Executive Report

Exploiting Target Boromir

(10.0.6.51)

Big Money Inc.

TEL:819-1280
EMAIL:akatsuki@hotmail.com
MEMBERS: David Thomsen and Robert Segee

Table of Contents

Introduction

The purpose of this report is to show the efforts and methods that were used in this investigation. It will include the exploits used, as well as command inputs, and lastly, the results.

Executive Summary

Boromir was first able to be accessed by gaining access to Rivendell via Wordpress Plugin Vulnerabilities. There is an exploit for this version of wordpress that allows the malicious creation of an admin account, granting full access to the back end of the webpage. This allowed credentials of Elrond(Rivendell SSH user) to be gathered and changed in order to create a foothold within Boromir.

Boromir has a known exploit through WebSVN, where attackers are able to perform remote code execution using a malicious payload. The malicious payload was delivered by first hosting the contents of boromir on the attacker's box where they were then able to inject the payload and gain unprivileged access to boromir@10.0.6.51. Due to improper configurations, using the command **[su root]** and using boromir's password, they were able to privilege escalate to an administrator account.

Objective

The objective is to conduct a penetration test on *boromir.rivendell.org*. Boromir is located on the *RIVENDELL* network which requires first gaining access to *fw-rivendell*. We were tasked with leveraging vulnerabilities found across both systems and network in order to elevate access up to administrator's access.

Recommendations

I would recommend a few solutions or possible deterrents to stop this type of attack happening in an uncontrolled environment.

1. DO NOT under any circumstances store the passwords or password hashes in a non-administrator accessible file or directory. Any credentials even if they are not administrator accounts can be used to pivot to different exploits allowing attackers to gain complete access to the machine.
2. DO NOT use versions of applications that have active verified and known vulnerabilities. WebSVN has a multitude of different CVEs, this can be solved by using an alternative to WebSVN or updating to the latest version.
3. In general, password security was a huge issue in this simulation. The sharing of passwords as well as the lack of protection regarding them had provided many additional points of access. It is highly recommended that there are security protocols in place regarding employee passwords and credentials. Implementing higher quality password policy would provide employees, as well as the company, to maintain a secure environment.

Exploitation of Rivendell

Side note: This target was on a different network than the previous machines on the shire network. This meant that it was necessary to go through the firewall: *wp-rivendell*

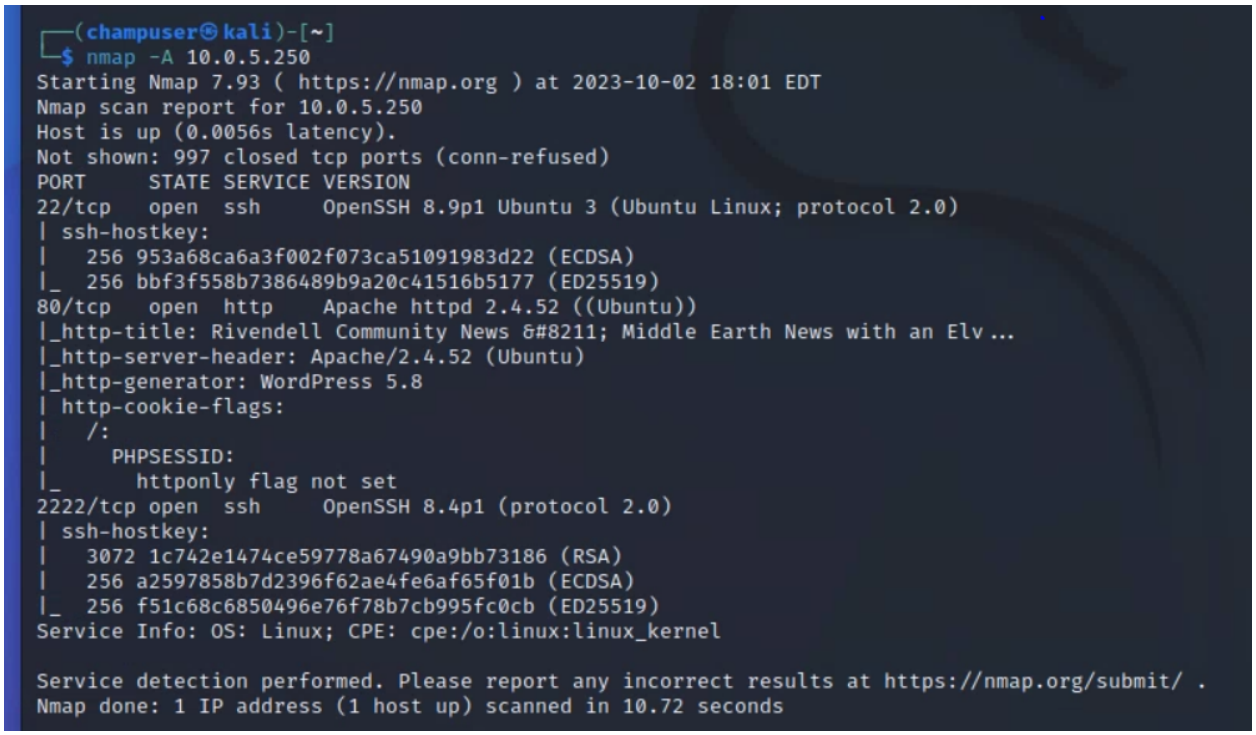
Network Access: fw-rivendell.shire.org // 10.0.5.250

