iGNITE Technologies

A DETAILED GUIDE ON SNIFFING

WWW.HACKINGARTICLES.IN

Contents

ARP Protocol

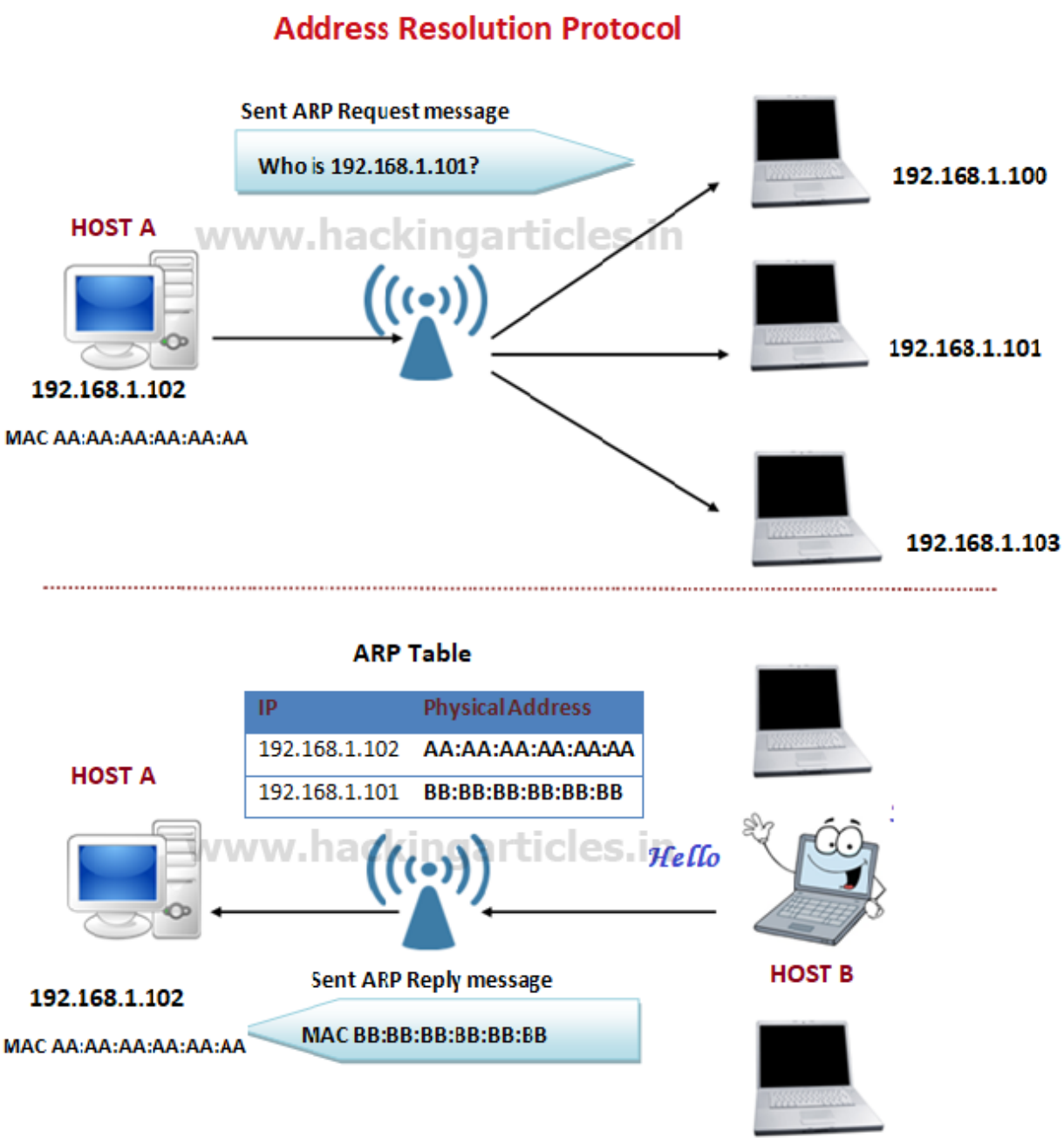
The Address Resolution Protocol (ARP) is a communication protocol. It is used for discovering the link layer address associated with a given Internet layer address, a critical function in the Internet protocol suite. ARP was defined by RFC 826 in 1982 and is Internet Standard STD 37. ARP is also the name of the program for manipulating these addresses in most operating systems.

ARP is used for mapping a network address (e.g. an IPv4 address) to a physical address like a MAC address. For more details visit **here.**

ARP Protocol Process

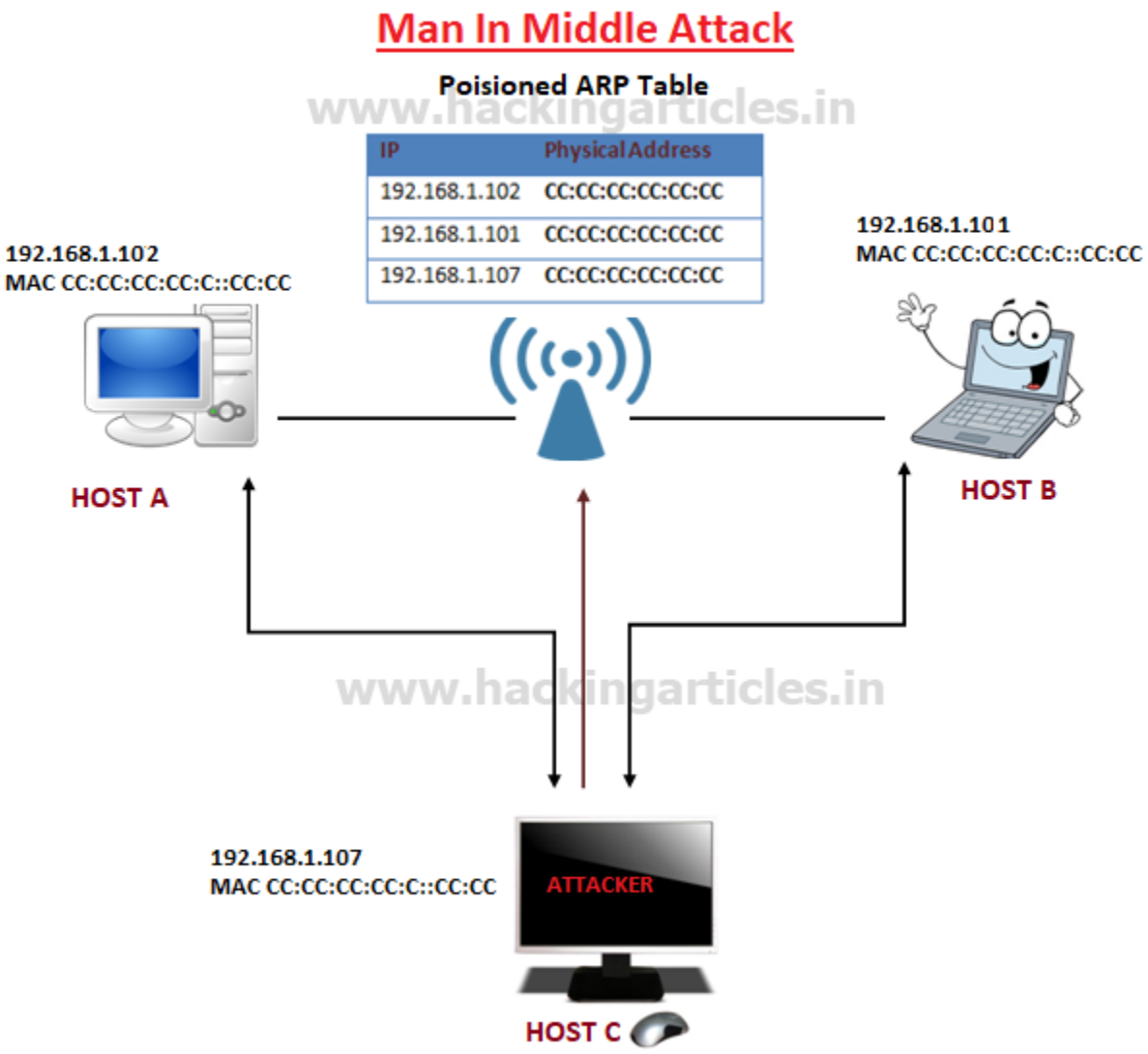
The Address Resolution Protocol is in many ways similar to a domain name service (DNS). An ARP resolves known IP addresses to unknown MAC addresses in the same way that DNS resolves known domain names to unknown IP addresses.As shown below in the given image,

If we observe the above image; IP address 192.168.1.102, wants to communicate with IP address 192.168.101, but does not know its physical (MAC) address. An ARP request is broadcast to all systems within that network, including IP X.X.X.100, X.X.X.101, and X.X.X.103. When IP address X.X.X.101 receives the message, it replies back via uni-cast with an ARP reply. This response contains the physical (MAC) address of BB-BB-BB-BB-BB-BB As shown above, this ARP reply information is then placed in the ARP cache and held there for a short duration, to reduce the amount of ARP traffic on the network. The ARP cache stores the IP, MAC, and a timer for each entry. The timer's duration may vary depending upon the operating system in use, i.e., the Windows operating system may store the ARP cache information for 2 minutes compared to a Linux machine, which may retain it for 15 minutes or so.



Let us now begin by exploiting the ARP protocol to our advantage!!!

Scenario: Consider the following scenario, in which we will use two Windows host machines to represent victims Host A and Host B, and Kali Linux Host C to target the victims. In the following image, you can see the attacker has conducted an arp poisoning attack, which has poisoned the arp table by adding the attacker's mac address to both the host's IP: A & B.



Let's Begin the ARP Poisoning Attack

The first step is to clear the ARP cache of both the hosts by typing the following command in the command prompt: **arp -d** for Host A, then ping Host A for the reply. Now type the command arp -a. This will show you the physical (MAC) address of the Host A machine.

```
arp -d
ping 192.168.0.101
arp -a
```

```
C:\Windows\system32>arp -d

C:\Windows\system32>ping 192.168.0.101

Pinging 192.168.0.101 with 32 bytes of data:
Reply from 192.168.0.101: bytes=32 time<1ms TTL=128
Reply from 192.168.0.101: bytes=32 time<1ms TTL=128
Reply from 192.168.0.101: bytes=32 time<1ms TTL=128
Reply from 192.168.0.101: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.0.101:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Windows\system32>arp -a

Interface: 192.168.0.102 --- 0xb
  Internet Address      Physical Address      Type
  192.168.0.101         00-0c-29-5b-4f-a1     dynamic
  224.0.0.252           01-00-5e-00-00-fc     static

C:\Windows\system32>
```

