**Project Title: Privilege Escalation Techniques on vulnerable Linux Servers**

Ali Zarafsani, CSCE University of North Texas

**Team Members: Group 2**

Dharan Kumar Kunati, MS in Cyber Security, Denton, TX (dharankumarkunati@my.unt.edu)

Sai Meghana Evuri, MS in Cyber Security, Denton, TX (saimeghanaevuri@my.unt.edu)

Ganesh Seeram, MS in Computer Science, Denton, TX (ganeshseeram@my.unt.edu)

Chandrika Gudipati, MS in Cyber Security, Denton, TX (venkatasaichandrikagudipati@my.unt.edu)

**Abstract:**

Privilege escalation is the process of giving authorized users more access and authority over a computer system or network than is appropriate. Hackers frequently try to get higher privileges by exploiting flaws in software, operating systems, or apps in order to access sensitive data, steal information, or engage in other illegal acts. An attacker can increase privileges in a variety of ways, some of which include.

* Security misconfiguration
* Sudo shell escaping
* Exploiting SUID binaries
* Known CVEs

**Introduction:**

This project is about various Privilege Escalation Techniques on vulnerable Linux systems. We have performed various enumeration techniques to gather information on the system, network and to know the applications running. A variety of manual enumeration methods were used, including network, operating system, application, and user enumeration. We have performed automatic enumeration using Linpeas and Linenum tools We have used privilege escalation techniques like sudo shell escape and SUID binary exploit in our project.

**Methods and Implementation**:

Privilege escalation techniques are used after we get some level of access to the machine and then we want to escalate the privileges to different users who might have access to more privileged and important information. For that task, we have performed manual enumeration techniques, automated enumeration techniques, Found some known vulnerabilities of sudo, used those vulnerabilities to perform privilege escalation. We can check ‘whoami’ to check whether we got root access or not.

**Enumeration Techniques:**

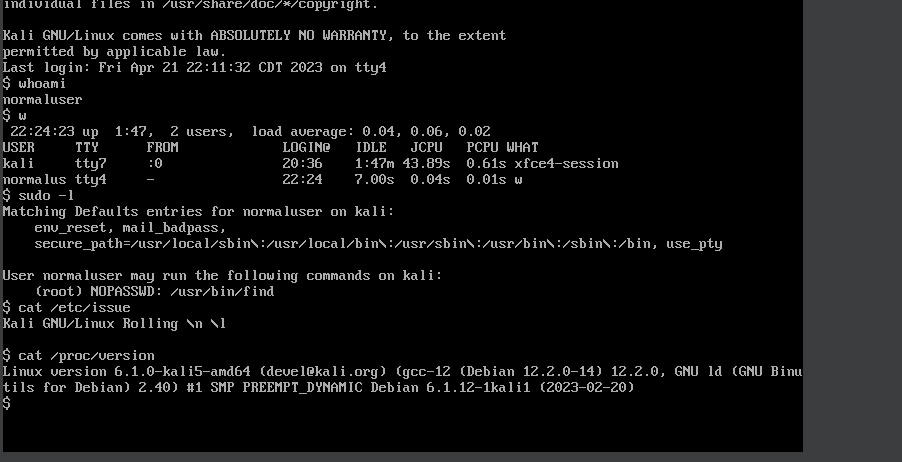
Enumeration techniques play a crucial role in identifying the required information to launch an attack. Port scanning, system fingerprinting, and banner grabbing are some of the approaches we have done to know more about the system . Attackers would be able to get different escalation paths with the information that has been obtained. Acquiring information through these paths is crucial for a successful attack because it allows hackers to locate vulnerable services or individuals to gain access to secured regions.

