Vulnerability Assessment & Penetration Testing (VAPT) Report

Sudo-Based Privilege Escalation

Lab Overview

- Platform: TryHackMe (Linux Privilege Escalation Lab)
- Attack Type: Local Privilege Escalation
- Initial Access Level: Low-privileged user
- Objective: Escalate privileges to root using sudo misconfiguration

In this lab, we analyzed the system for potential privilege escalation vectors related to misconfigured sudo environment variables, specifically LD_PRELOAD.

The objective of this lab was to determine whether the LD_PRELOAD environment variable could be leveraged to execute arbitrary code with elevated privileges.

```
user@debian:/home$ 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
```

LD_PRELOAD is a function that allows any program to use shared libraries. This blog post will give you an idea about the capabilities of LD_PRELOAD. If the "env_keep" option is enabled we can generate a shared library which will be loaded and executed before the program is run. Please note the LD_PRELOAD option will be ignored if the real user ID is different from the effective user ID.

The steps of this privilege escalation vector can be summarized as follows;

1. Check for LD_PRELOAD (with the env_keep option)
2. Write a simple C code compiled as a share object (.so extension) file
3. Run the program with sudo rights and the LD_PRELOAD option pointing to our .so file

Step 1: Check for LD_PRELOAD via env_keep

The first step involved enumerating sudo permissions using the following command:

sudo -l

We reviewed the sudo configuration to check whether the LD_PRELOAD environment variable was preserved using the env_keep option. If LD_PRELOAD is allowed, it can be abused to preload a malicious shared library before the execution of a privileged binary.

```
$ sudo -l
Matching Defaults entries for karen on ip-10-48-152-156:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User karen may run the following commands on ip-10-48-152-156:
    (ALL) NOPASSWD: /usr/bin/find
    (ALL) NOPASSWD: /usr/bin/less
    (ALL) NOPASSWD: /usr/bin/nano
```

However, the sudo configuration was found to have env_reset enabled, and LD_PRELOAD was not preserved in the environment variables. As a result, the LD_PRELOAD privilege escalation technique was not applicable on this system.\

Since LD_PRELOAD was not allowed via env_keep, it was not possible to preload a malicious shared object when executing commands with sudo privileges. Therefore, this attack vector was ruled out during the assessment.

