Post-Connection Attacks

- works with WiFi or w/ ethernet
- can gather more info
- intercept data
- modify data on the fly.

* Now we'll install a windows VM for our testing.

Information Gathering

- lets discover all devices connected to the same network. (works for all networks)

```
# netdiscover -r 10.0.2.1/24
```

your range of access (check inet of eth0 / your network)

Subject :

PAGE NO.:
DATE:

* To connect to a network thru Wifi, you'll need a wireless adapter. With that you can work with real networks.

Network Mapping

- use NMAP to gather more information.
- like open ports, operating system, etc.

↳ Open zenmap in kali. (Zenmap = NMap + GUI)

→ in target, give the whole range.
eg: 192.168.1.1|24

① Profile: Ping Scan
→ pings every ip in the range & shows which one responded, & their MAC addresses

② Quick Scan
→ also shows the open ports on each ip that is alive.

③ Quick Scan plus
→ shows the OS running
→ shows device type
→ shows program & program version running on open ports.

* If you see an apple device that has ssh running on it. You can try to connect to it using pass = 'alpine'.

Man-in-the-Middle Attacks

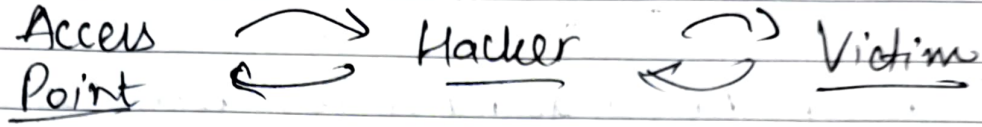
- intercepting connection to sit in the middle
- can see any transferred data.
- can be done by many methods.

Subject :

PAGE NO :
DATE :

① ARP Spoofing

↳ any data, will now flow through the hacker.



★ ARP (Address Resolution Protocol) allows us to link IP addresses to MAC addresses.

→ do 'arp -a' to see how ips are mapped to macaddress.

→ we need to exploit the ARP protocol.

→ we'll tell the AP that we are the IP of victim

→ we'll tell the victim that we are the AP.

Why ARP spoof is possible?

↳ clients accept responses w/o a request

↳ clients trust responses w/o a verification.

★ Lets do the attack now.

Using arpspoof

→ simple & reliable

→ ported to most operating systems

→ usage is always same

★

```
#arpspoof -i eth0 -t 10.0.2.7 10.0.2.1
```

(interface) (target) ip of target (get by doing arp -a on windows)

name of interface

ip of kali gateway

```
#arpspoof -i eth0 -t 10.0.2.1 10.0.2.7
```

★ These two must be run to redirect traffic from client as well as router

Subject : _____

PAGE NO.:
DATE:

★ This attack will work the same way against WiFi, ethernet or wireless adapters.

→ Since your computer (kali) isn't the router, there is a stoppage of requests.
∴ Do port forwarding.

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

★ Other than arpspoof, we can also use bettercap.

Using Bettercap

↳ better than arpspoof.
↳ can be used to capture data
↳ bypass https
↳ dns spoofing
↳ etc...

To run: `# bettercap -iface eth0` (interface name)

★ There are many modules that we can use (type help to see). You can check how to use that module by doing help {name of module}

★ Lets use net.probe:

↳ `# net.probe on`
(discovers clients connected to the network)

→ Do `#net.show` to get a tabulated data of IPs, MAC & names etc for the devices probed.

★ Now lets do arpspoofing using bettercap.

Subject :

PAGE NO.:
DATE:

ARP Spoofing using Bettercap

★ Become the Man-in-the-middle first. So that we can redirect output/input.

→ we can turn on arp.spoof.fullduplex so that it will spoof both router & target.

∴ we do:

```
# set arp.spoof.fullduplex true
```

→ Now we want to specify targets.

```
# set arp.spoof.targets 10.0.2.7
```

↳ target IP (multiple supported)

★ Now we can run the attack:

```
# arp.spoof=on
```

→ Now we are in the middle of the connection. And ∴ we can now spy.

★ Tell bettercap to capture & analyse the data:

```
# net.sniff on
```

(wont work against https)

→ Now any activity done on target can be saved on kali.

★ To do these things, we had to do many things (many steps). Its inefficient. ∴ We'll use a script.

→ We'll make a text file to do this automatically.

```
• net.probe on
• set arp.spoof.fullduplex true
• set arp.spoof.targets {}
• arp.spoof on
• net.sniff on
```

★ To run it, quit bettercap.

Subject :

PAGE NO.:
DATE:

★ do:

bettercap --help (to learn)

★ We are gonna use caplets to run our script.

→ # bettercap -iface eth0 -caplet spoof.cap

And thats it, managed to intercept data succesfully.

★ We need to bypass Https to do this for https sites also.

★ HTTPS

↳ HTTP sends data as plain text.

↳ HTTPS encrypts the data using TLS / SSL.

↳ An interceptor needs to decrypt data even if he becomes MITM. (man-in-the-middle)

★ To bypass this, we need to downgrade HTTPS to HTTP.

→ In our spoof caplet, do:

→ remove:

→ add: set net.sniff.local true

→ then turn net.sniff on.

★ Lets bypass https to http now.

→ Go to terminal & run our spoof.cap.

→ caplets.show will give you all available caplets.

→ we will run hstshijack caplet.

↳ # hstshijack/hstshijack

• now https is downgraded to http.

★ More famous websites have an additional security layer: hsts. Only https is sent or received.

→ ~~earlier we were using ssl stripping.~~

→

Subject : _____

PAGE NO.
DATE:

★ The only practical sol^{n} currently is to make the browser think it loads and correct site but trick it to load another website.
∴ → replace all links for hsts websites with similar links

★ In hstshijack caplet file, you'll see targets & their replacements. You can change them.

★ For chrome, make sure to input the website in the HSTShijack caplet.
→ Having secure DNS disabled is a good feature. because DNS then isn't encrypted. (for hacking)

★ When any website is requested, ~~its dns~~ it is sent to a dns server that returns the ip address of the server hosting that website.
★ When we are MITM, we can redirect users to another dns server rather than a secure one.