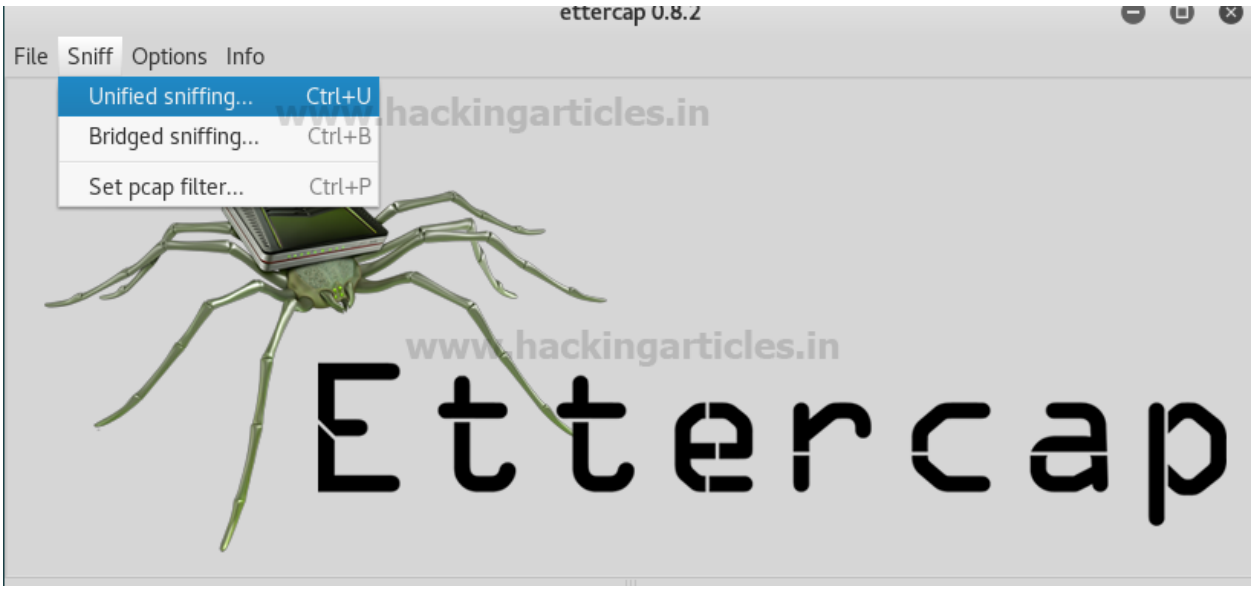
Similarly, let us do the same activity on the other systems, which is Host B.

```
C:\Windows\system32>arp -d

C:\Windows\system32>ping 192.168.0.102

Pinging 192.168.0.102 with 32 bytes of data:
Reply from 192.168.0.102: bytes=32 time<1ms TTL=128
Reply from 192.168.0.102: bytes=32 time<1ms TTL=128
Reply from 192.168.0.102: bytes=32 time<1ms TTL=128
Reply from 192.168.0.102: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.0.102:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Windows\system32>arp -a

Interface: 192.168.0.101 --- 0xb
  Internet Address      Physical Address      Type
  192.168.0.1           84-16-f9-47-df-7a     dynamic
  192.168.0.102         00-0c-29-19-c2-8b     dynamic

C:\Windows\system32>_
```

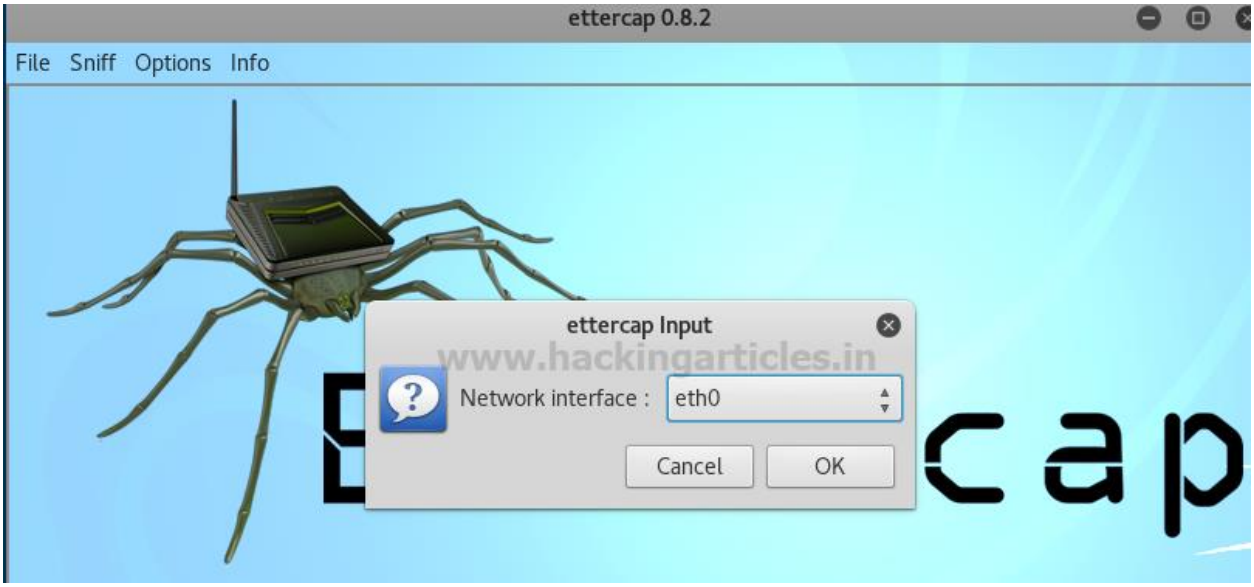
Start Sniffing with Ettercap

Let us now start to exploit both Host A and Host B. From the Host C machine, which is our Kali Linux, start sniffing with the Ettercap tool as shown in the below image on Kali.

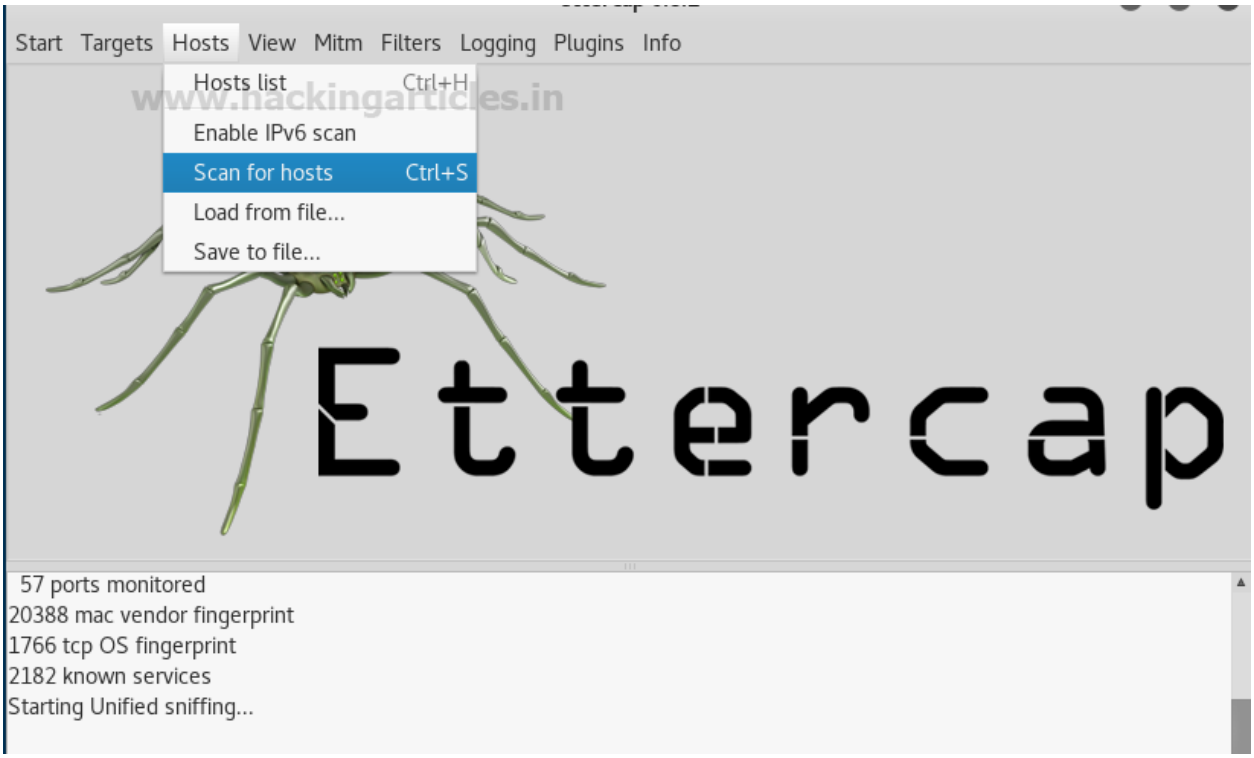
Go to Sniff and select Unified sniffing.



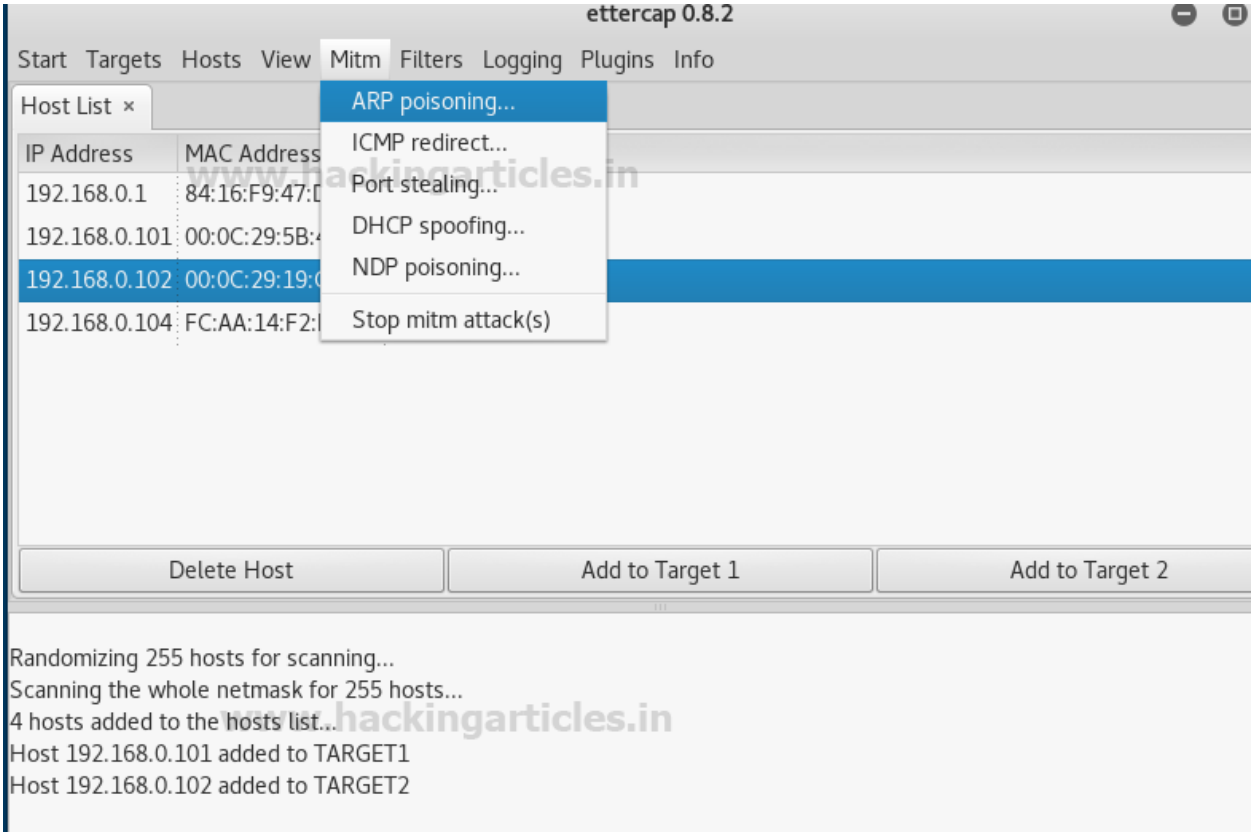
Select the network interface as appropriate. In this case, it is eth0. Click on **OK.**



Now go to the Hosts Tab and select **Scan for Hosts** as shown below to scan the connected system on a local network.

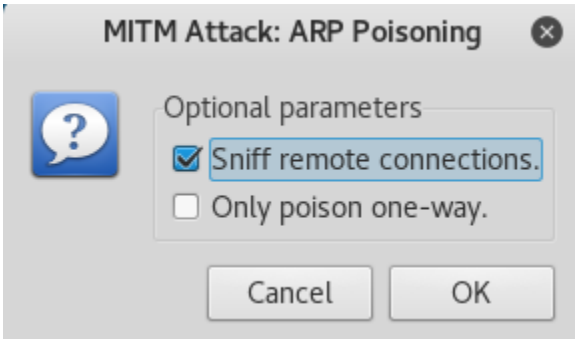


You will get the host list of all the scan hosts as shown below. Let us now select our targets from the host list X.X.X.101 and X.X.X.102. Add both the targets one by one by clicking on the tab Add to Target 1 and 2 respectively. From the given image, we can see that both the targets are now added to our list.