LD_PRELOAD WORK WITH SUDO WHEN env_keep+=LD_PRELOAD explicitly allowed

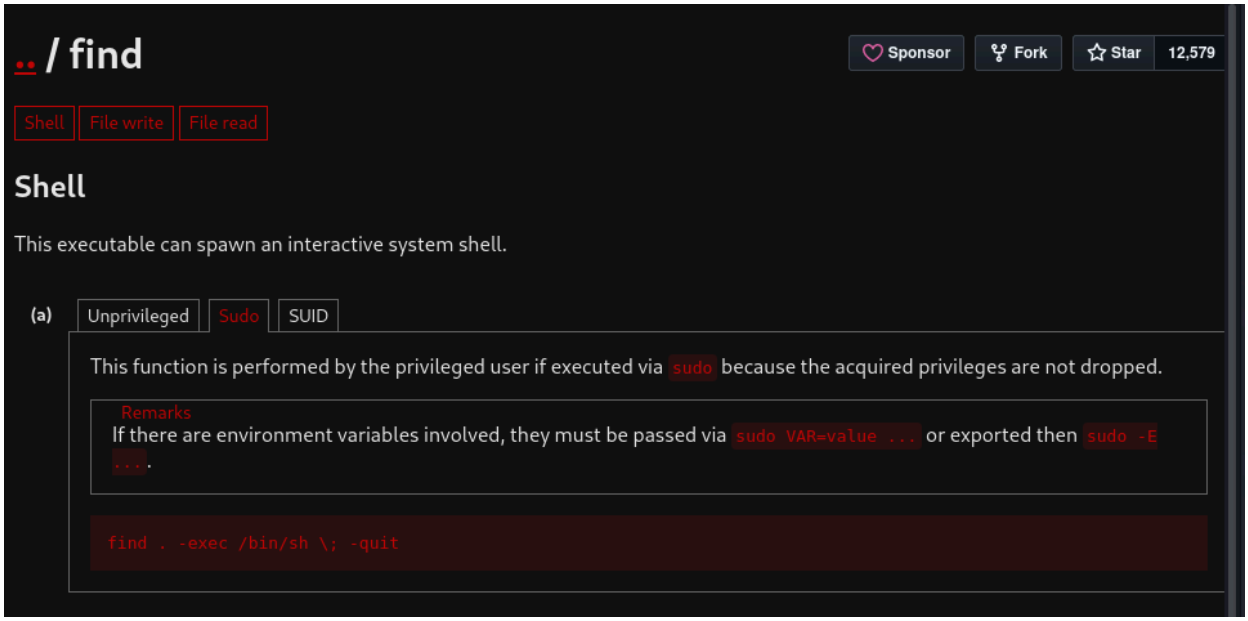
BUT GOOD NEWS 😈

We have direct sudo privilege escalation sign

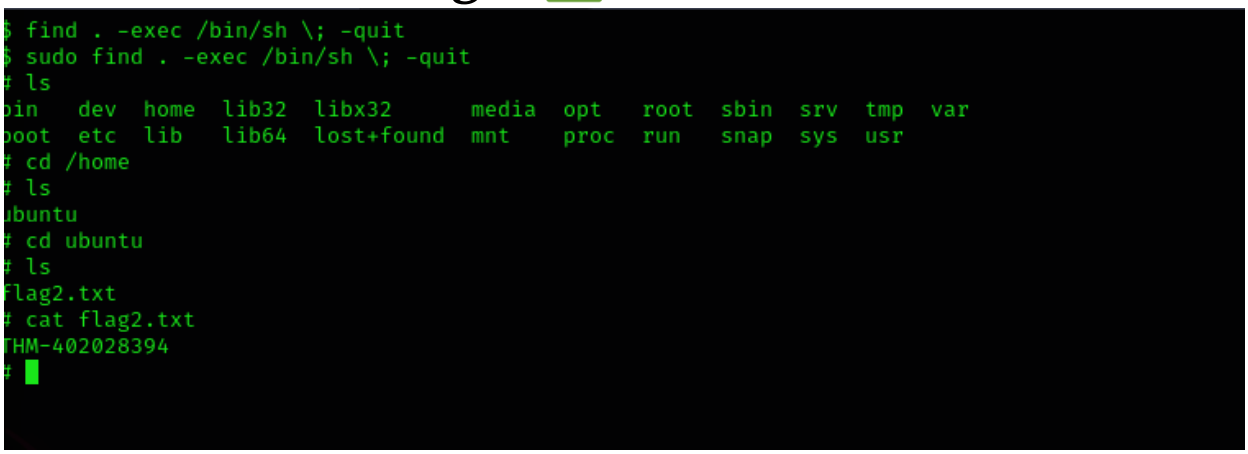
Lets look this 3 are listed in GTFObins

/usr/bin/find
/usr/bin/less
/usr/bin/nano

First we look for /find Here we got it



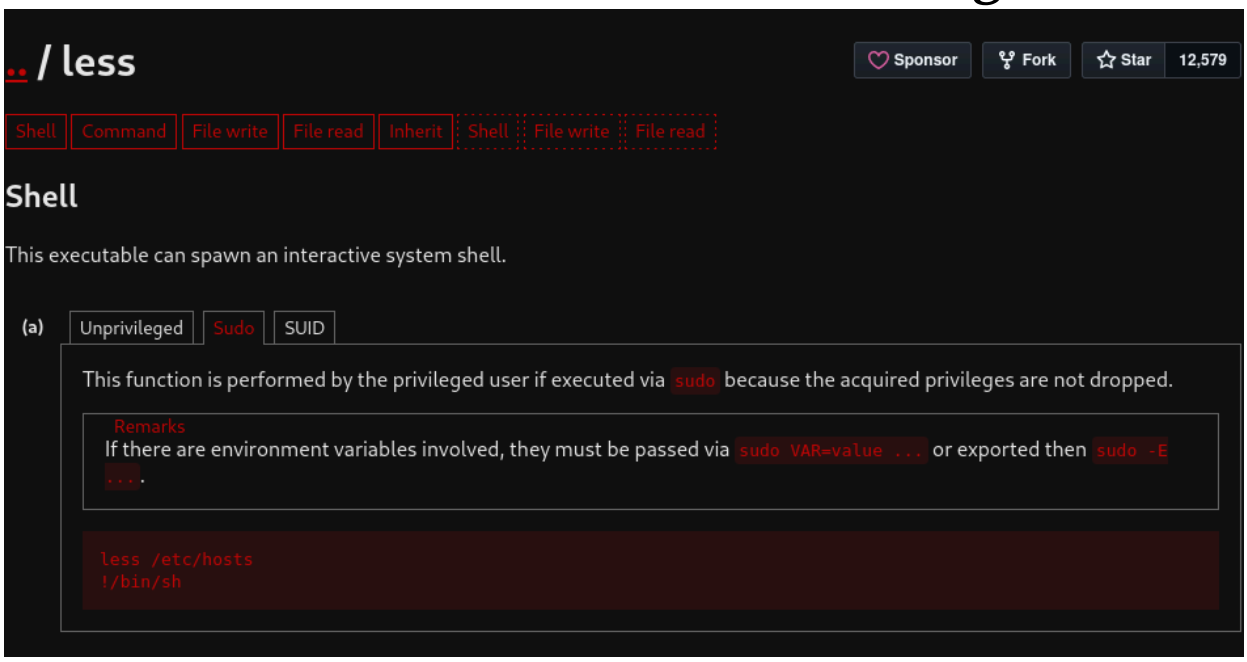
I copy the command sudo find . -exec /bin/bash \; -quit
Past in terminal and look ....wowww ... boom we got ✅ ROOT shell



