

Tutorial

DNS Spoofing

Greetings my fellow hackers.

As you may have noticed by my lack of posts, I've been away for a while working on a big project with a team (which won't be finished anytime soon), and I have also been putting together a small side project for Null-Byte that I will be announcing soon. So sorry if I've been lagging, I'm back now and I'm finally making a tutorial. I know that DNS Spoofing has been covered here already by OTW but I feel like I have to make my own input on this. I'll be using Ettercap so that's something original, am I right? Alright, let's get to it.

What Is DNS Spoofing?

DNS Spoofing (sometimes referred to as DNS Cache Poisoning) is an attack whereby a host with no authority is directing a Domain Name Server (DNS) and all of its requests. This basically means that an attacker could redirect all DNS requests, and thus all traffic, to his (or her) machine, manipulating it in a malicious way and possibly stealing data that passes across. This is one of the more dangerous attacks as it is very difficult to detect, but today I will show you both how to perform it and how to detect if it is being performed by somebody else on your network.

Step 1: Preparation

Let's start by booting up Kali Linux, whether it's a Virtual Machine (VM), a native boot, or a dual boot. If you haven't got Kali yet (which you should by now, granted that you're on this website) go get it on the official website.

Make sure you have a working internet connection before you continue and make sure that you are on the same network as your target. This is a LAN (or WLAN) attack and so both the attacker and victim must have the same network gateway. Let me point out in advance that the victim could be running any operating system, it does not matter.

Step 2: Configuring

We now need to edit the Ettercap configuration file since it is our application of choice for today. Let's navigate to `/etc/ettercap/etter.conf` and open the file with a text editor like **gedit** and edit the file. We can use Terminal for that.

```
root@Kali:~# gedit /etc/ettercap/etter.conf
```

So now we want to edit the **uid** and **gid** values at the top to make them say 0 so go ahead and do that.

```
# (at your option) any later version. #
# #
# #
#####

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

[mitm]
arp_storm_delay = 10 # milliseconds
arp_poison_smart = 0 # boolean
arp_poison_warm_up = 1 # seconds
arp_poison_delay = 10 # seconds
arp_poison_icmp = 1 # boolean
arp_poison_reply = 1 # boolean
arp_poison_request = 0 # boolean
arp_poison_equal_mac = 1 # boolean
dhcp_lease_time = 1800 # seconds
port_steal_delay = 10 # seconds
port_steal_send_delay = 2000 # microseconds
```

Now scroll down until you find the heading that says Linux and under that remove both the # signs below where it says “if you use iptables”.

```

#-----
#   Linux
#-----

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

# if you use iptables:
#redir_command_on = "iptables -t nat -A PREROUTING -i %iface -p tcp --dport
#redir_command_off = "iptables -t nat -D PREROUTING -i %iface -p tcp --dport

#-----
#   Mac Os X
#-----

# quick and dirty way:
#redir_command_on = "ipfw -q add set %set fwd 127.0.0.1,%rport tcp from any
#redir_command_off = "ipfw -q delete set %set"

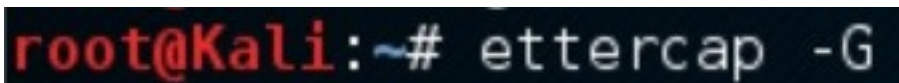
# a better solution is to use a script that keeps track of the rules inserted

```

Great, we're done with the configuration.

Step 3: Ettercap

Now let's run this show by opening Ettercap. You can do it the lame way through launchpad or the cool way using Terminal. I'm going to teach you the cool way. Go ahead and open up Terminal and type:



```

root@Kali:~# ettercap -G

```

[image missing due to format inconsistencies]