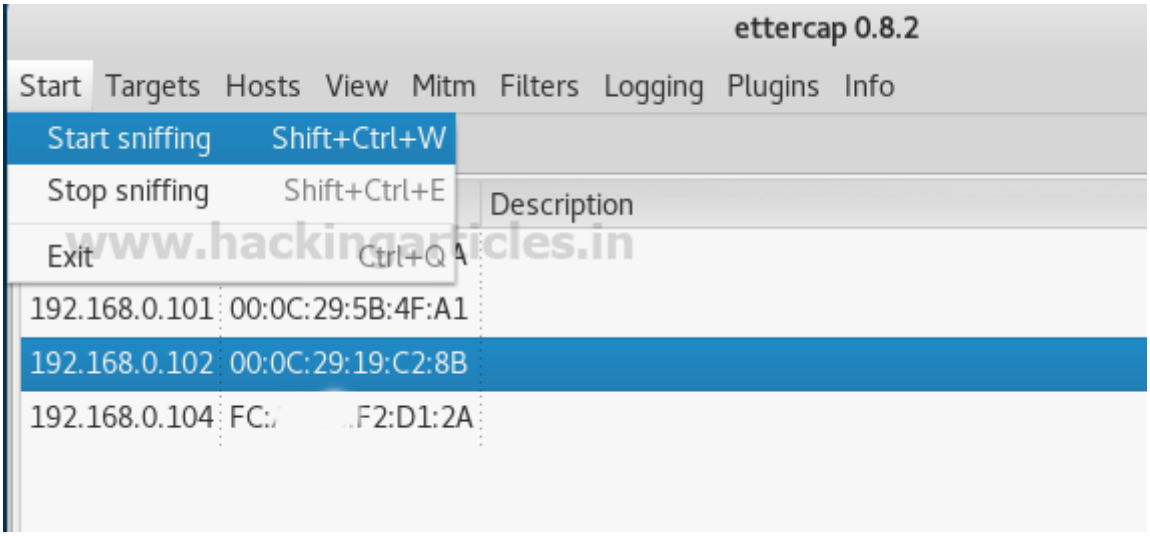


Now go to MitM (the man in the middle) and select ARP Poisoning. A dialogue box will appear for optional parameters.

Check the box "Sniff remote connection" and click OK.



Go to the start tab and click on "Start Sniffing" to target the hosts A and B added.



Let us now go to our Kali machine and open the terminal. Let us now type the command **ifconfig** to determine our IP address and physical (MAC) address. In our case, it is **00:0c:29:5b:8e:18** as highlighted in the given image.

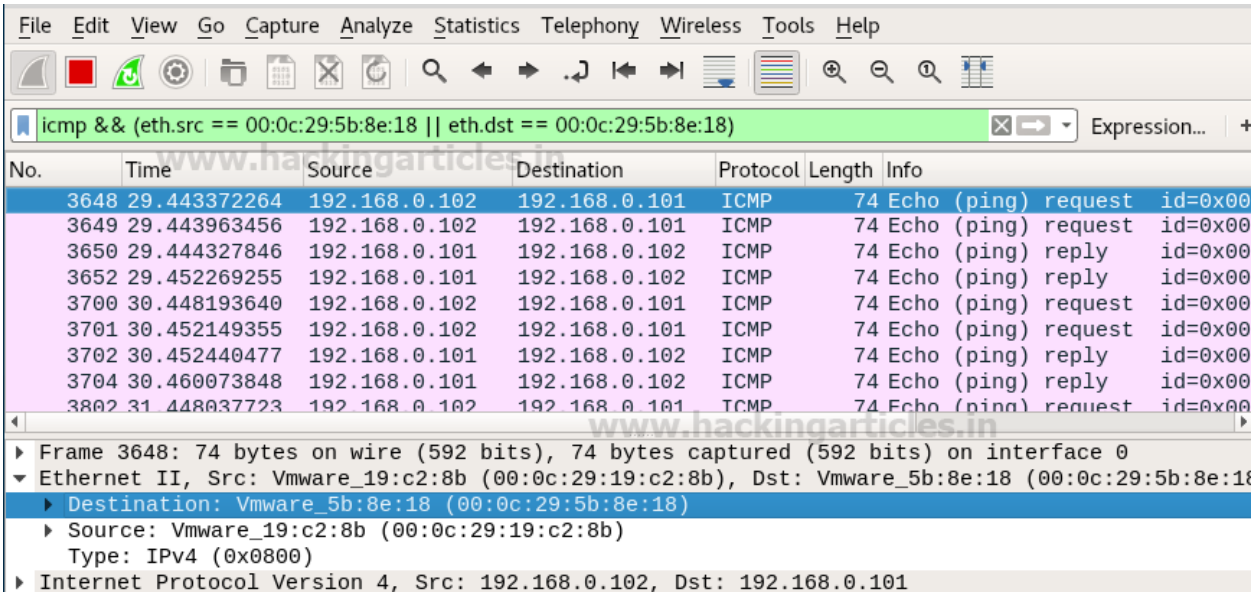
```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.107  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::20c:29ff:fe5b:8e18  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:5b:8e:18  txqueuelen 1000  (Ethernet)
        RX packets 18938  bytes 5853523 (5.5 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 668  bytes 47347 (46.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

As we have started the arp poisoning attack on both the victim machines X.X.X.101 and 102 from our Kali machine, if we go to any host and type arp -a on the command prompt, you will clearly see that the physical (MAC) address of the victim machine has changed to the physical (MAC) address of the Kali machine. As shown above, the physical (MAC) addresses of both the IP X.X.X.102 and X.X.X.107 are the same, which means that all the traffic from host X.X.X.102 is passing through Kali machine X.X.X.107

```
C:\Windows\system32>arp -a

Interface: 192.168.0.101 --- 0xb
  Internet Address      Physical Address      Type
  192.168.0.1           84-16-f9-47-df-7a     dynamic
  192.168.0.102         00-0c-29-5b-8e-18     dynamic
  192.168.0.107         00-0c-29-5b-8e-18     dynamic
  192.168.0.255         ff-ff-ff-ff-ff-ff     static