**Manual enumeration**:

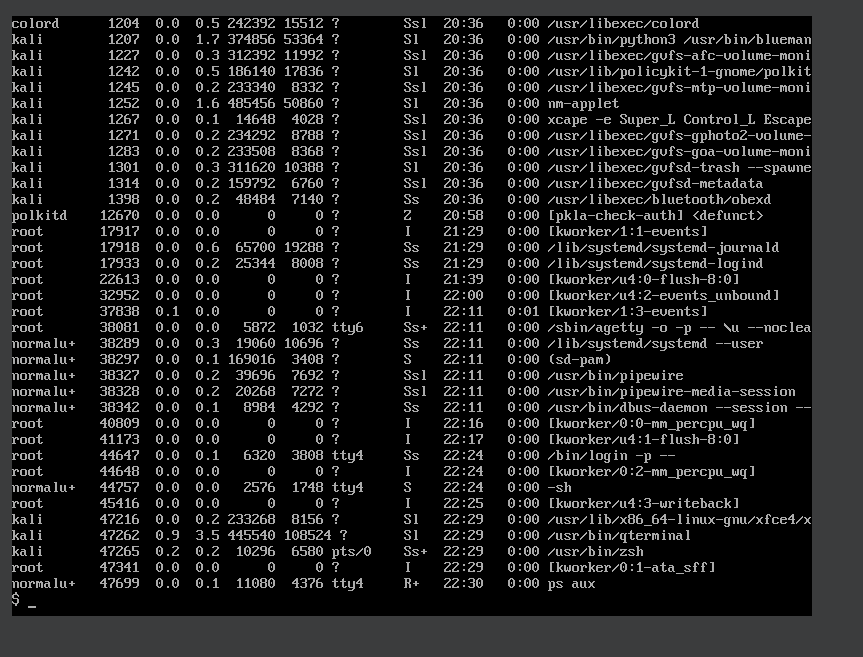
Network Enumeration, Operating System Enumeration, Application Enumeration and User Enumeration are the types of manual enumeration techniques that are used.

For the purpose of the project we have created a ***normal user*** which we have used to escalate.

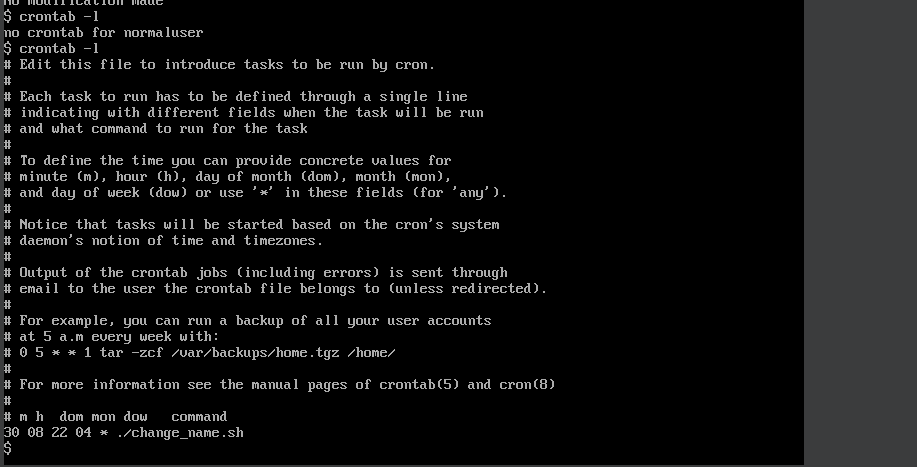
Using the command “Cat /proc/version” an attacker gets the Linux version 6.1.0-kali i5-amd64



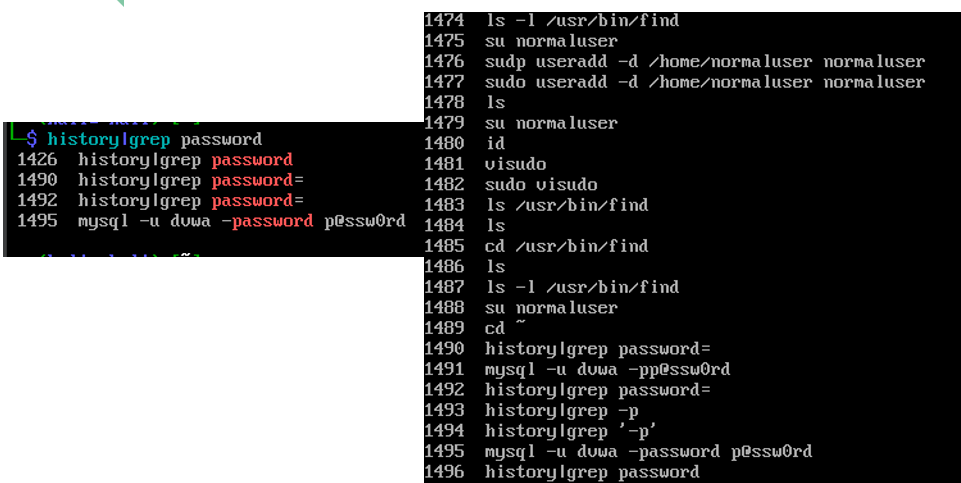
The next step would be to know the services running. Using the “***ps aux***” command we were able to find all running and active processes on the system. The process IDs are also displayed using the command.