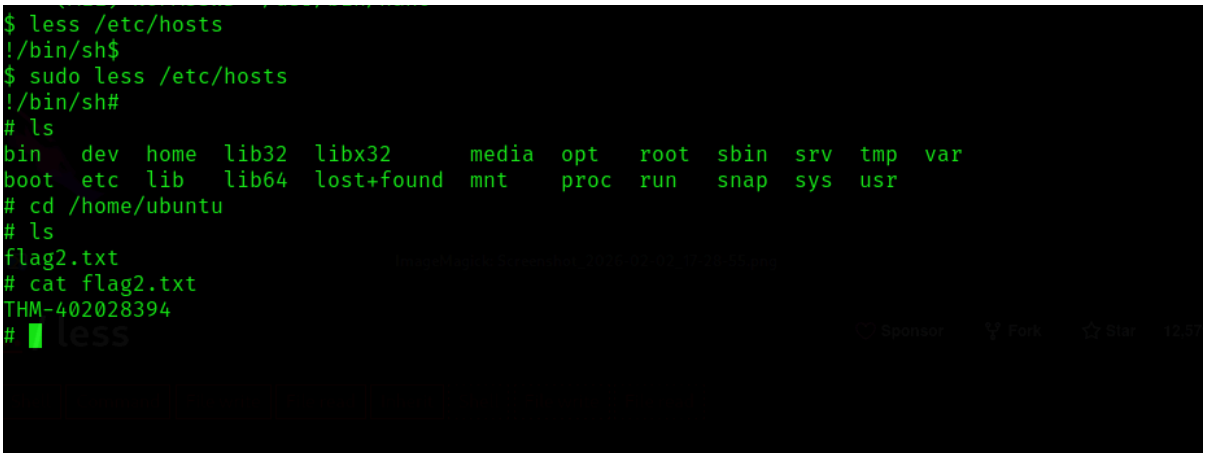
Also we capture the flag

The user was allowed to execute the following binary with root privileges
(ALL) NOPASSWD: /usr/bin/less