C:\Windows\system32>
```

Demonstrate MITM with Wireshark

Let us now open Wireshark on our Kali machine and analyse the packets. Let us filter the packets by typing the following command **ICMP && (eth.sec = = 00:0c:29:5b:8e:18 || eth.dst == 00:0c:29:5b:8e:18)**, here in the command eth.sec means (Ethernet source) and eth.dst means (Ethernet destination), the MAC address are common in both source and destination which is the physical MAC address of our Kali machine, what we see is the source IP X.X.X.102 and destination X.X.X.101 are getting captured by the Kali machine which has a Physical (MAC) address **00:0c:29:5b:8e:18**, hence proving successful sniffing of the victim machine.



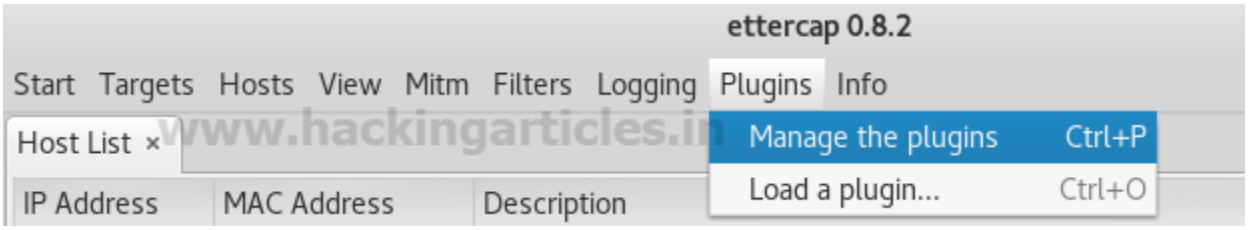
Combining DNS Spoofing with sniffing

Let us now exploit both of our victim machines with a DNS Spoofing attack.

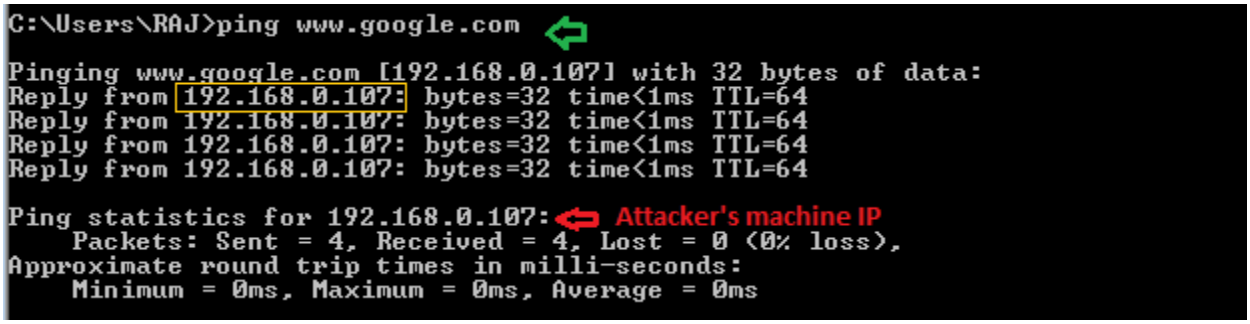
From your Kali machine, go to the path: **/root/etc/ettercap/etter.dns,** open the file and remove any content if available, then type the value *** A (your Kali Linux IP address)** as shown below, and save the file.

The next step is to go to the ettercap tool and select plugins, then click on Manage the Plugins as shown below:

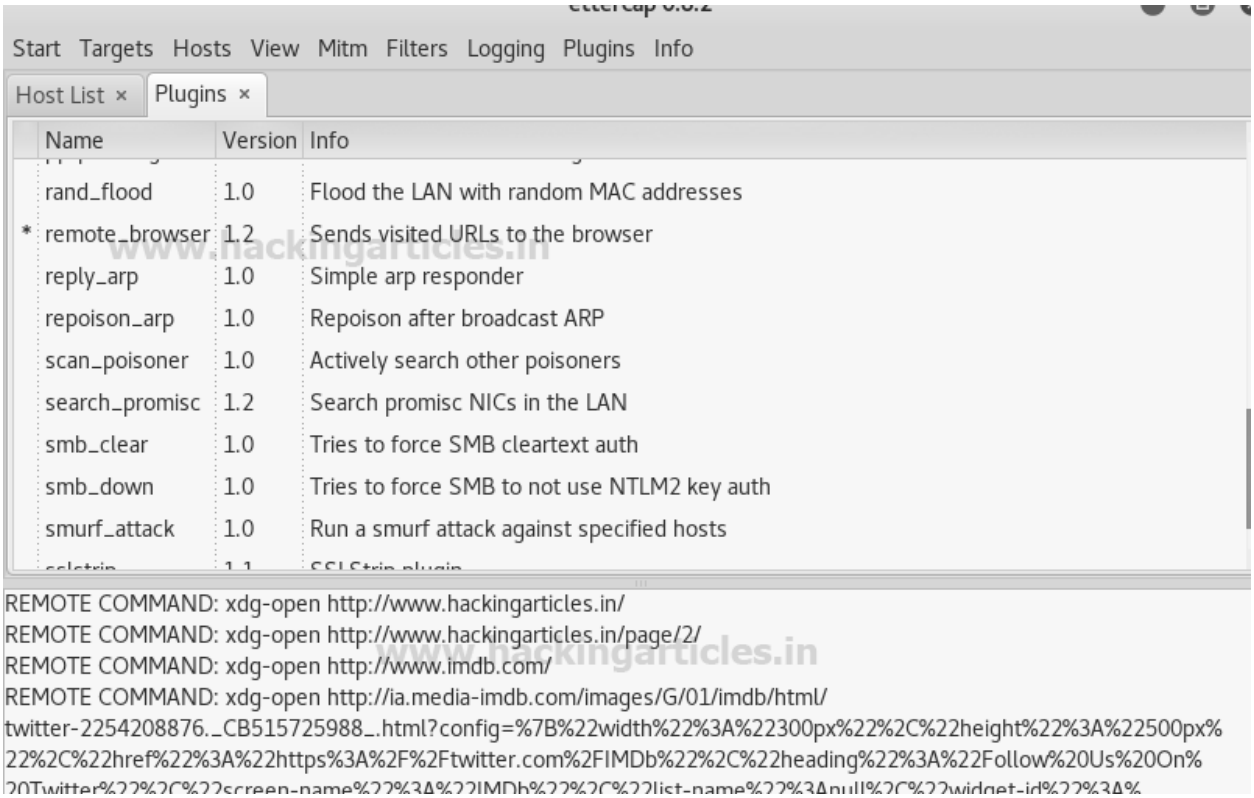
Now select the dns_spoof plug-in. Once selected, you will see a (*) sign on the said plug-in.



Now if from the victim machine we type the command **ping www.google.com**, you will observe that the reply is getting received from IP X.X.X.107 which is the IP for our Kali machine, which means that the Kali machine has become the DNS server for the victim machine.



Let us now add one more plug-in the same way we added the dns_spoofing plug-in. This time we will use the remote browser plug-in as shown in the image below. Once this plug-in gets added, you can capture all the browser activity performed by the victim on his browser, including user names and passwords.



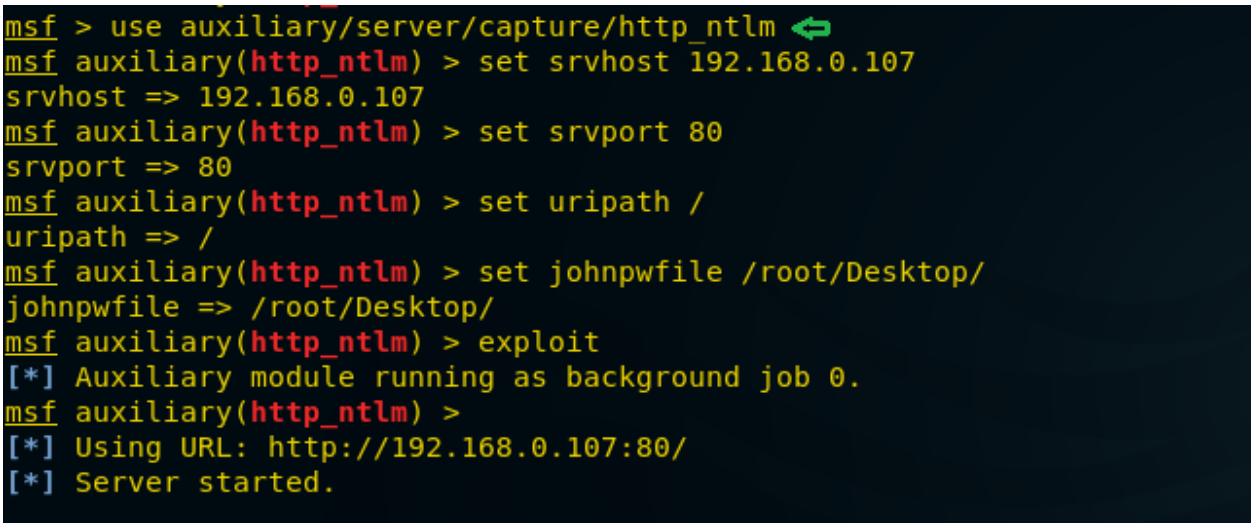
Capturing NTLM passwords

Open Kali terminal and type msfconsole. once the console starts to type: search http_ntlm, now type: use auxiliary/server/capture/http_ntlm as shown in the below image:

This module attempts to quietly catch NTLM/LM Challenge hashes.

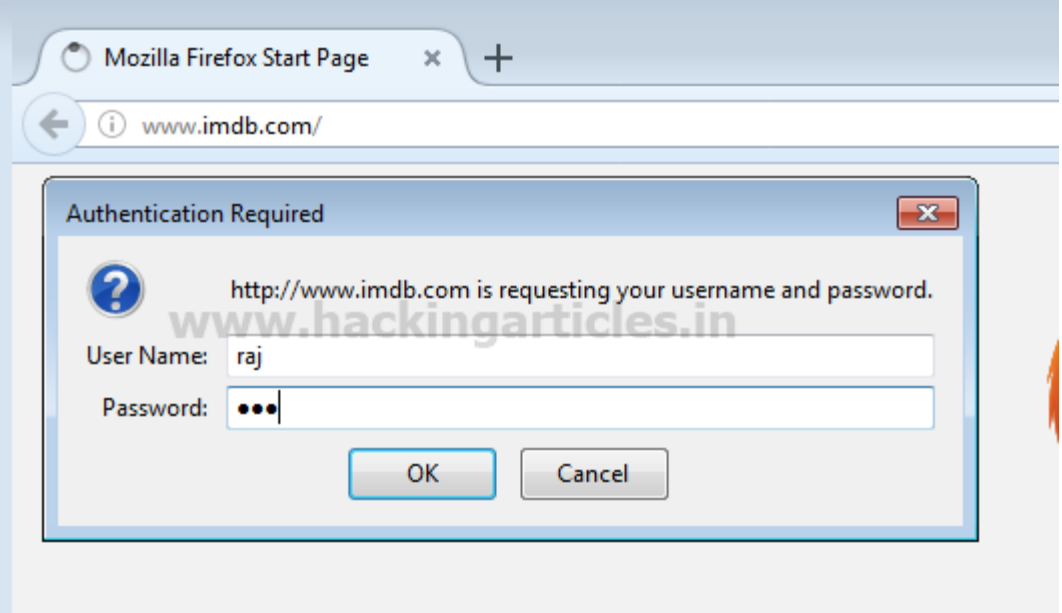
```
use auxiliary/server/capture/http_ntlm
set srvhost 192.168.0.107
set srvport 80
set uripath /
set johnpwfile /root/Desktop/
exploit
```

Now, according to the above trap set for the victim, this module will capture the NTLM password of the victim's system when he opens any http web site in his browser, which will redirect that web site to the attacker's IP.



The victim is attempting to browse "IMDb.com" on his web browser, but it requires authentication, which is requesting his username and password, as shown in the image below.Now if he tries to open something else, let's say google.com, it will also ask for a username and password for authentication. Until the victim submits his username and password, he cannot browse anything on his web browser.

As the victim enters username and password, the attacker in the background will capture the NTLM hash on his system.



Great!! The attacker had captured NTMLv2 hash; now let count detail apart from the hash value that the attacker has captured.

From the given image you can see that the attacker has captured two things more:

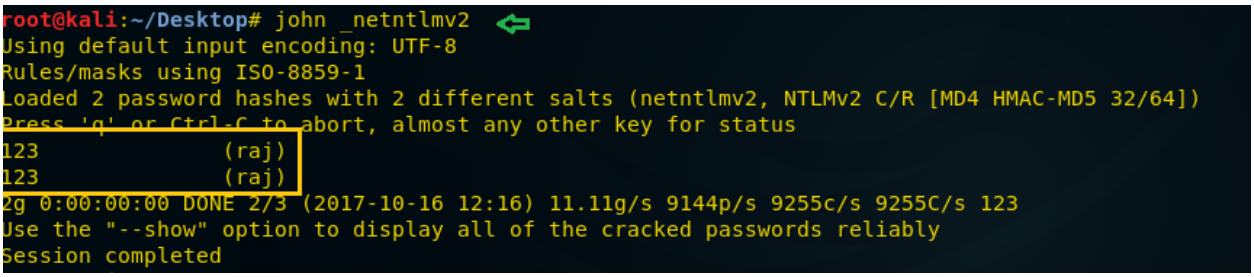
- **Username:** raj
- **Machine name:** WIN-1GKSSJ7D2AE

```
[*] 2017-10-16 12:14:23 -0400
NTLMv2 Response Captured from WIN-1GKSSJ7D2AE
DOMAIN:  USER: raj
LMHASH:Disabled LM_CLIENT_CHALLENGE:Disabled
NTHASH:200fb6bc22b5bae591348f5312520460 NT_CLIENT_CHALLENGE:0101000000000006ced
9dcd9946d301a082798b9bf7c9590000000002000c0044004f004d00410049004e00000000000000
0000