Cron jobs are tasks that are scheduled to run at specific times, typically to carry out system maintenance or any other administrative activities. “***crontab -l***” command is used to show the cron jobs currently set up for the current user.



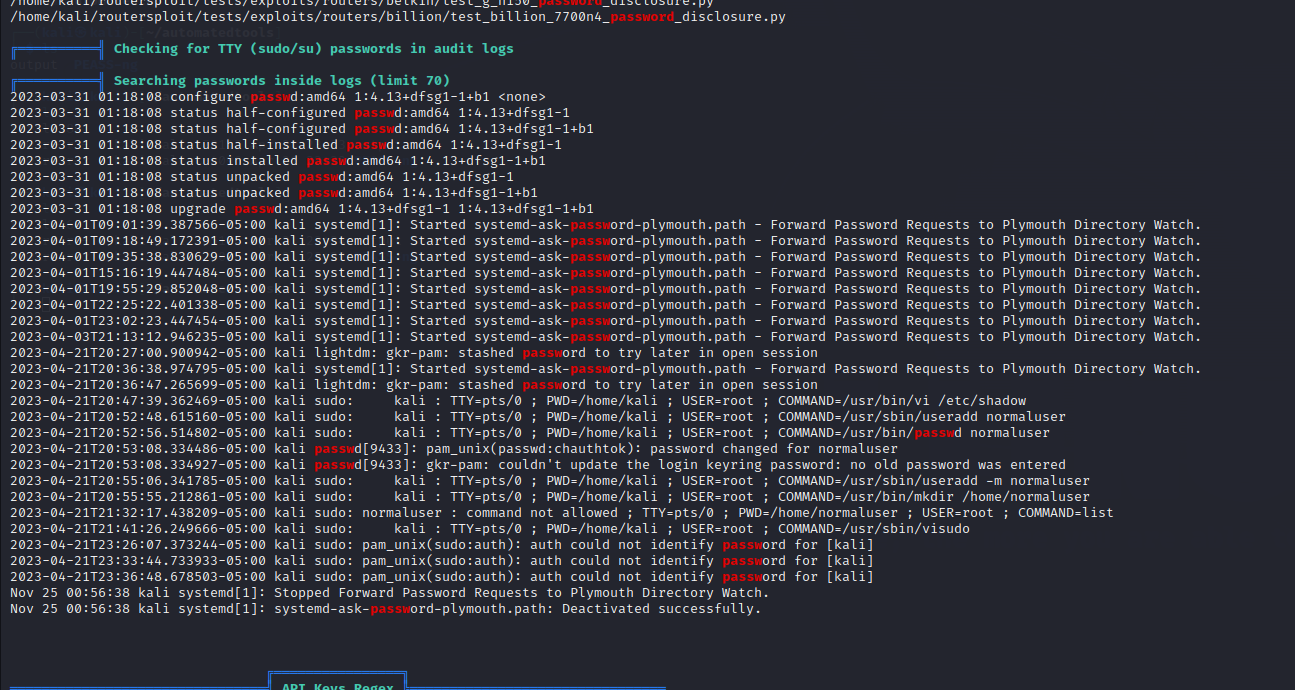
Using the “history” command we can see all the commands the user entered, attackers would also try to find any commands with string “password” in it to obtain the passwords that can be used for further exploitation.