I look in GTFOBins for less and got

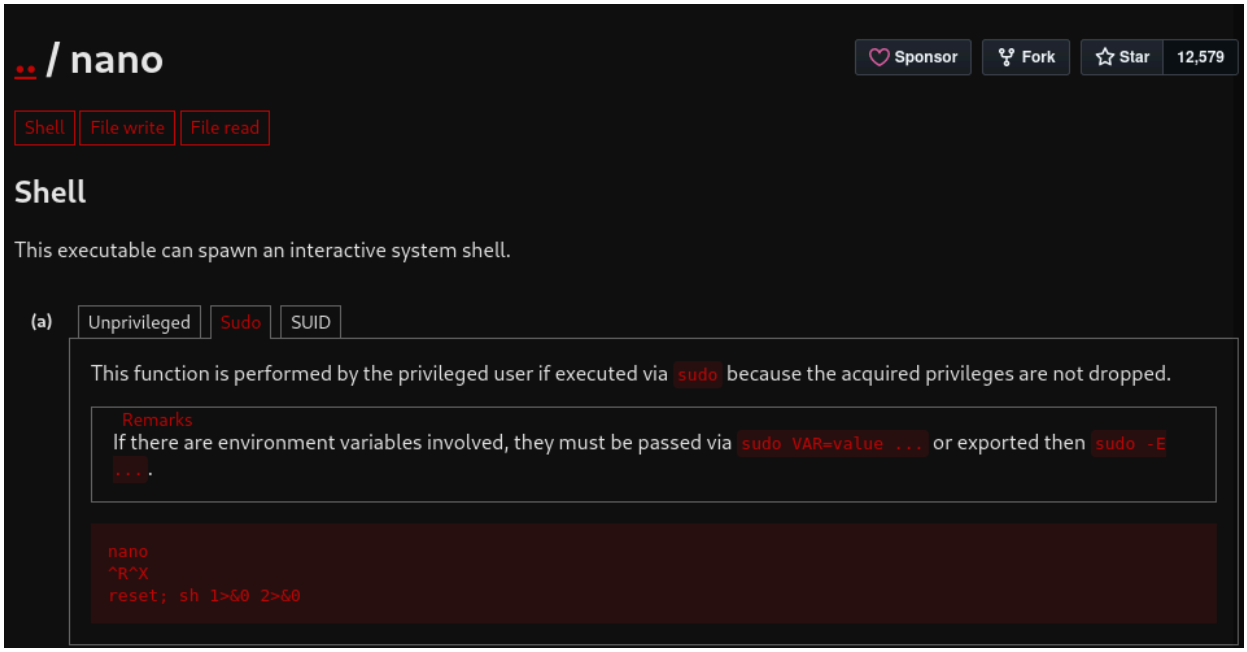


Copy the command and paste into terminal



See root gotted and flag also

✅ ROOT via nano
Again look in GTFOBins

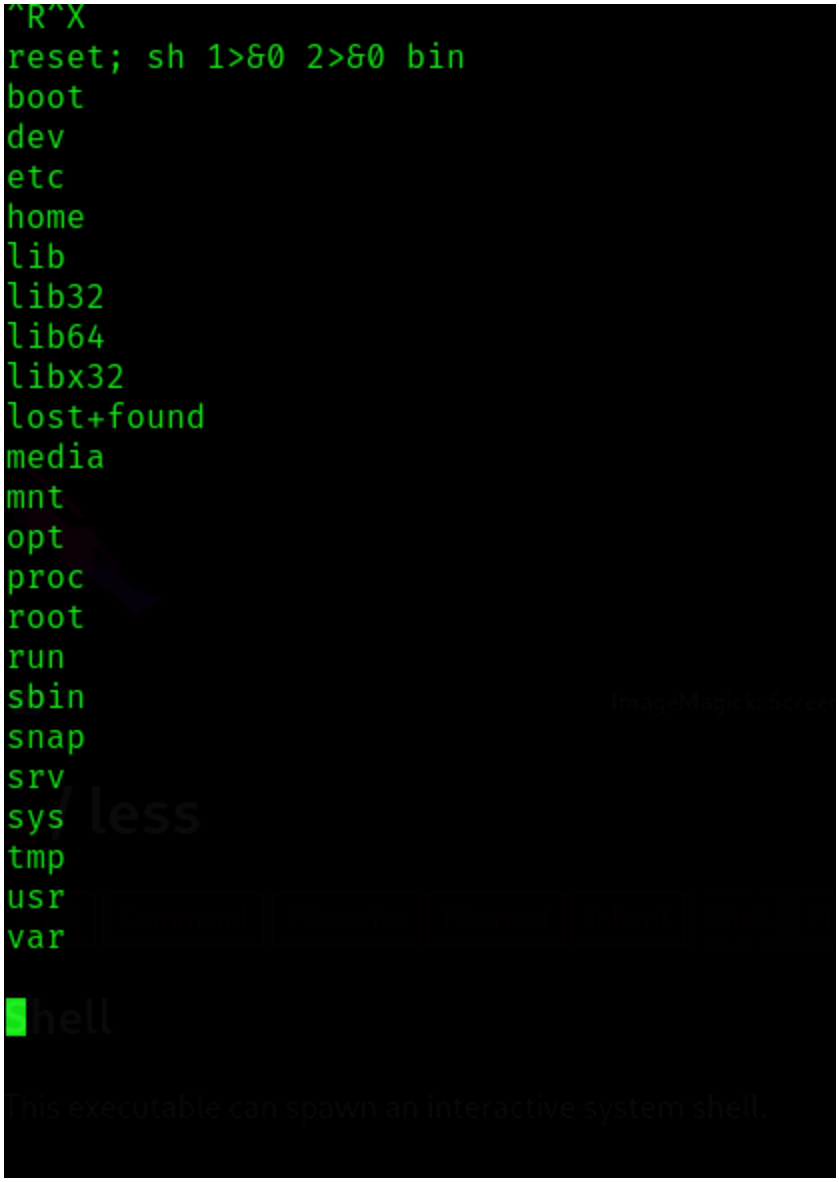


copy this command in the terminal

Type Ctrl + R | Ctrl + X

reset; bash

Boom 💣 root shell



Risk Management

LD_PRELOAD exploitation was checked but not applicable due to env_reset.
However, misconfigured sudo permissions

(NOPASSWD) allowed execution of dangerous binaries as root, leading to privilege escalation.

Conclusion

Even with secure environment variables, insecure sudo rules resulted in full root compromise. Sudo permissions must follow the principle of least privilege.