Kernel Privilege Escalation – VAPT Report

Platform: TryHackMe
Attack Type: Linux Kernel Exploitation
Assessment Type: Educational / Lab Environment

---

1. Executive Summary

This report documents a successful **Linux Kernel Privilege Escalation** attack performed in a controlled TryHackMe lab environment. The assessment demonstrates how an outdated and vulnerable Linux kernel can be exploited by a local attacker to gain **root-level access**, leading to complete system compromise.

---

2. Objective

The objective of this assessment was to:

- Gain initial low-privileged access via SSH
- Identify the Linux kernel version
- Research publicly available kernel exploits
- Exploit the vulnerability to escalate privileges to root
- Demonstrate impact by accessing restricted files (flag)

---

3. Scope

- **Target:** Linux Virtual Machine (TryHackMe)
- **Initial Access:** SSH (Low-privileged user)
- **Attack Type:** Local Kernel Privilege Escalation

- **Tools Used:** SSH, uname, gcc, Exploit-DB, nano

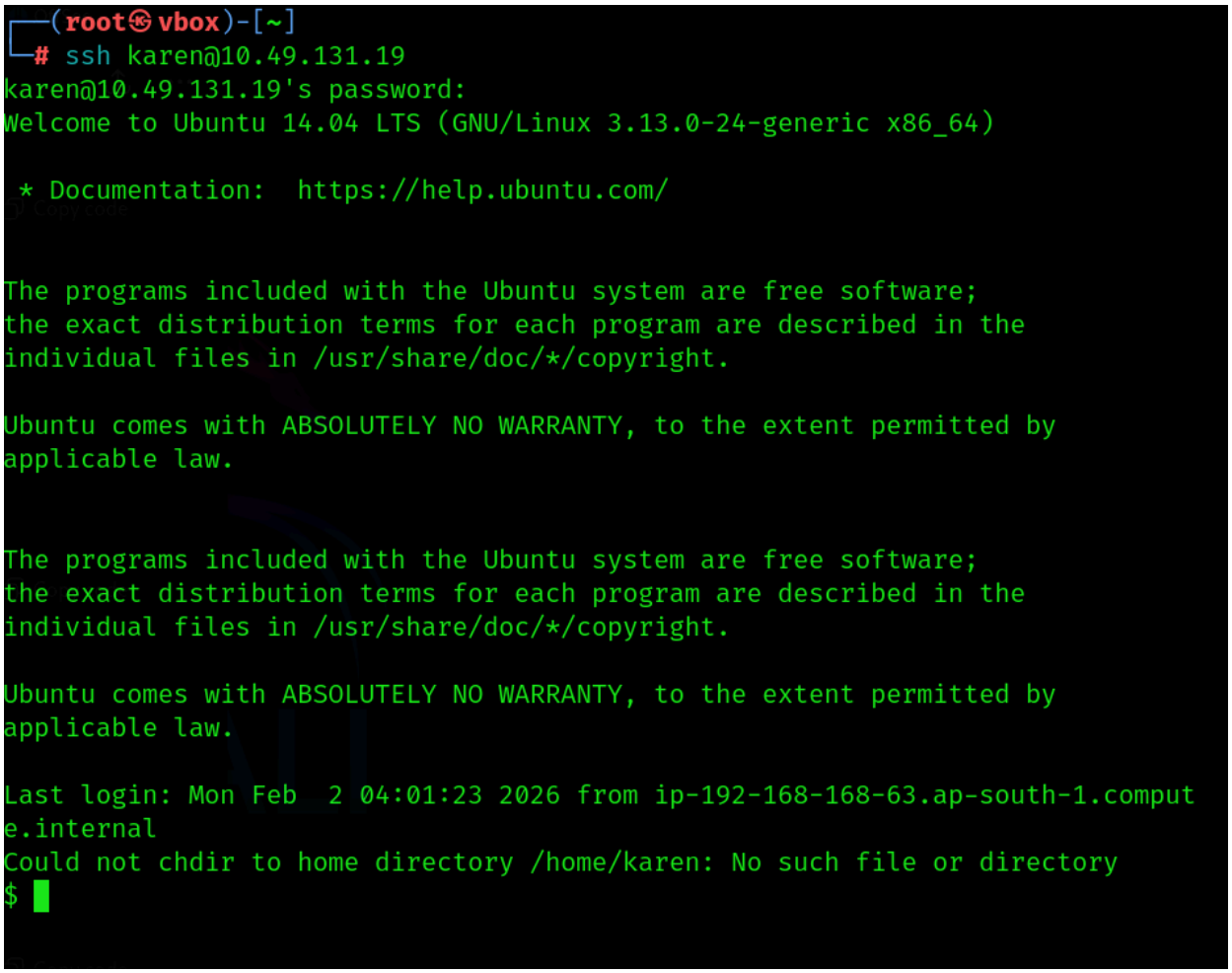
---

4. Initial Access (SSH Login)

The attacker successfully logged into the target system using SSH with provided credentials.