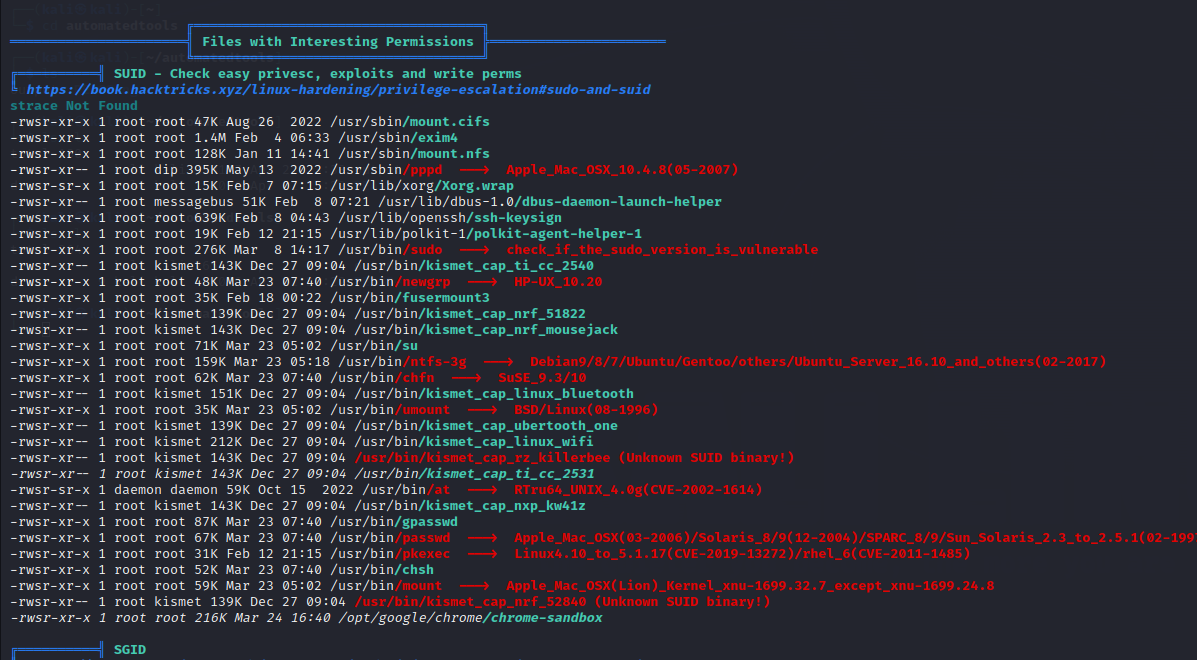
**Automatic Enumeration:**

**Linpeas:**

Linpeas is an enumeration tool used to gather information regarding user, network and vulnerabilities. Here we are searching the word ‘passw’ on log files trying to find user passwords and later use it for privilege escalation.



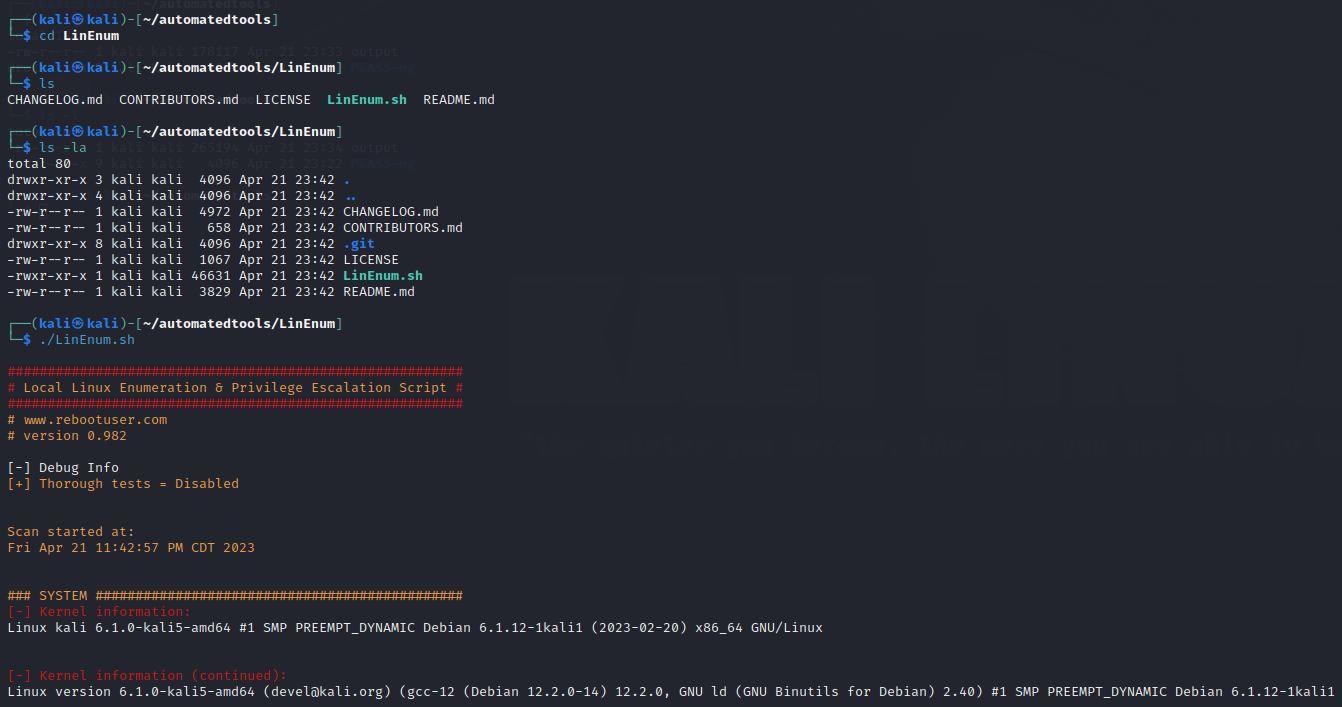
Here are interesting files with SUID permissions given by linpeas

