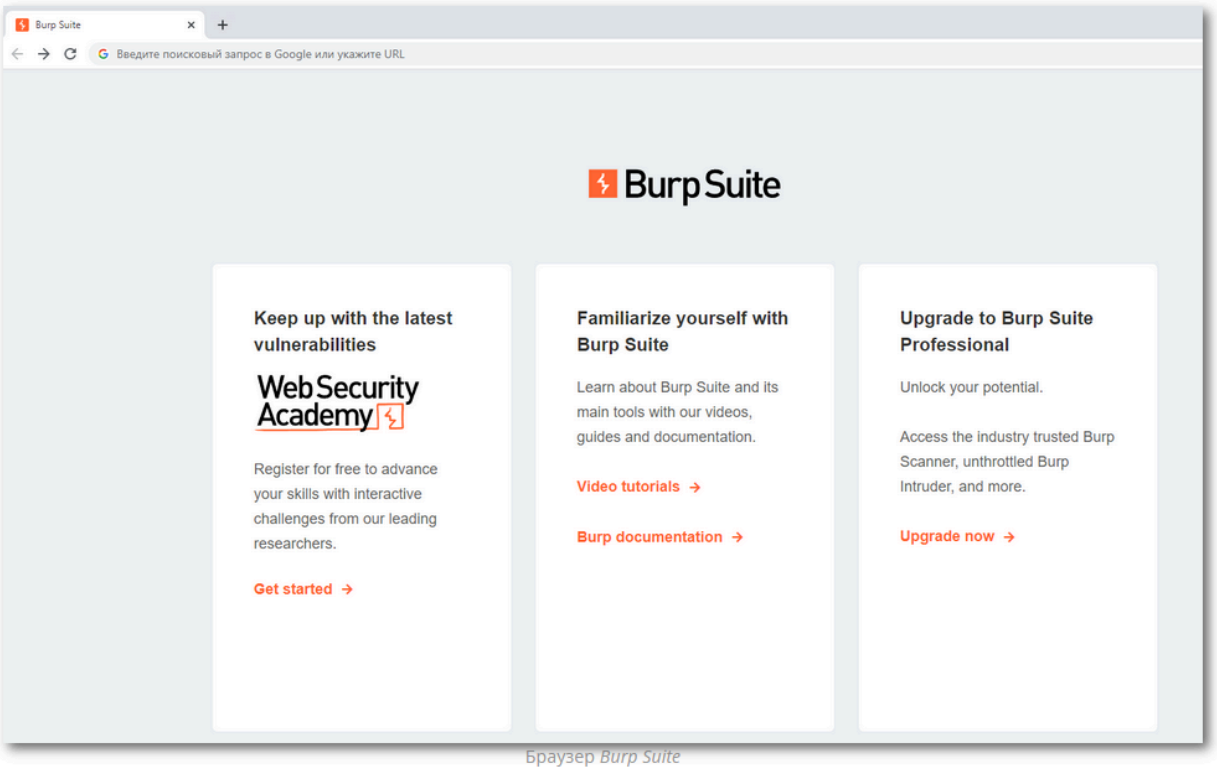
10 Мы наладили взаимодействие *Burp Suite* с нашим браузером, однако данная утилита предоставляет свои «услуги» браузера на базе *Chromium*. Для этого необходимо перейти во вкладку *«Proxy»* → *«Intercept»*:



Модуль изменения/модификации пакетов

11 **Модуль Intercept** позволяет на лету изменять запросы к браузеру (при условии активной кнопки *«Intercept is on»*.

Для того чтобы протестировать данную возможность, нажимаем кнопку *«Open Browser»* и попадаем в браузер от *Burp Suite:*



Браузер *Burp Suite*

12 Теперь при любых запросах *Burp Suite* перед отправкой пакета будет выводить в панели содержимое запроса, которое мы можем менять как нам захочется.

Процесс изменения пакета

С помощью данного модуля мы можем тестировать поля ввода на различных веб-сайтах и приложениях, а также отслеживать информацию, отправляемую конкретному хосту.

Использование Wireshark для декрипта трафика

Для того чтобы расшифровать защищенный трафик, необходим секрет **(pre master secret key),** используемый во время рукопожатия и установления сессии между сервером и клиентом.

Для того чтобы его получить, нужно выполнить следующие шаги:
Закройте все браузеры.

Установите браузер Chromium (в Mozilla по умолчанию это не работает):

```
sudo apt update $$ sudo apt install chromium -y
```

Откройте терминал для доступа к командной строке.

Установите переменную среды SSLKEYLOGFILE для своей учетной записи, используя следующий синтаксис команды:

```
export SSLKEYLOGFILE="/home/<имя_аккаунта>/sslkeyfile"
```

Запустите перехват пакетов с помощью приложения Wireshark.

Запустите Chromium с того же терминала. Например:

```
chromium &
```

Примечание. Вы должны запустить браузер из того же командного терминала, поскольку переменная сеанса устанавливается только для данной сессии терминала.

Откройте веб-приложение и начните переходить по сайтам, поддерживающим протокол HTTPS. При этом ключи сеанса SSL регистрируются в файле sslkeyfile.

Остановите перехват пакетов и сохраните файл в своей клиентской системе.
Примечание. После выполнения процедуры загрузки файла журнала ключей SSL в Wireshark и просмотра расшифрованного файла захвата пакетов вы можете удалить файл журнала SSL, чтобы отменить изменения, введя следующую команду: `Export SSLKEYLOGFILE=""`.

Загрузите файл журнала ключей SSL в Wireshark.
Откройте Wireshark в своей клиентской системе.
Перейдите в «Правка» → «Настройки» → «Протоколы» → TLS.
Примечание. Для версий Wireshark ниже 3.0.0 выберите «Правка» → «Настройки» → «Протоколы» → «SSL». Для Mac перейдите в Wireshark → Настройки → Протоколы → TLS.

В качестве имени файла журнала (Pre)-Master-Secret выберите Обзор и найдите созданный вами файл журнала SSL.
Выберите ОК.
Откройте файл перехвата пакетов в Wireshark.
В окне пакетов Wireshark выберите ранее зашифрованные пакеты, чтобы просмотреть незашифрованные данные приложения.

Net-Creds

Ещё одна утилита, которая может пригодиться нам для проведения MITM, — **Net-Creds.** Это по своей сути обычный *Python*-скрипт, который в автоматическом режиме позволяет вытаскивать нужную информацию из трафика при проведении MITM атаки. Данная утилита значительно упрощает работу хакера.

Мы можем запустить утилиту Net-Creds с использованием контейнера Docker.
Для этого скачаем образ контейнера:

```
sudo docker pull mkesenheimer/net-creds
```

Далее запустим его:

```
docker run -it --rm --name net-creds --network host mkesenheimer/net-creds /bin/bash
```

И запустим скрипт внутри самого контейнера:

```
root@computername:/app/net-creds# ./net-creds.py --help
```

В качестве параметров можно передавать следующие значения:

```
-i <интерфейс> //Задать вручную интерфейс для прослушивания
-v //Запретить обрезку пакетов
-f <ip> //Игнорировать пакеты заданного ip-адреса
-p <файл> //Выполнить поиск в приведенном pcap-файле
```