Commands Used:

```
ssh <user>@<target-ip>
whoami
id
```



Result:

✔ Access confirmed as a **low-privileged user**

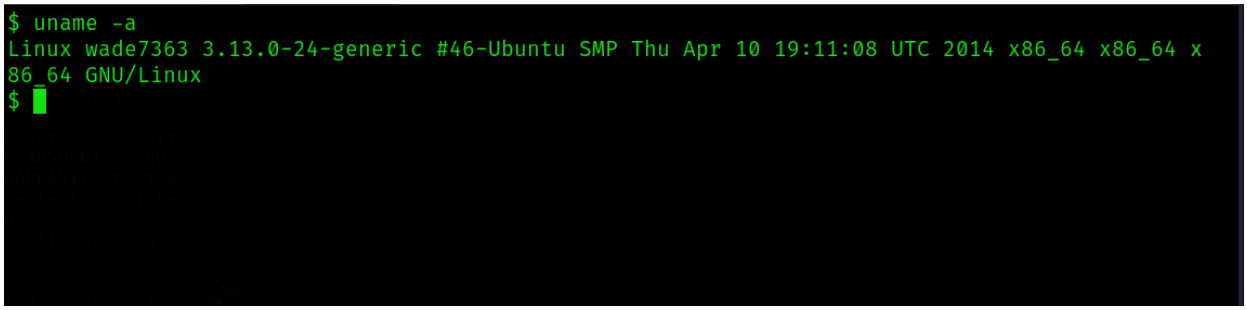
---

5. Enumeration – Kernel Version Discovery

After initial access, system enumeration was performed to identify kernel-related vulnerabilities.

Command Used:

```
uname -a
```



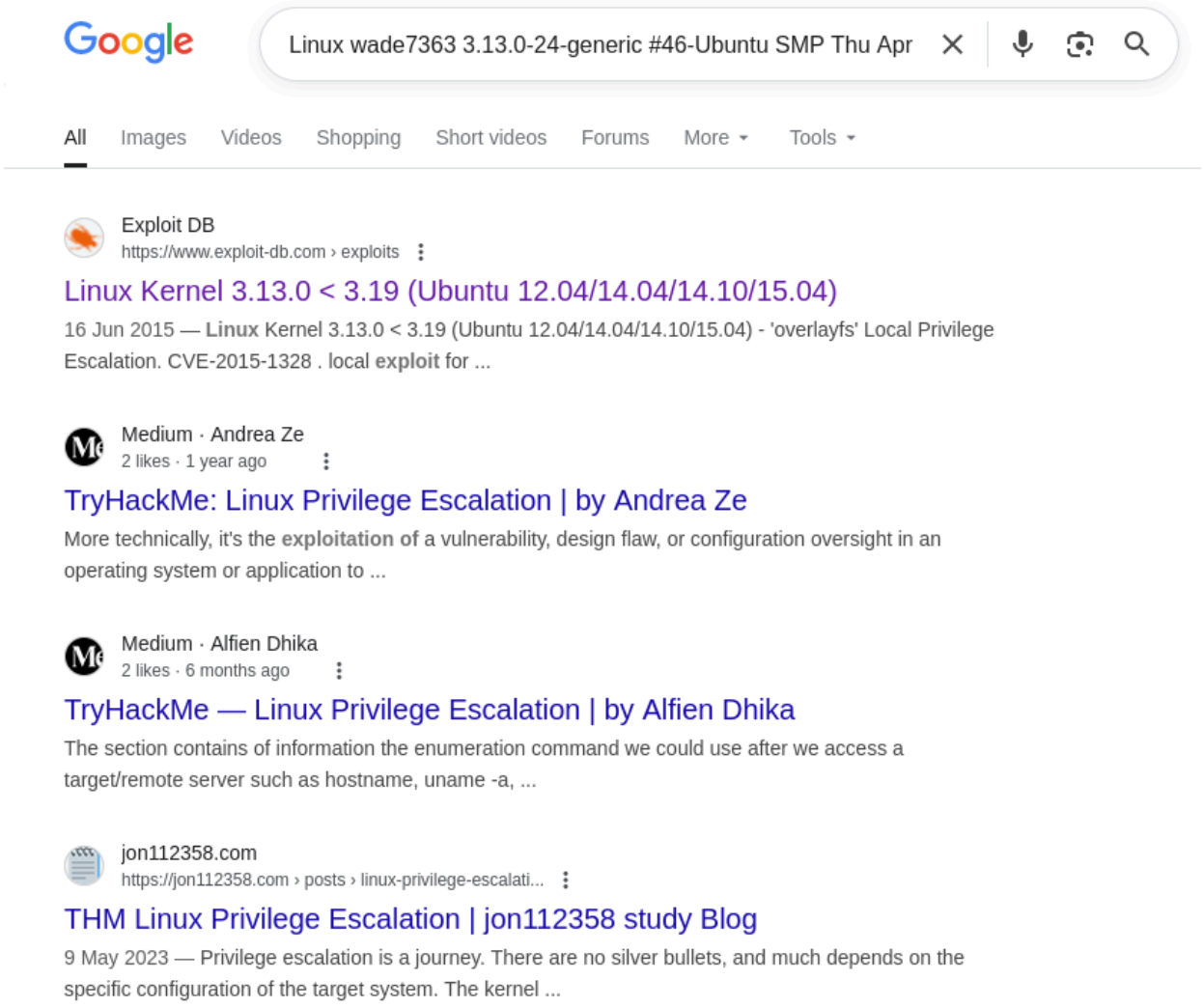
The kernel version information was noted and used for further vulnerability research.