****

**LinEnum:**

LinEnum is also used for the same purpose gathering information about the system, user accounts, network configuration and installed software. It's better to use multiple tools because lots of time one tool might miss out information useful for us.

Using ‘./LinEnum.sh’, we can see the kernel version, number of cpus and vulnerabilities in the system which can be used for exploitation. This can be used to select our exploitation techniques.

****

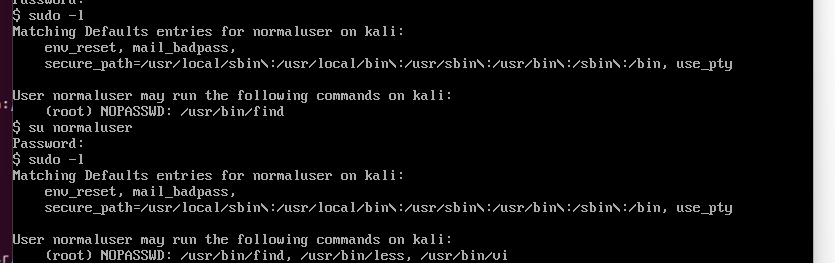
**Implementation(Privilege Escalation):**

**Privilege escalation using Sudo Shell escaping:**

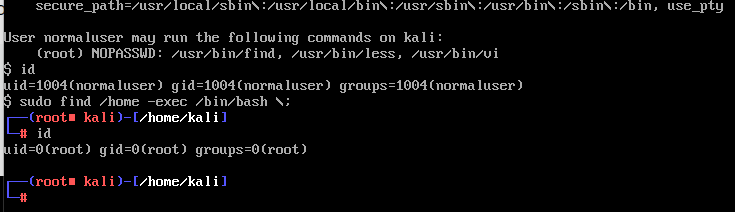
One privilege escalation technique we have used is exploiting sudo command permission. In linux there are normal users but they can run some commands with elevated privileges as sudo users.

Administrators often give permissions to normal users to run some commands with sudo privileges to perform certain tasks that require elevated privileges without granting them full administrative rights. This is a security best practice that helps to limit the amount of access that each user has to sensitive system resources and data.

Given below is the screenshot which shows we have logged in as a normal user, Using command '***sudo -l'*** lets us see what commands the user can run as sudo. The user can run 'find', 'less', 'vi' commands.



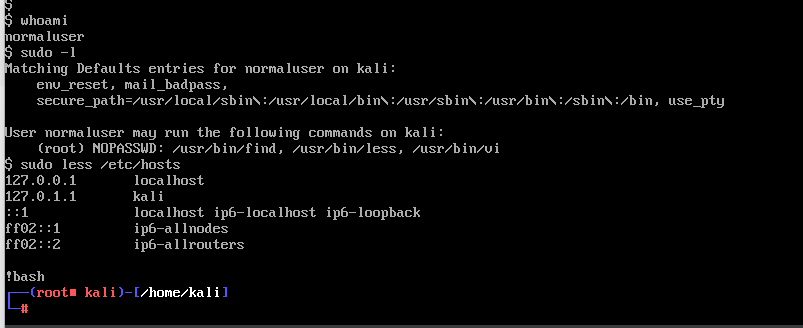
In linux 'find' is a utility which helps users to search files, binaries, executables etc. By running the command “***sudo find /home -exec /bin/bash \;***” we got the shell without even entering any password and got access to root from a normal user.