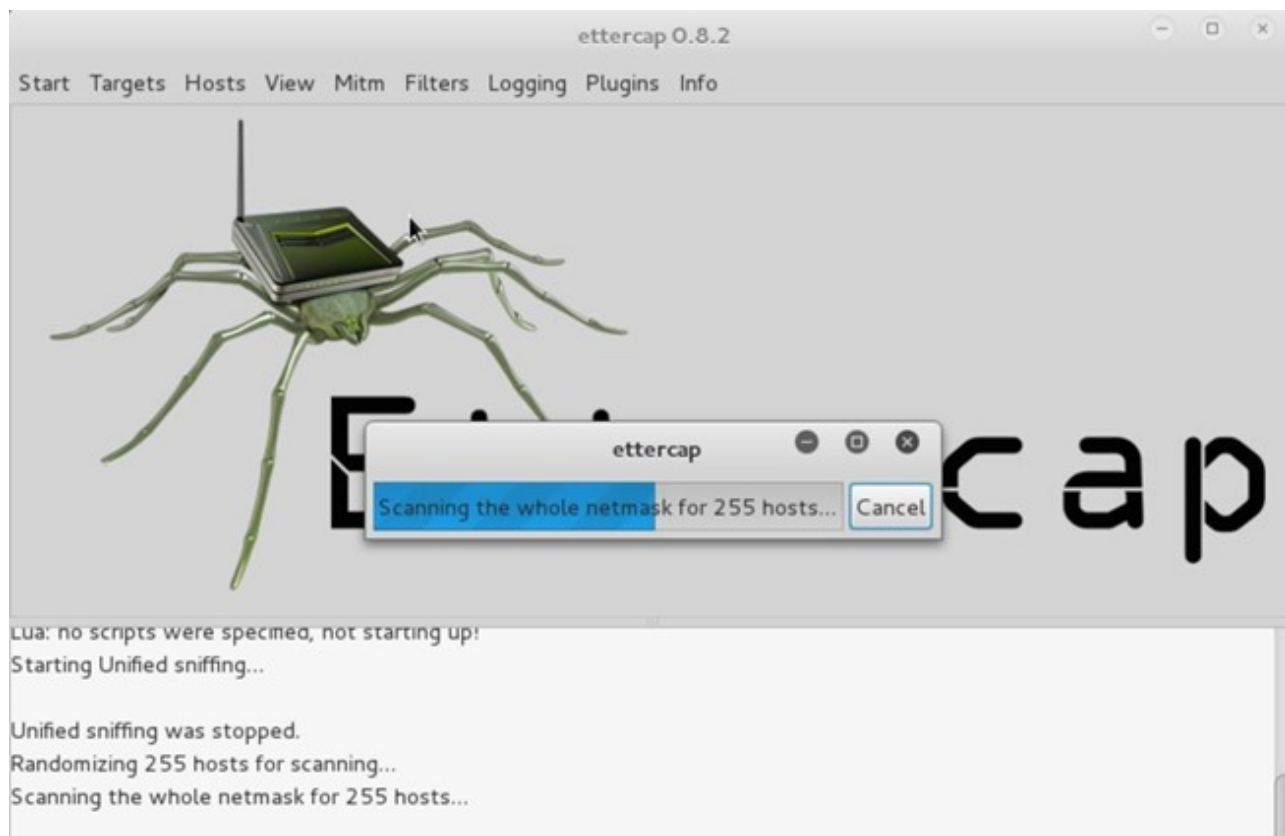
What we want next is to select our sniffing interface. Let's zoom through the steps real quick.

First select **Sniff > Unified sniffing...** > (Select the interface connected to the internet) > **OK**

(You can find out which interface is connected to the internet by typing in Terminal ifconfig and seeing which interface gives you an IP address).

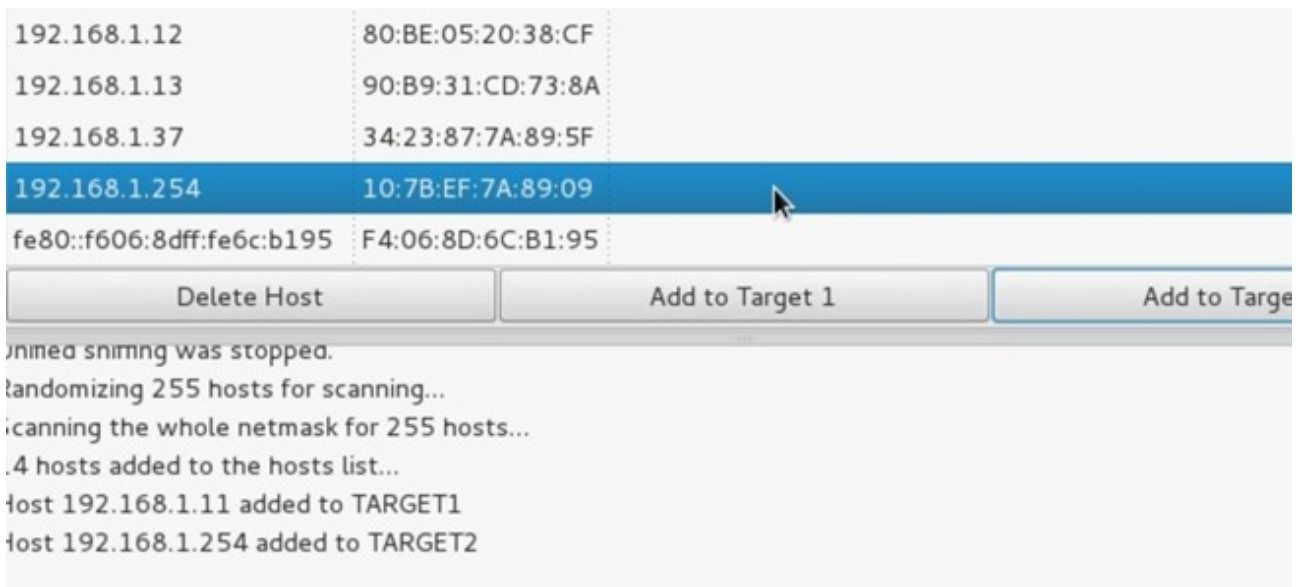
Then swiftly do **Start > Stop sniffing** because it automatically starts sniffing after we press **OK** and we don't want that.