[*] 2017-10-16 12:14:24 -0400
NTLMv2 Response Captured from WIN-1GKSSJ7D2AE
DOMAIN:  USER: raj
LMHASH:Disabled LM_CLIENT_CHALLENGE:Disabled
NTHASH:735fa82277b2a44a8ff7e10616c9a7de NT_CLIENT_CHALLENGE:0101000000000005d21
2fce9946d30122e025ed4ccc1c150000000002000c0044004f004d00410049004e00000000000000
0000
```

Now use John the Ripper to crack the ntlmv2 hash by executing the given below command.

john _netntlmv2

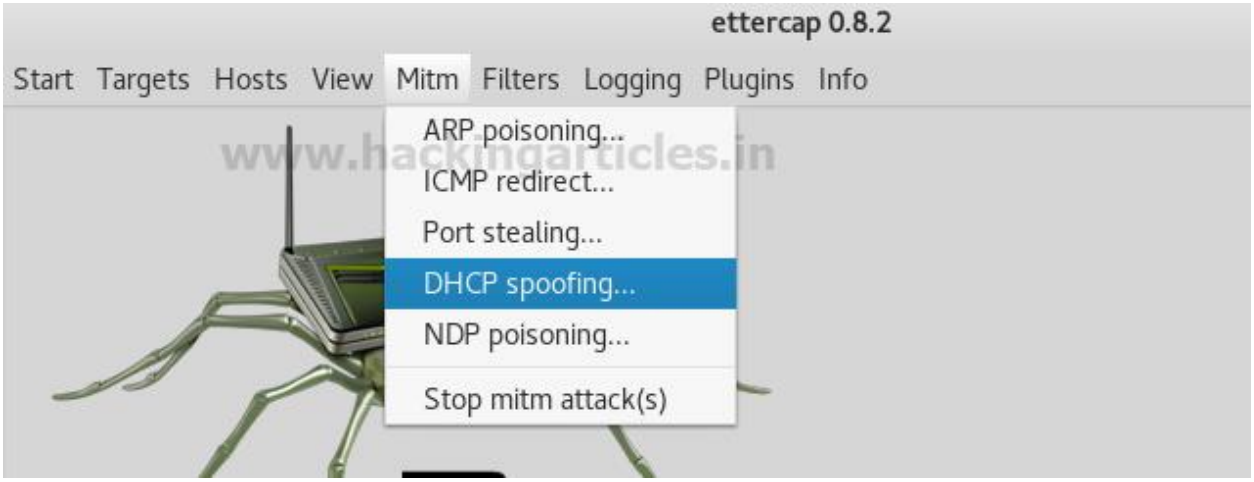
From the given below image, you can confirm that we have successfully decoded the captured hashes with the user name as **raj** and password as **123.**



Combining DHCP Spoofing with sniffing

DHCP spoofing: a fake DHCP server is set up by an attacker on a local network to broadcast a large number Request message of false IP configurations to genuine clients.

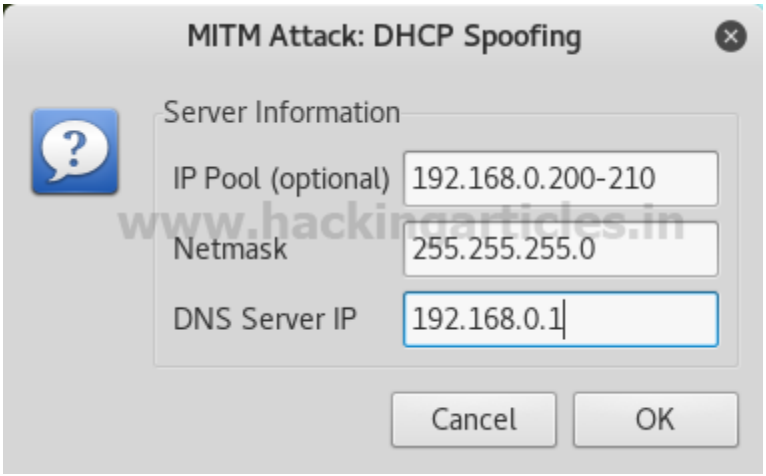
Go to ettercap and click on MitM. Select DHCP spoofing.



Form the below image, provide the necessary information

- IP Pool – **168.0.200-210** (put an IP range to issue IP to the system connected to the network, this will work as DHCP server)
- Net-mask **255.255.0** (as per the IP Class)
- DNS Server IP **168.0.1** (as per the IP Class)

Click OK and Start sniffing.

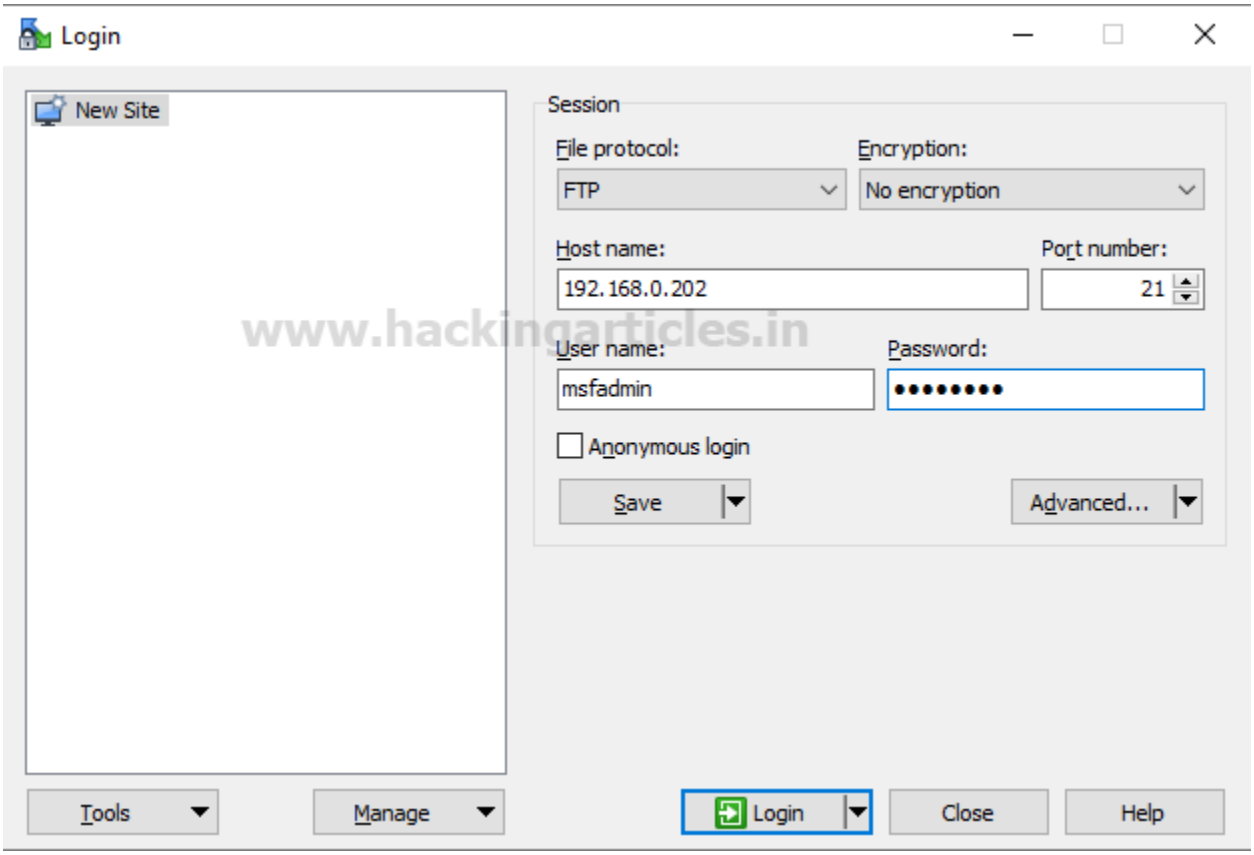


I've enabled the "metasploitable server" here, and the image below shows the **IP 192.168.0.202**, which is from the pool of IP ranges we provided on ettercap DHCP.

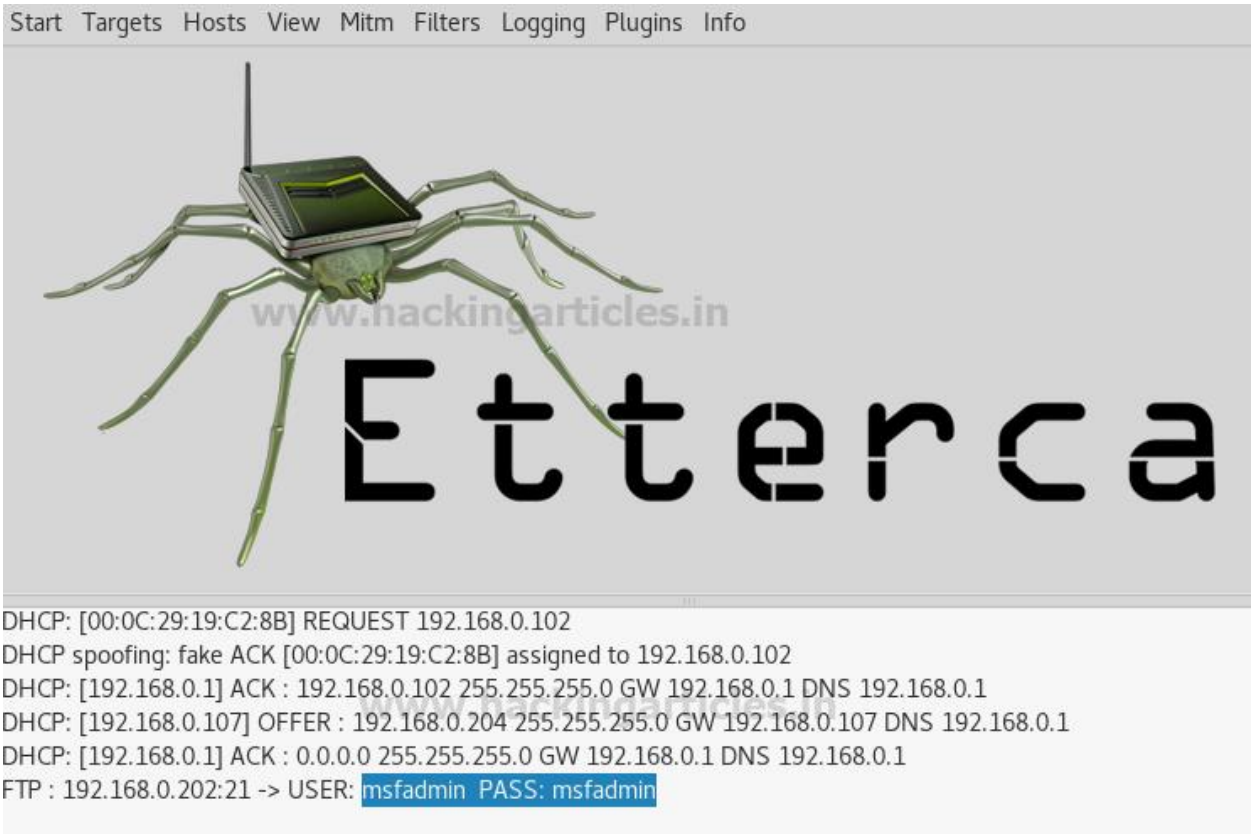
```
No mail.
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:ce:2b:57
          inet addr:192.168.0.202  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fece:2b57/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:56 errors:0 dropped:0 overruns:0 frame:0
          TX packets:70 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:9810 (9.5 KB)  TX bytes:7355 (7.1 KB)
          Interrupt:19 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:92 errors:0 dropped:0 overruns:0 frame:0
          TX packets:92 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19393 (18.9 KB)  TX bytes:19393 (18.9 KB)