Nslookup:

```
┌──(champuser㉿kali)-[~]
└─$ nslookup fw-rivendell.shire.org 10.0.5.22
Server:         10.0.5.22
Address:        10.0.5.22#53

Name:   fw-rivendell.shire.org
Address: 10.0.5.250
```

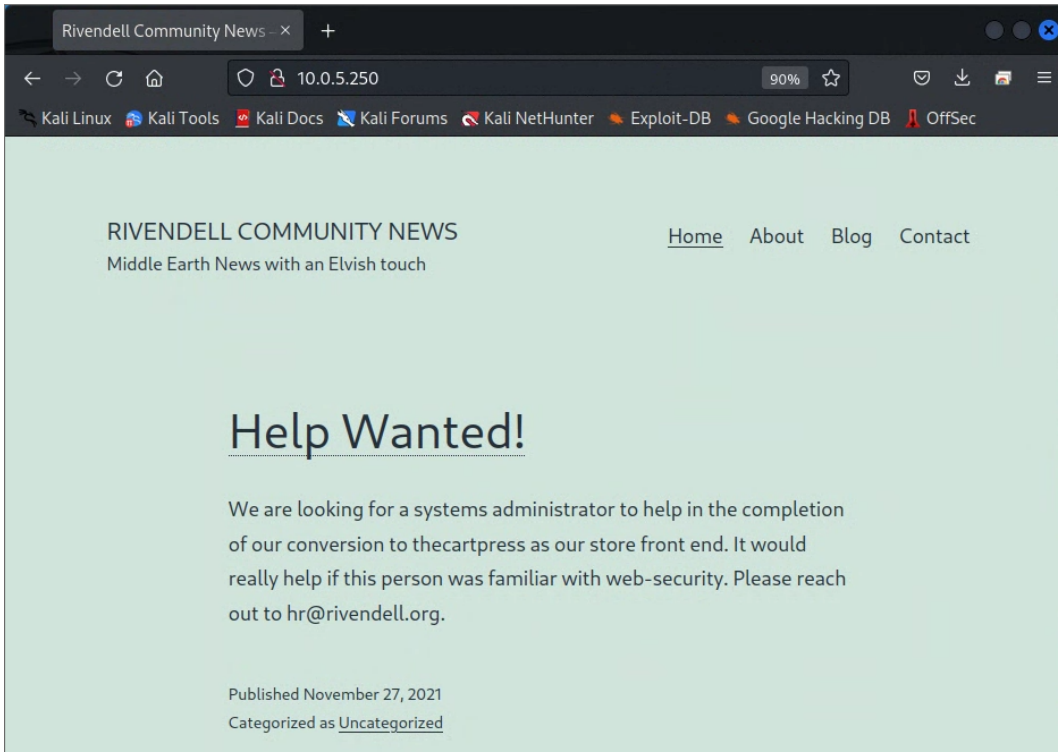
Used nslookup in order to grab the address for the server.

NMAP (looking for open ports and possible vulnerabilities):