Now we want to scan for targets on our network and pick one. To do this, go to **Hosts > Scan for hosts** and wait until it does the scan. It should only take a few seconds depending on the size of your network (which I assume isn't very large).



So we've dealt with the scanning but how do we see our targets? Well, go back to **Hosts** and select **Host list** to see all the targets that Ettercap has found.

Now what we want to do is add our victim machine to Target 1 and our network gateway to Target 2 but first we need to know both of their IP addresses. To find out our victim's IP address, we first need to know who we are attacking, and we can do so using nmap to find the information we need on the target machine. Once you are sure who your victim is, select their IP address from the host list in Ettercap and choose **Add to Target 1**. Now you need to find your gateway IP address (your router). To do this, open Terminal and type ifconfig and look at where it says **Bcast:** and that will tell you the IP address of your gateway. Now select that from the host list as well and choose **Add to Target 2**.



Step 4: Action

Now that we have both Targets set to our victim and gateway, we can proceed to the attack.

Go to the **MITM** tab and select **ARP poisoning**, choose **Sniff remote connections** and press **OK**. Now go to **Plugins > Manage the plugins** and double click **dns_spoof** to activate that plugin.

We now need to edit another file in the Ettercap folder.

```
root@Kali:~# gedit /etc/ettercap/etter.dns
```

This *etter.dns* file is the hosts file and is responsible for redirecting specific DNS requests. Basically, if the target enters **facebook.com** they will be redirected to Facebook's website, but this file can change all of that. This is where the magic happens, so let's edit it.

First, however, let me explain what can and should be done with the hosts file. So in a real life scenario, an attacker would use this opportunity to redirect traffic to their own machine for data sniffing. This is done by starting an Apache server on the Kali machine and changing the default homepage to a clone of, let's say facebook.com or chase.com so that when the victim visits those websites, after being redirected to the attacker

machine they will see the clones of the aforementioned sites. This will probably fool the unsuspecting user into entering their credentials where they really shouldn't. Enough talk, let's do it.

First, redirect traffic from any website you would like to your Kali machine. For that, go down to where it says "microsoft sucks ;)" and add another line just like that below it, but now use whatever website you would like. Also, don't forget to change the IP address to **your** IP address.

```
# or for TXT query (value must be wrapped in double quotes):
# google.com TXT "v=spf1 ip4:192.168.0.3/32 ~all"
#
# 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
#
microsoft.com      A 107.170.40.56
*.microsoft.com   A 107.170.40.56
www.microsoft.com PTR 107.170.40.56 # Wildcards in PTR are not allowed
facebook.com      A 192.168.1.39
*.facebook.com   A 192.168.1.39

#####
# no one out there can have our domains...
#
www.alor.org      A 127.0.0.1
www.naga.org      A 127.0.0.1
www.naga.org      AAAA 2001:db8::2

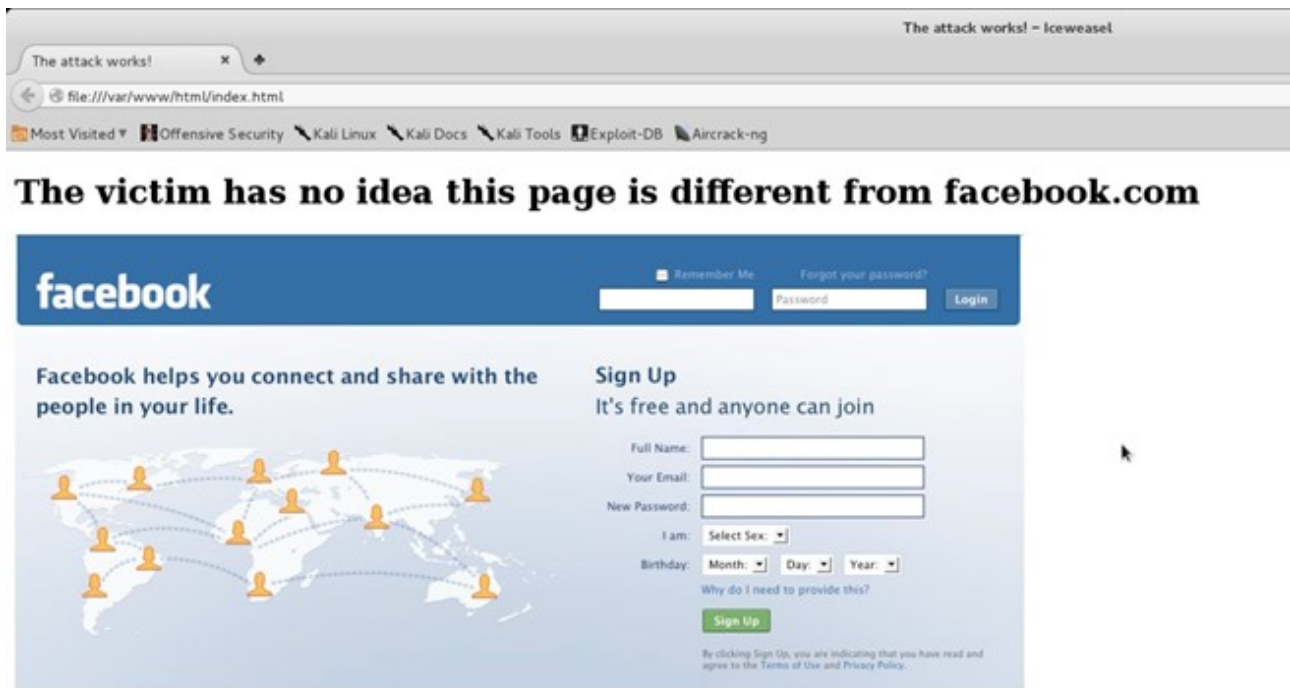
#####
# dual stack enabled hosts does not make life easy
# force them back to single stack
www.ietf.org      A 127.0.0.1
www.ietf.org      AAAA ::

www.example.org   A 0.0.0.0
www.example.org   AAAA ::1
```