msfadmin@metasploitable:~$
```

Let us now go to the client machine and try to connect the metasploitable server with an **FTP (File Transfer Protocol)** client as shown in the below image.

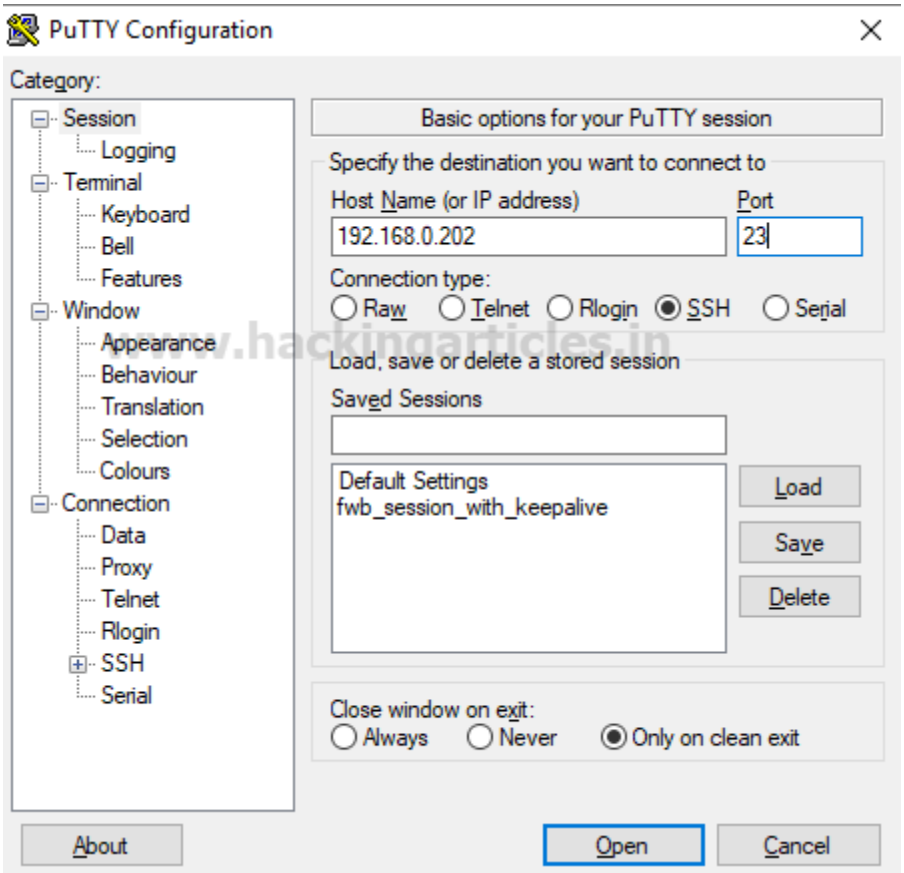
Provide the hostname (IP), user name, and password to connect to the FTP server.



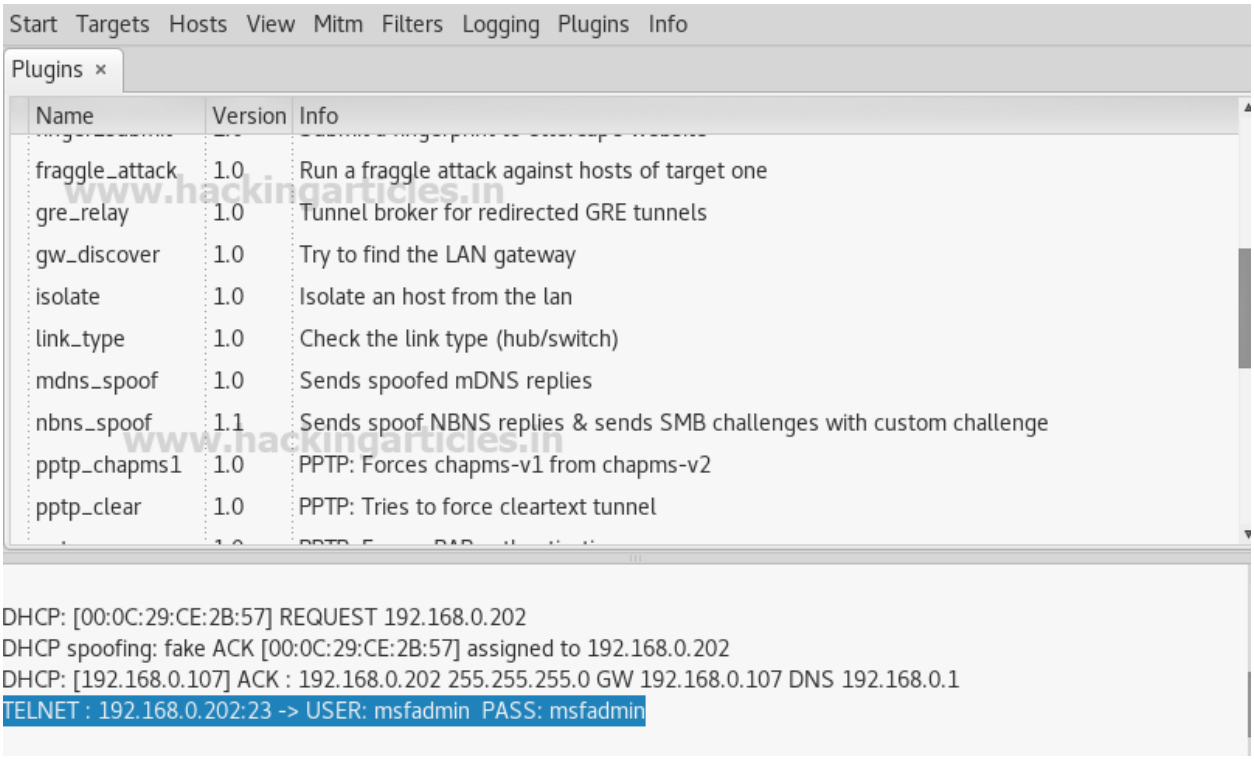
From the given below image, we can see that, the information such as username and password for FTP is getting captured by the ettercap provided by the host machine. In our case, it is **User: msfadmin, PASS: msfadmin.**



From the given below image, you can perceive that now we are trying to connect with the metasploitable server (192.168.0.202) through telnet via port 23 using putty. It will prompt you for the user name and password, and provide the necessary information.



From the above image, we can clearly see that ettercap has captured the credential information provided by the user. In our case, it is **User: msfadmin**. **Pass: msfadmin** for telnet service.

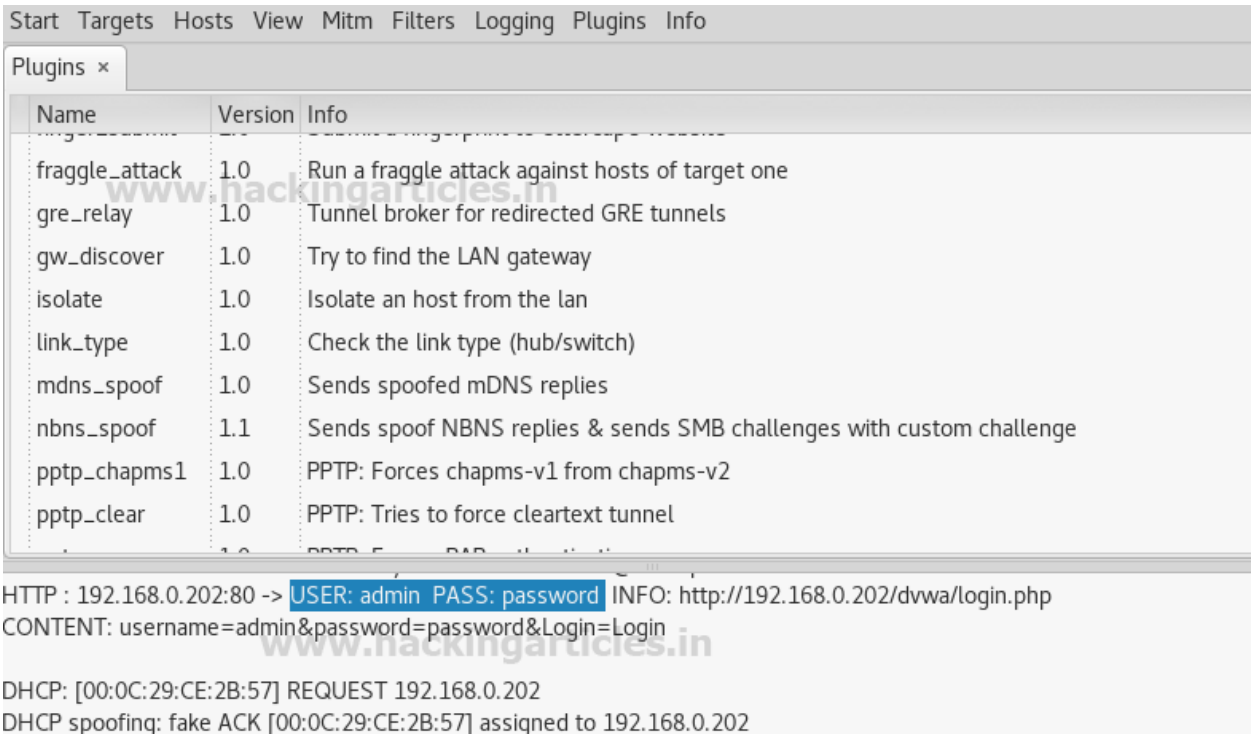


HTTP Password Sniffing

Let us now do the same through **HTTP (Hypertext Transfer Protocol)**

From the below image, we can see DVWA service is running in our metasploitable server, through the client browser let us type **192.168.0.202/dvwa/login.php**, it will prompt for **username** and **password**, let's provide the credentials.

From the below image, we can see that ettercap has once again captured the username and password provided by the user from the browser. In our case, it is username: **admin** and PASS: **password f**or HTTP service.



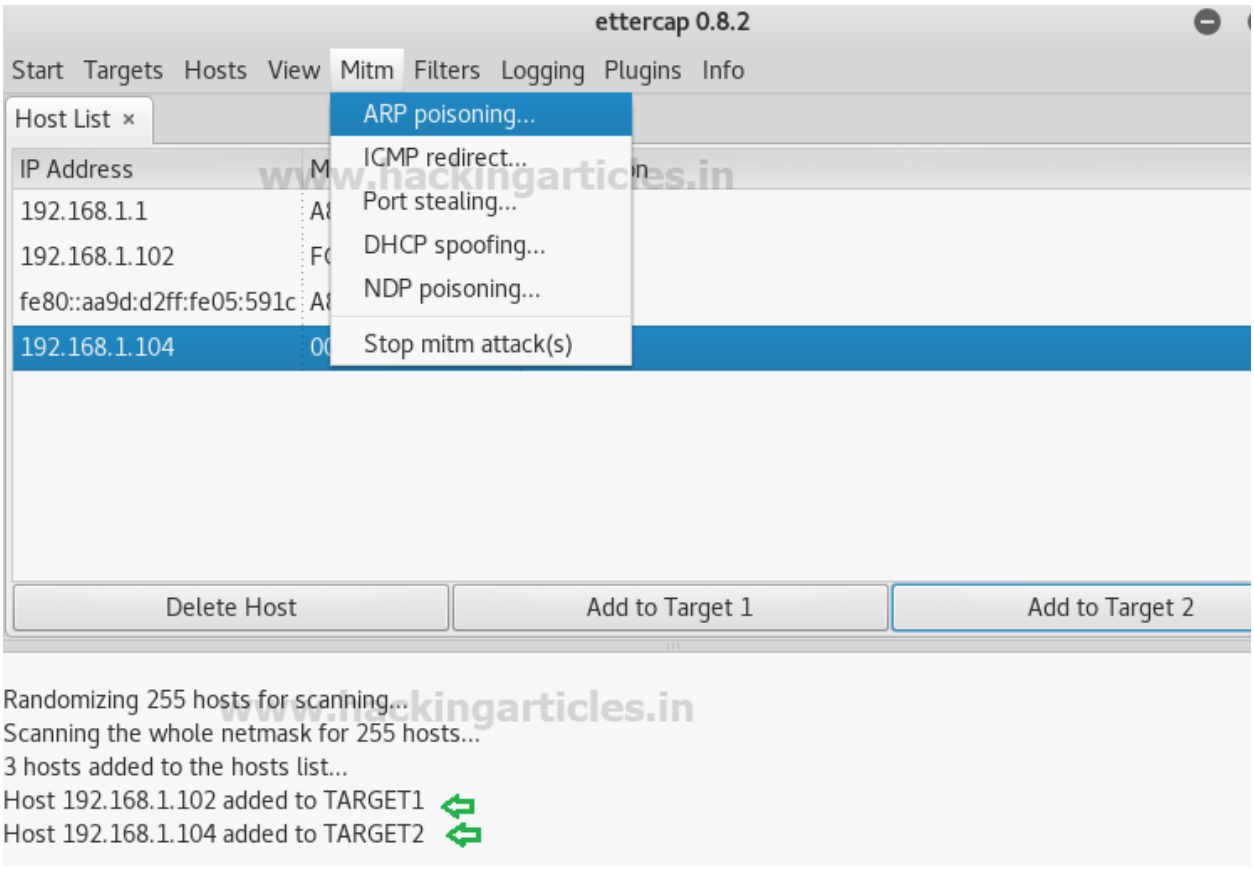
SMTP Password Sniffing

Lastly, let us now try this with SMTP (Simple Mail Transport Protocol) sniffing.

The first step is to configure an SMTP server in your environment. Please click **here** to see how we can configure an SMTP server on a Windows machine.

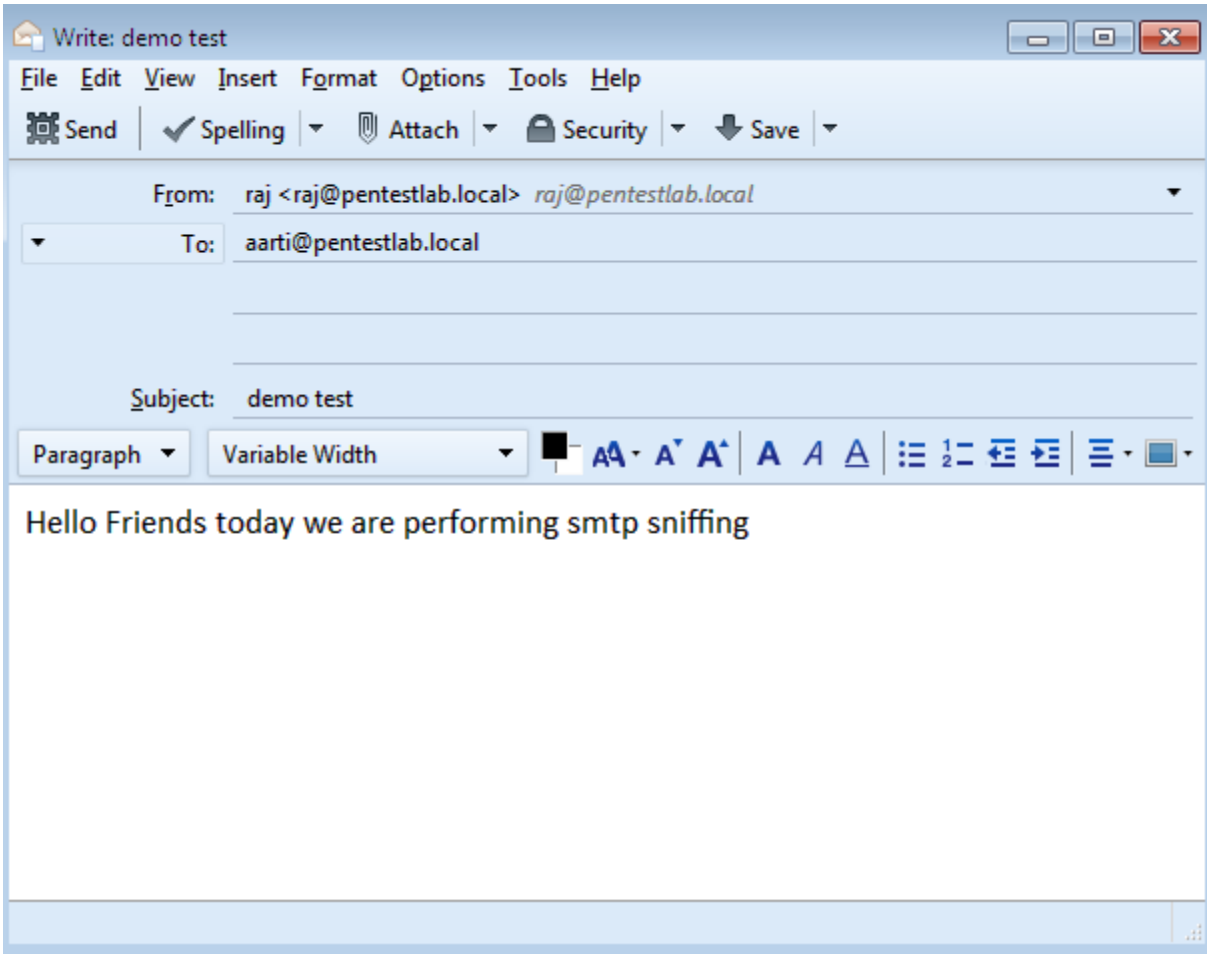
Once the server is configured, and we have set up email clients on the target machines,

let us open Ettercap and add both our targets X.X.X.102 and X.X.X.104 and select ARP poisoning.