---

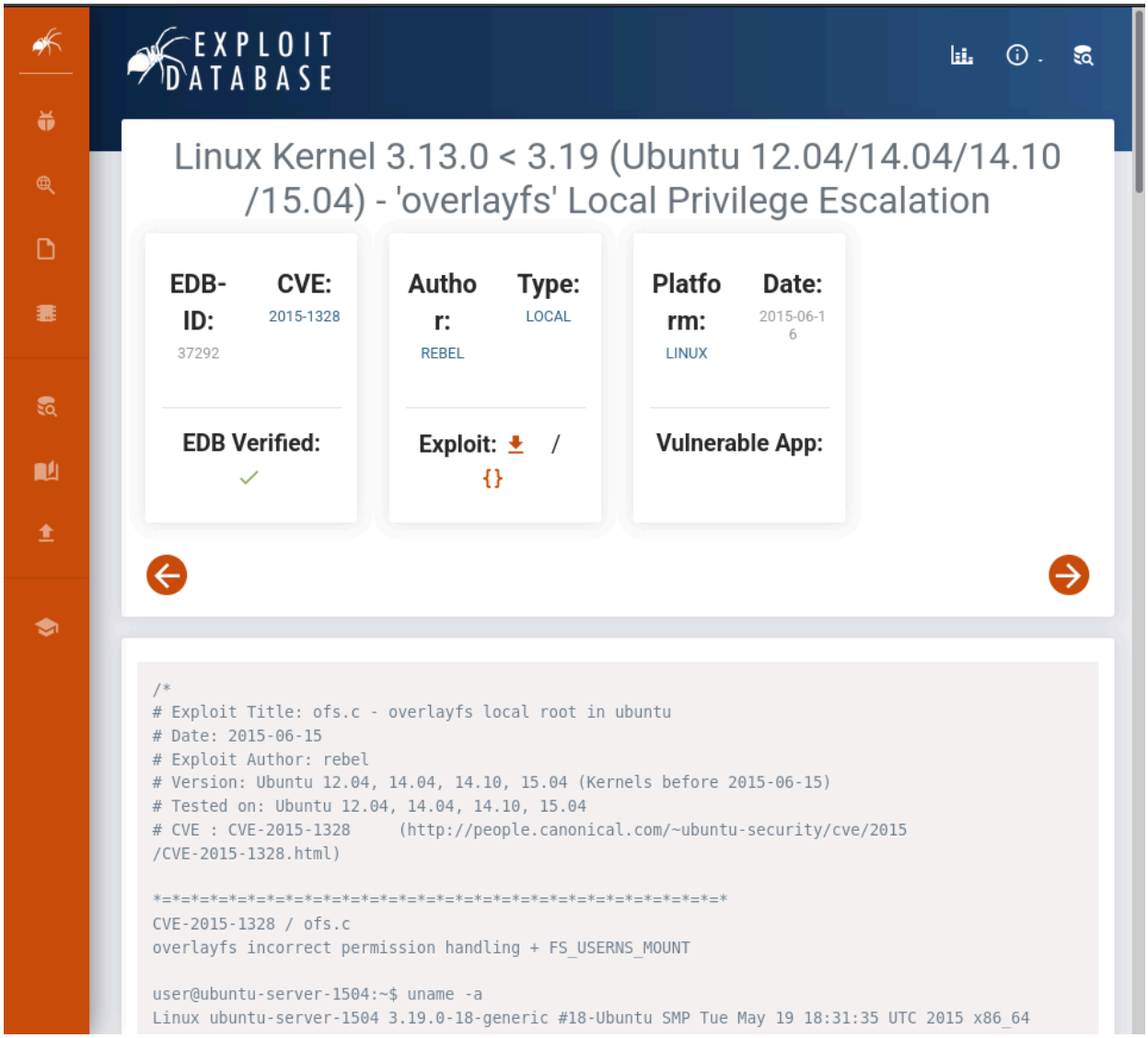
6. Vulnerability Research

The identified kernel version was searched on **Google** and **Exploit-DB**.

Click on the first link also highlight in screenshot