Now we need to start Apache to accept incoming traffic.

```
root@Kali:~# service apache2 start
```

Let's head over to the default html page folder. That is where we can take control of what the victim sees when they get redirected. The location is `/var/www/html` where you will find the `index.html` page. You can alter the document to your needs and, once you think you have done sufficient fooling to your victim, you can save the page and changes will take effect instantly. Let's see here...



The victim has no idea this page is different from facebook.com

The final thing left to do here is to start the attack. Go back to Ettercap and select **Start** > **Start sniffing** and that should do it.

Now every time the victim visits the webpage you indicated in the **etter.dns** file (in my case it's facebook.com) they will be redirected to the fancy and inconspicuous page above. You can see how this can be extremely malicious, since the attacker could write a script that fetches the requested page immediately and sets up the **etter.dns** file and listens in on the login, all automatically. This should really alert you that it is really that simple to perform a DNS Spoofing attack with very few resources.

Detection

So how do you protect yourself from it? There are a couple of ways: using software built for ARP poisoning detection or checking the **arp** command manually on a regular basis (which is a pain).

Let's look at the software first, there are a few that I will mention.

1. XArp

A GUI advanced ARP spoofing detection and active probing software. It is designed for this kind of job and works on both Windows and Linux (configurable for OS X as well).

2. Snort

You most probably know Snort for its IDS amazingness, but I'm sure you haven't heard that it also detects ARP spoofing (you may have).

3. ArpON

This is a portable handler daemon for securing ARP against spoofing and cache poisoning.

This is a portable handler daemon for securing ARP against spoofing and cache poisoning.

There are a few others like Arpwatch, Antidote and ArpAlert but you could just Google them.

Now how about manual checking? Well, this one is a little tricky since it requires you know something beforehand. What you could do is remember the MAC address (or parts of it that will help you recognise it when you see it) of the default gateway (i.e. your router) and check if you can see it in the ARP cache.

To check the ARP cache, go to the Terminal and type **arp -a** and you will see several entries like this:

```
root@Kali:~# arp -a
? (192.168.1.11) at [redacted]:0e:30 [ether] on eth0
[redacted] (192.168.1.254) at [redacted]:89:09 [ether] on eth0
? (192.168.1.4) at [redacted]:36:59 [ether] on eth0
```

(I removed a few lines for 'security' reasons (not really))

If you can remember something like the first 6 characters of your gateway's MAC address and continually check `arp -a` to see if it matches, then you've got yourself a way to detect ARP poisoning without needing any 3rd party software. Isn't that great?

Conclusion

Now you know how DNS spoofing works and, most importantly, how to protect yourself from it. Being in a White Hat forum means not only learning attacks but also their remedies. This is particularly useful in real life scenarios and I hope that if you get yourself in this sort of heap you will know how to escape it.

I hope you enjoyed today's tutorial and hopefully you learned something from it. Any suggestions for future tutorials I will be happy to take in. Very soon I will be releasing something to the Null-Byte community so stay tuned.

As always have a great day, peace.

TRT