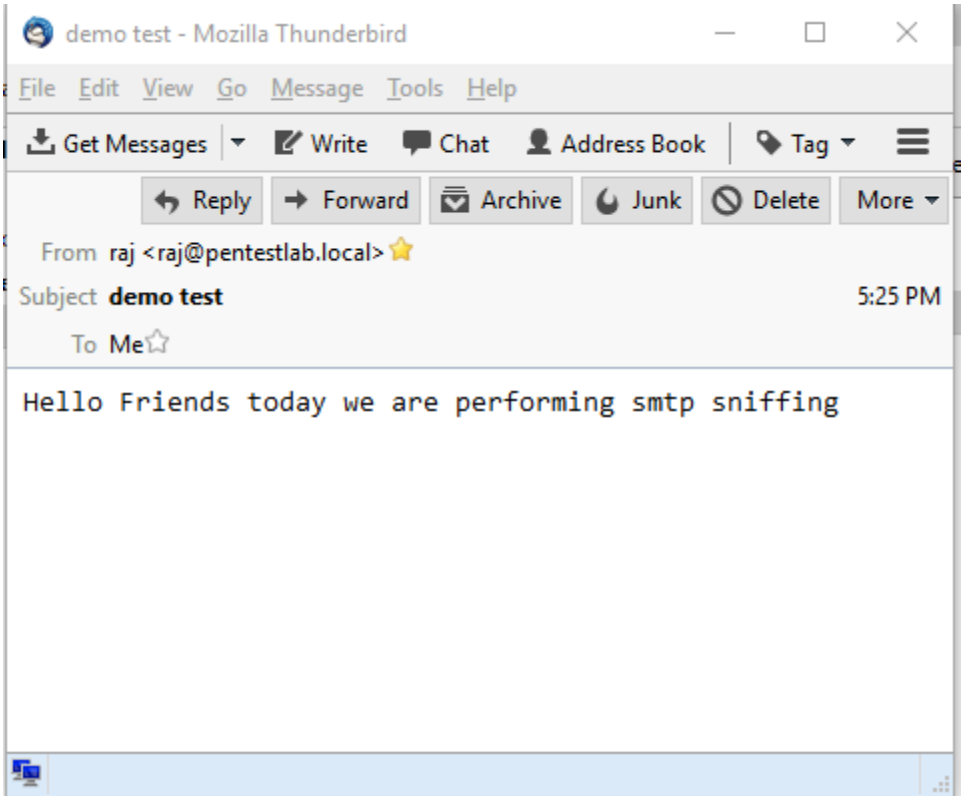


Now let us send an email from Target A to Target B as shown below.

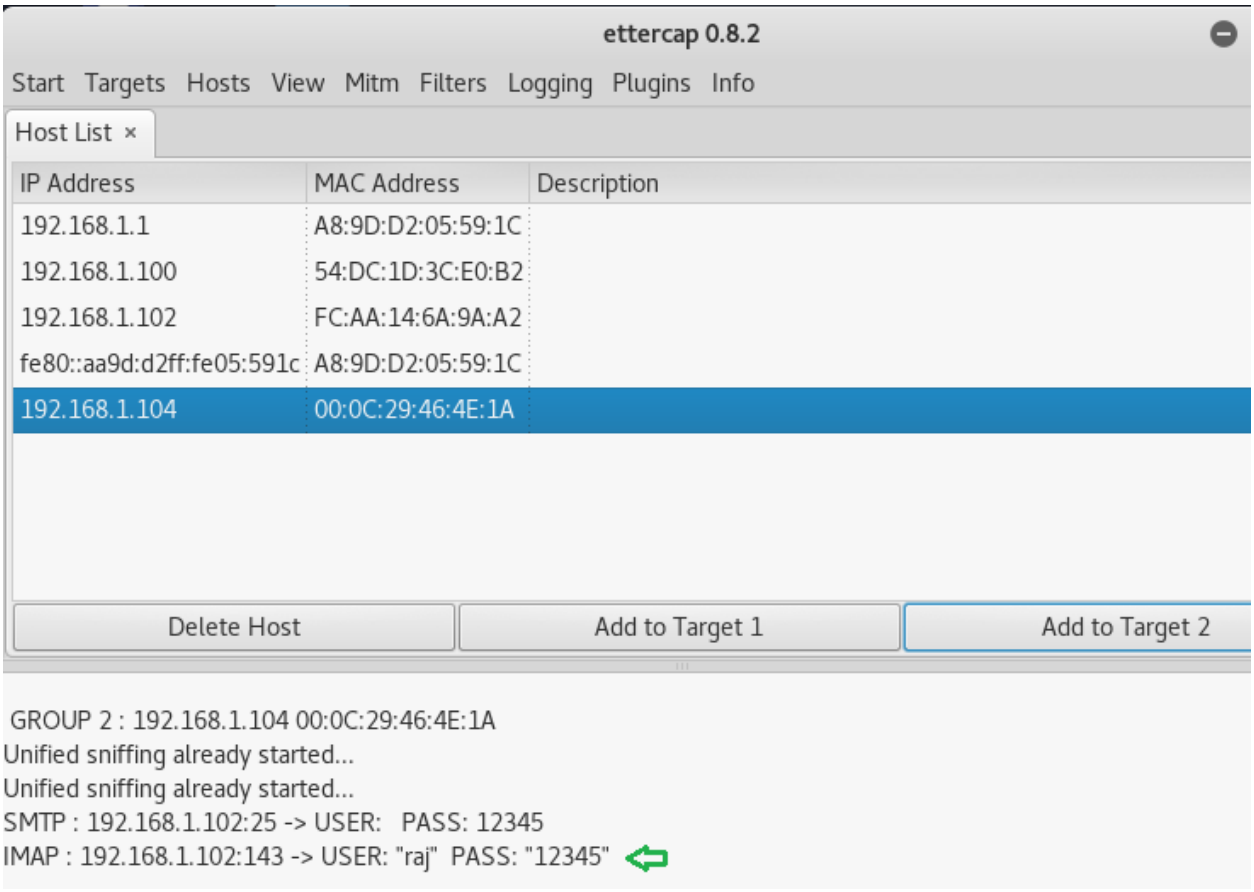
Here, target A: **raj@pentestlab.local** is sender who is the sending the message to target B: **aarti@pentestlab.local** and hence port 25 for SMTP service will get into action.



The given below image has confirmed that Aarti has received raj's mail successfully, while in the background, the attacker is sniffing all the traffic passing through the router.

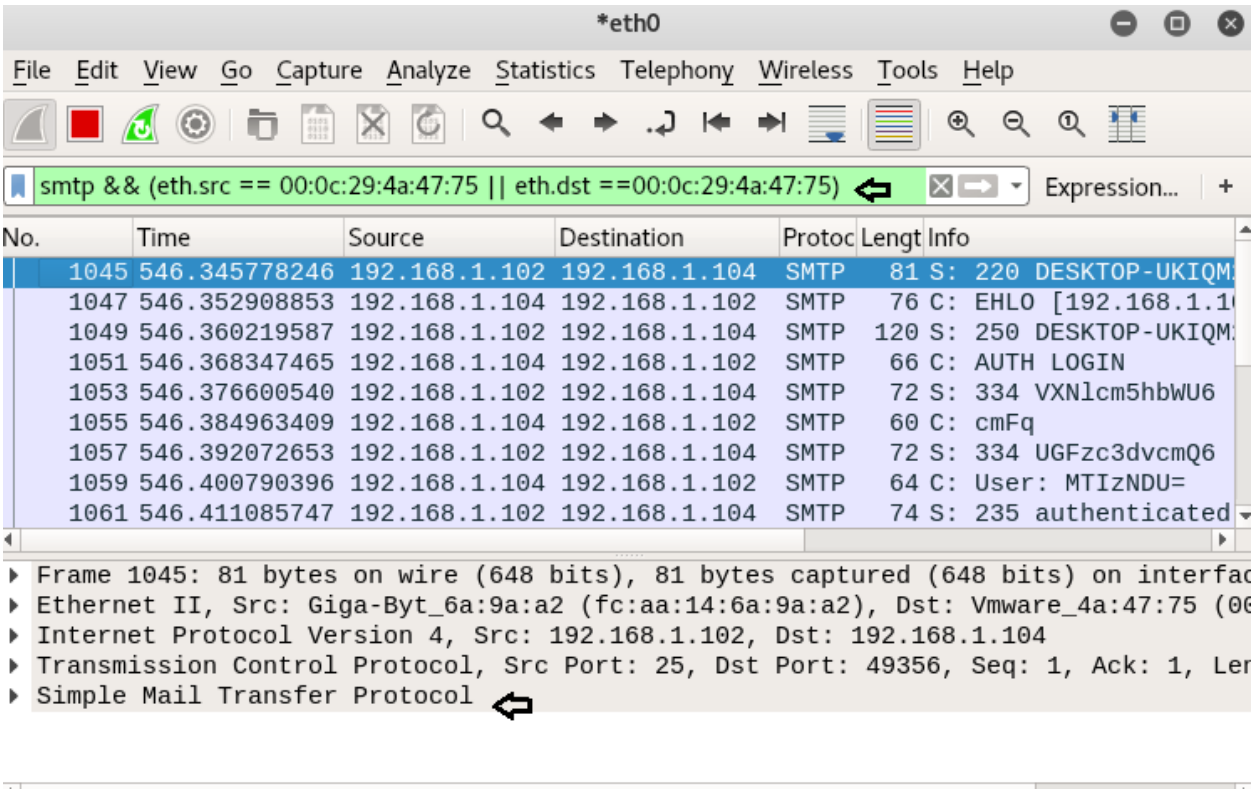


If we now go to the Ettercap console, we can clearly see that it has successfully sniffed the traffic between Target A and Target B and captured the credential of Target A (Raj) as shown in the above image.



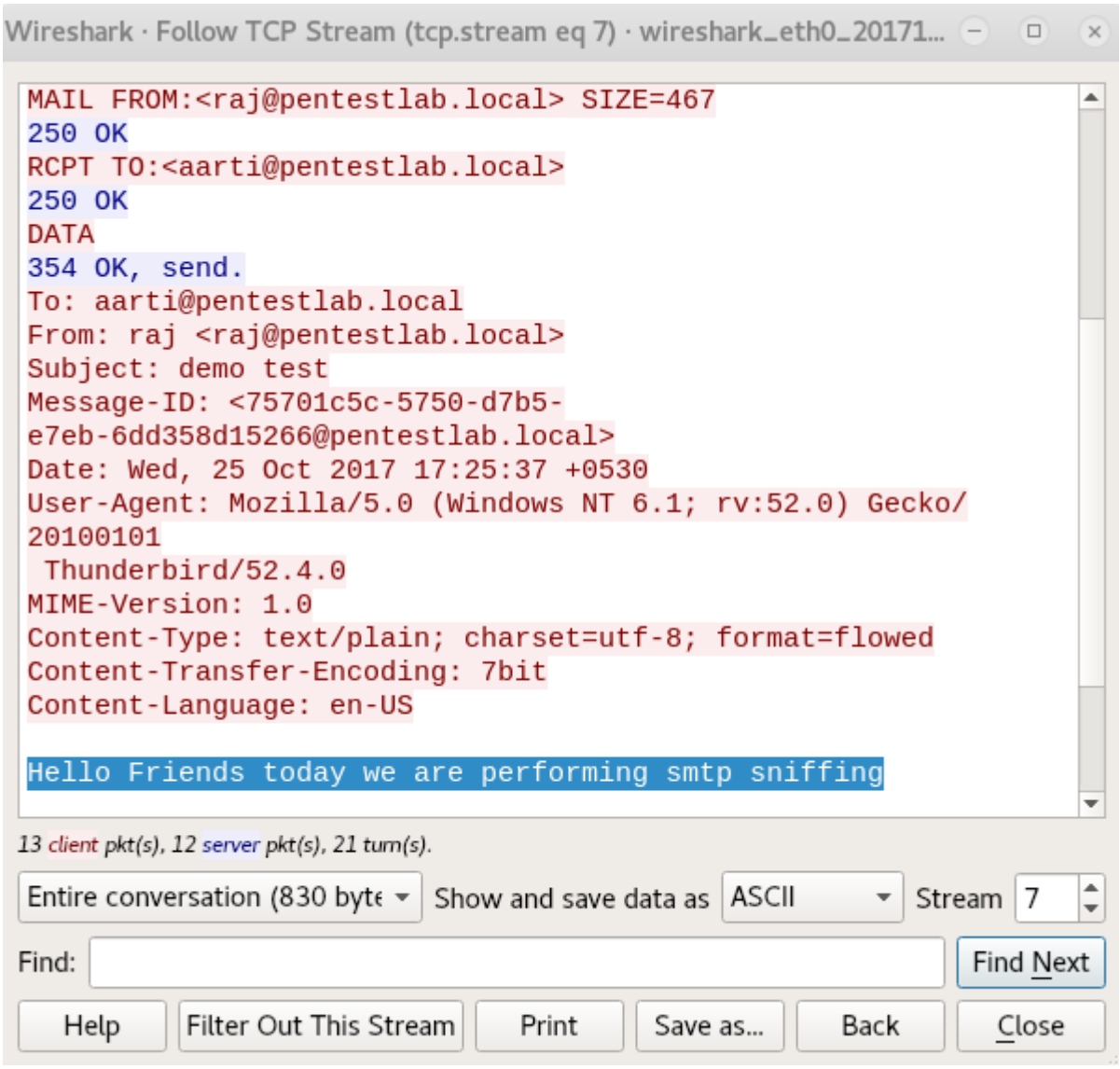
Capture Email of SMTP server with Wireshark

Go to wire shark and put the filter **smtp && (eth.src == 00:0c:29:4a:47:75 || eth.dst == 00:0c:29:4a:47:75)** the MAC address filter is for our Kali machine. You will observe it has captured packets from both our target machines.



It has sniffed every SMTP packet, captured both email IDs, i.e., sender and receiver, with the message being sent to Target B, which is Hello friends, today we are performing SMTP sniffing, which shows that we have been successful in our attack on the selected targets, as shown in the image below.

Throughout this article, we discussed ways and techniques that can be used to exploit the Arp protocol successfully. Let us now briefly discuss the technique that can be used to detect the arp attack.

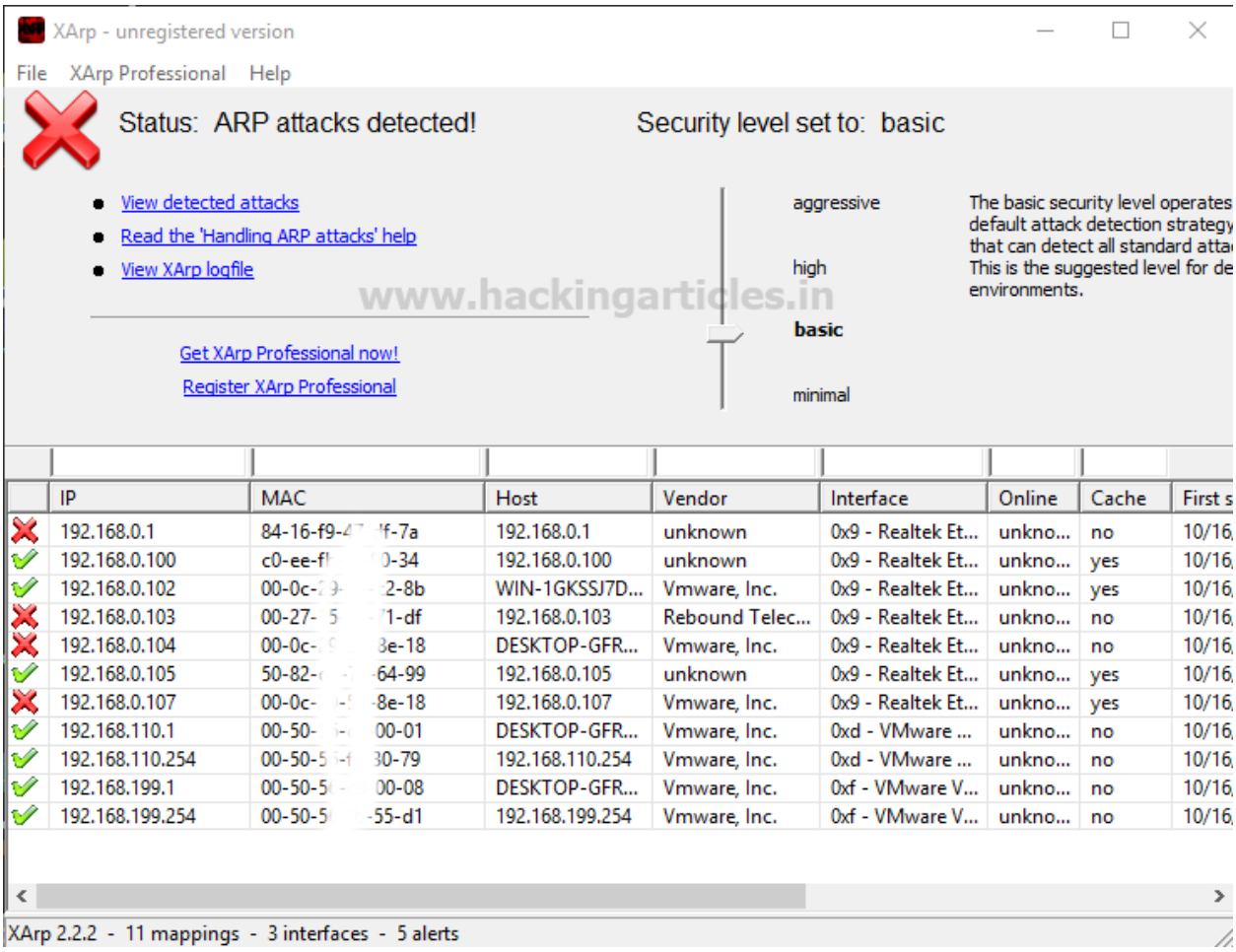


ARP Attack Detection

There are various tools available to detect the arp attack. One of the most common tools is the XArp tool, which we will be using for this article.

We can run this tool on any host machine in the network to detect the arp attack. The above image shows the affected systems on the network highlighted in red (X). We can disconnect these hosts from the network and decide upon the next course of action to mitigate this risk by implementing the following controls:

1. **Dynamic address inspection**
2. **DHCP snooping**