A publicly available **Local Privilege Escalation (LPE)** exploit written in C was found that matched the target kernel version.



---

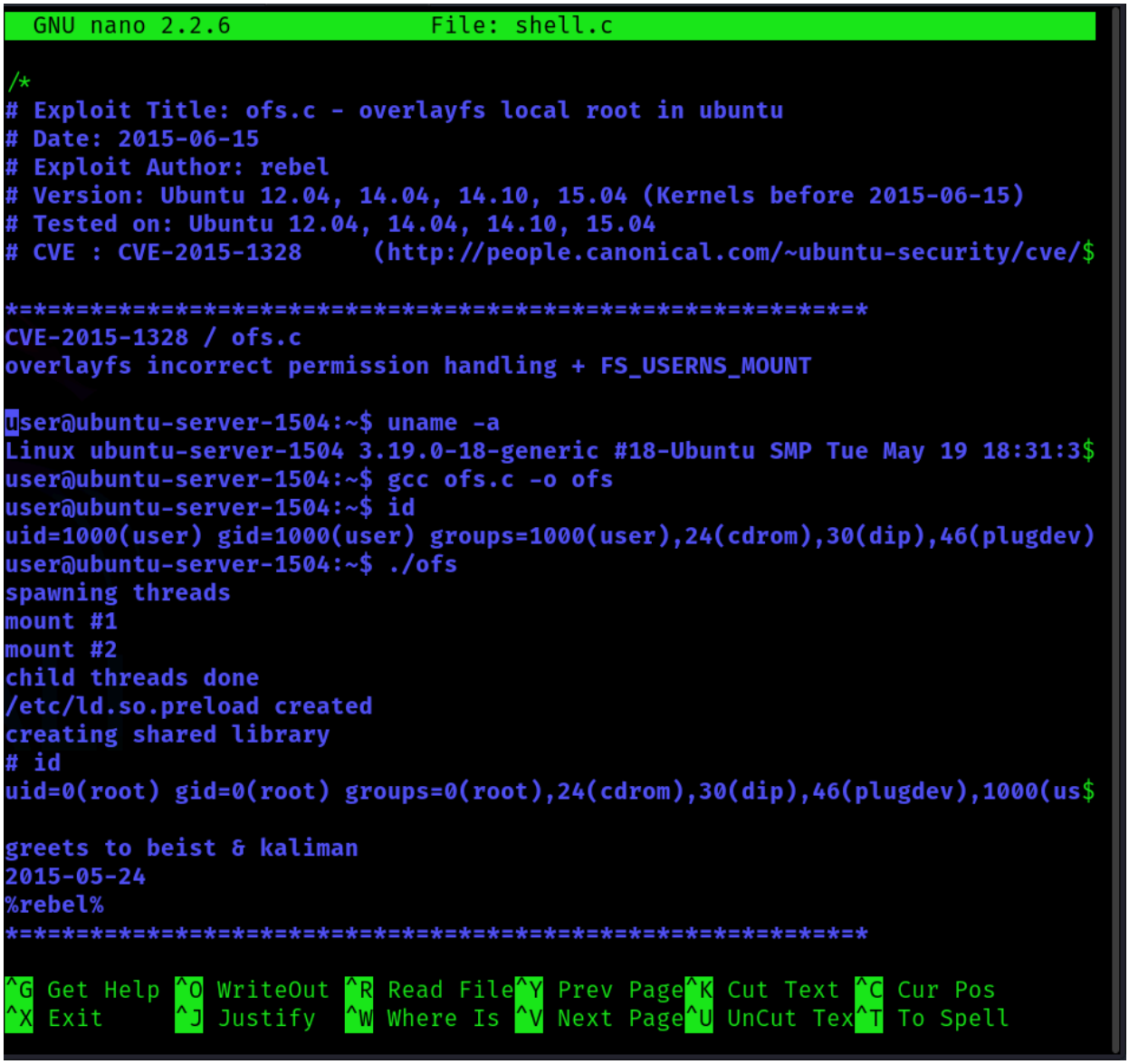
7. Exploitation Process

The exploit code was copied and compiled locally on the target system.