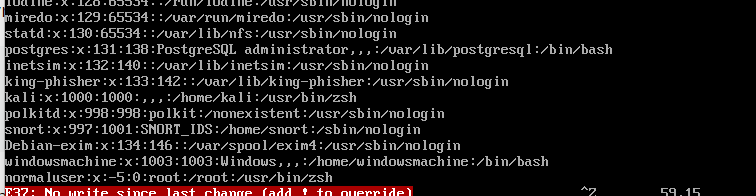
**‘Vi’** is a text editor in linux which allows users to create, view and edit text. Usually, people use it to edit configuration files. By just entering ***!bash*** we were able to get root access directly.

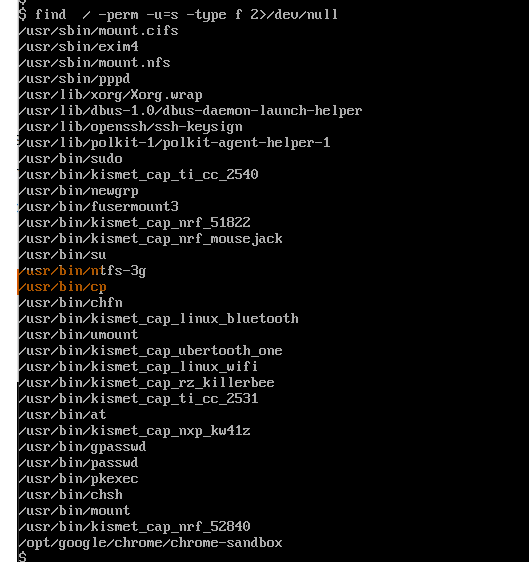


‘Less’ is a pdf viewer which doesn’t have any write access but by giving ***!bash*** we were able to get root access

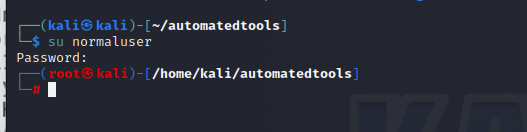


Unlike shadow file users will have access to password files. To the user ***normaluser*** which we have created we have given command /root/usr/bin/zsh - using this command and zshell we got access to root shell. This is a SUID binary vulnerability, the program should check whether or not the write user has access to files before performing next step or giving access to the files.





We are able to get access to root just by logging in as a normal user.



**Analysis and Results:**

This project illustrates how to get root shell access by exploiting the sudo shell escaping vulnerability. The sudo command-line utility in Linux is intended to allow users to run tasks with elevated rights. A vulnerability in the sudoers configuration file, on the other hand, can allow an attacker to escalate privileges to obtain root access.

We were able to uncover and exploit the sudo shell escaping vulnerability to acquire elevated shell access. We were able to overcome the sudo configuration file's constraints and run arbitrary commands as root by inserting commands into the sudo shell escape sequence.

According to our analysis, the vulnerability can be mitigated by making sure that the sudoers configuration file is set up correctly and that users don't have unauthorised access to the sudo function. Regular system updates and security checks can also aid in preventing such exploits from being used.

**Conclusions:**

This project has shown how serious the sudo shell escaping vulnerability is and how crucial it is to set up the sudoers file correctly and restrict user access to the sudo program. To stop such exploits from being used, system administrators must frequently update and monitor their systems. Below we have also written some ways to defend against common privilege escalation techniques.

* Implement least privilege principle.
* Update software with security patches as necessary.
* Use two-factor authentication and strong passwords.
* Utilise endpoint security tools like antivirus software and intrusion detection systems.
* Applying Role based access control.
* Through the use of firewalls and network segmentation, reduce network exposure.

**References:**

[PEASS-ng/linPEAS at master · carlospolop/PEASS-ng (github.com)](https://github.com/carlospolop/PEASS-ng/tree/master/linPEAS)

[Linux Privilege Escalation | part of Kali Linux Penetration Testing Bible | Wiley Data and Cybersecurity books | IEEE Xplore](https://ieeexplore.ieee.org/document/9936713)

[rebootuser/LinEnum: Scripted Local Linux Enumeration & Privilege Escalation Checks (github.com)](https://github.com/rebootuser/LinEnum)

[Home - Alexis Ahmed](https://alexisahmed.com/) - A book on privilege escalation techniques

[Linux Privilege Escalation for Beginners - YouTube](https://www.youtube.com/watch?v=ZTnwg3qCdVM)