3. **VLAN hopping prevention**


XArp - unregistered version
File XArp Professional Help
Status: ARP attacks detected!
Security level set to: basic
View detected attacks
Read the 'Handling ARP attacks' help
View XArp logfile
www.hackingarticles.in
Get XArp Professional now!
Register XArp Professional
aggressive
high
basic
minimal
The basic security level operates
default attack detection strategy
that can detect all standard atta
This is the suggested level for de
environments.
IP MAC Host Vendor Interface Online Cache First s
192.168.0.1 84-16-f9- f-7a 192.168.0.1 unknown 0x9 - Realtek Et... unkno... no 10/16
192.168.0.100 c0-ee-f -34 192.168.0.100 unknown 0x9 - Realtek Et... unkno... yes 10/16
192.168.0.102 00-0c- 2-8b WIN-1GKSSJ7D... Vmware, Inc. 0x9 - Realtek Et... unkno... yes 10/16
192.168.0.103 00-27- 1-df 192.168.0.103 Rebound Telec... 0x9 - Realtek Et... unkno... no 10/16
192.168.0.104 00-0c- 3e-18 DESKTOP-GFR... Vmware, Inc. 0x9 - Realtek Et... unkno... no 10/16
192.168.0.105 50-82- -64-99 192.168.0.105 unknown 0x9 - Realtek Et... unkno... yes 10/16
192.168.0.107 00-0c- -8e-18 192.168.0.107 Vmware, Inc. 0x9 - Realtek Et... unkno... yes 10/16
192.168.110.1 00-50- 00-01 DESKTOP-GFR... Vmware, Inc. 0xd - VMware ... unkno... no 10/16
192.168.110.254 00-50-5 30-79 192.168.110.254 Vmware, Inc. 0xd - VMware ... unkno... no 10/16
192.168.199.1 00-50-5 00-08 DESKTOP-GFR... Vmware, Inc. 0xf - VMware V... unkno... no 10/16
192.168.199.254 00-50-5 -55-d1 192.168.199.254 Vmware, Inc. 0xf - VMware V... unkno... no 10/16
XArp 2.2.2 - 11 mappings - 3 interfaces - 5 alerts

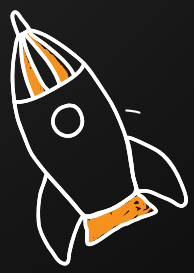
iGNITE Technologies

JOIN OUR TRAINING PROGRAMS

CLICK HERE

BEGINNER

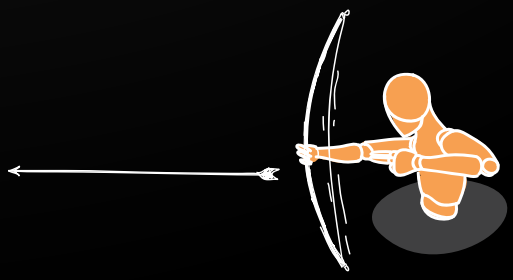
- Ethical Hacking
- Network Pentest
- Bug Bounty
- Wireless Pentest
- Network Security Essentials



ADVANCED

- Burp Suite Pro
- Android Pentest
- Web Services-API
- Advanced Metasploit
- Pro Infrastructure VAPT
- CTF
- Computer Forensics

EXPERT



- Red Team Operation
  - APT's - MITRE Attack Tactics
  - Active Directory Attack
  - MSSQL Security Assessment
- Privilege Escalation
  - Windows
  - Linux