Steps Performed:

```
cd /tmp
```

nano shell.c



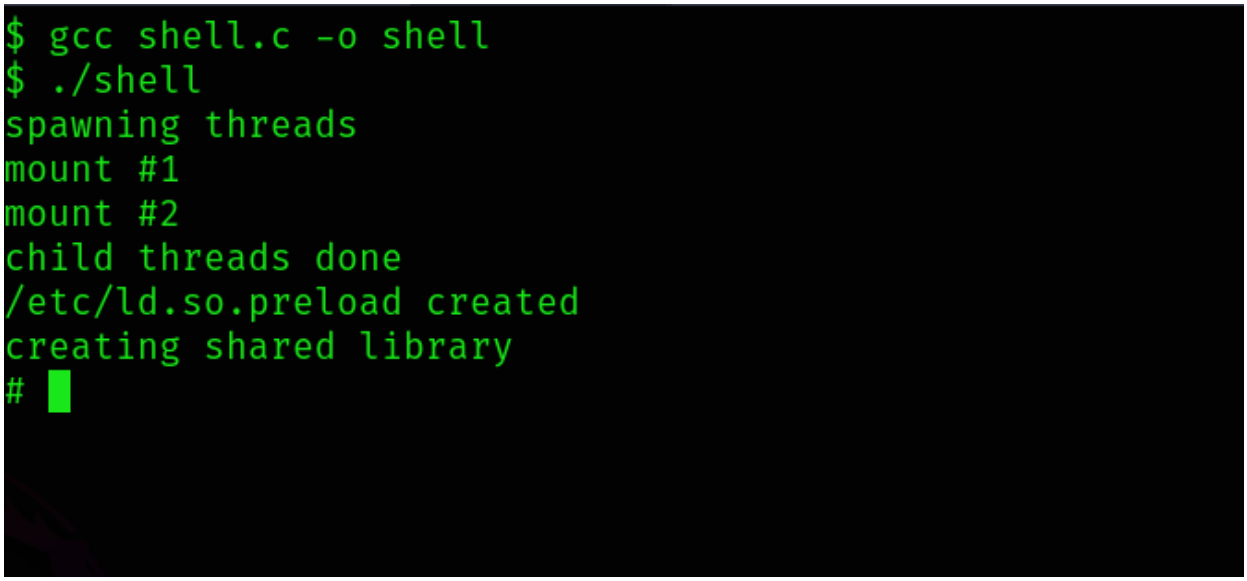
Save the code into shell.c

Save & exit

RUN —-- this file by command

gcc shell.c -o shell

./shell



Yeahhhh !!! boom .... We got root access

Result:

✔ Successful execution of exploit

✔ **Root shell obtained**

---

8. Post-Exploitation & Proof of Compromise

After privilege escalation, root access was verified and sensitive files were accessed.

Commands Used:

```
whoami
cd /home
ls
```

```
cd matt
ls
cat flag1.txt
```

```
# cd /home
# ls
matt
# cd matt
# ls
Desktop    Downloads  Pictures  Templates  examples.desktop
Documents  Music      Public    Videos     flag1.txt
# cat flag.txt
cat: flag.txt: No such file or directory
# cat flag1.txt
THM-28392872729920
#
```

Flag Retrieved:

`THM-28392872729920`

This confirms **full system compromise**.

---

9. Risk & Impact Analysis

Vulnerability	Risk Level	Impact

Outdated Linux Kernel	**Critical**	Full root access, total system compromise

---

10. Mitigation Recommendations

- Regularly update and patch the Linux kernel
- Apply automatic security updates
- Restrict local shell access for untrusted users
- Monitor for exploit compilation tools (gcc) on production systems
- Conduct periodic vulnerability assessments

---

11. Conclusion

This assessment highlights the critical risk posed by outdated Linux kernels. Kernel-level

vulnerabilities allow attackers to bypass all security controls and gain root access. Proper patch management and system hardening are essential to prevent such attacks.