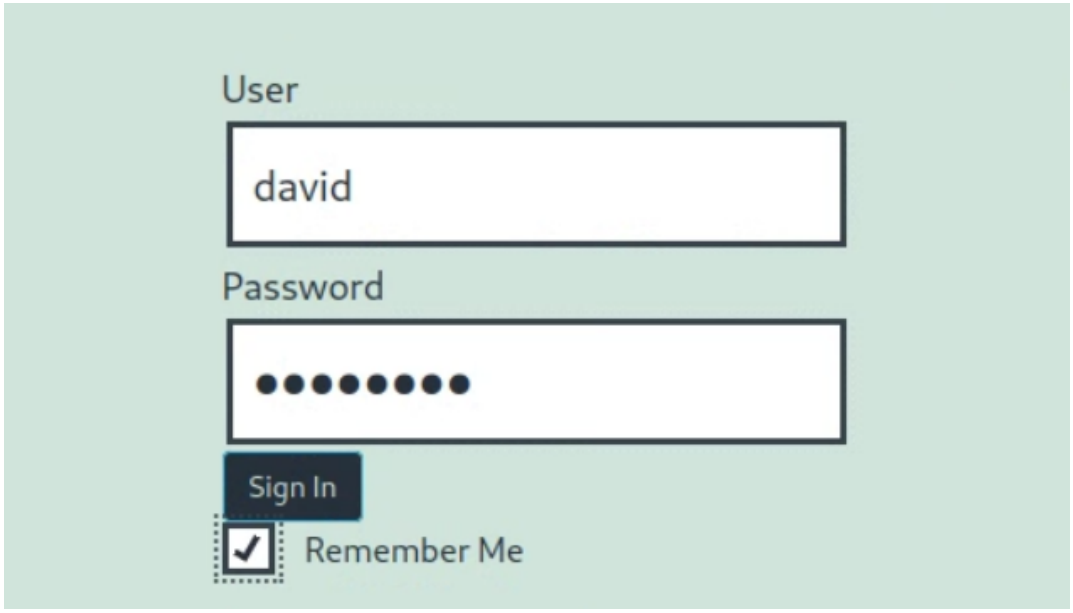


Noticed port 80 was open, so we navigated to this webpage.

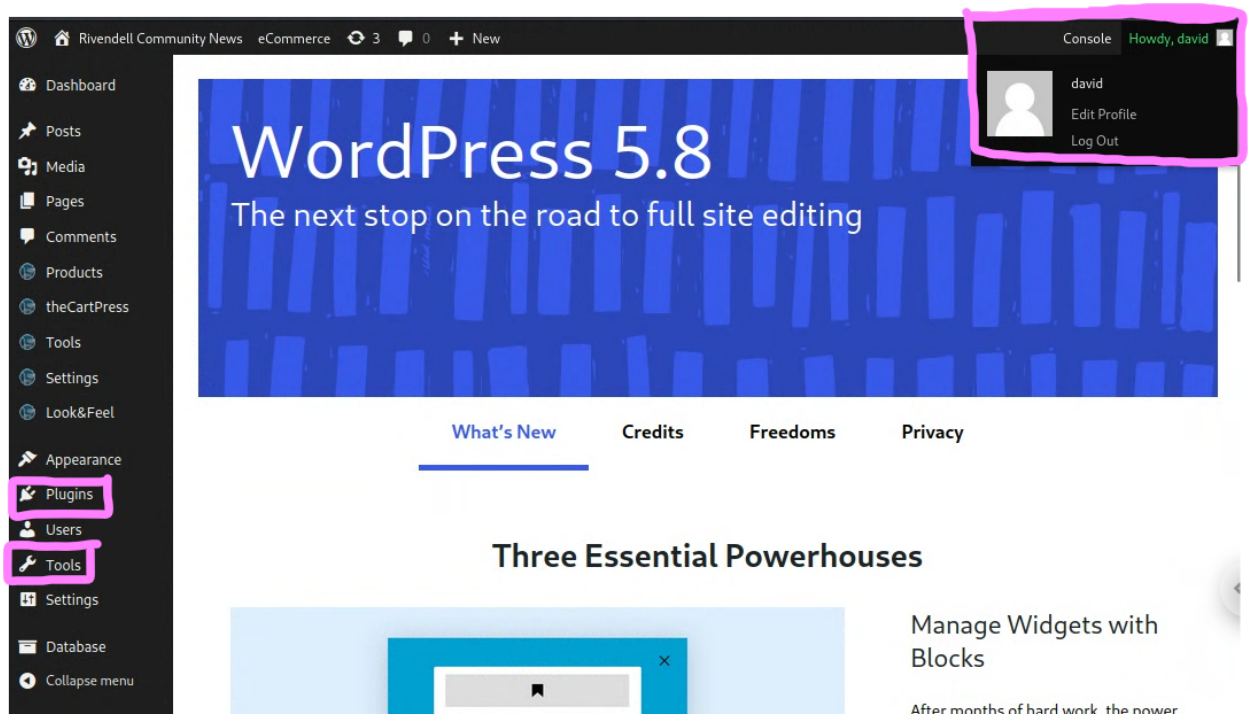
Webpage:

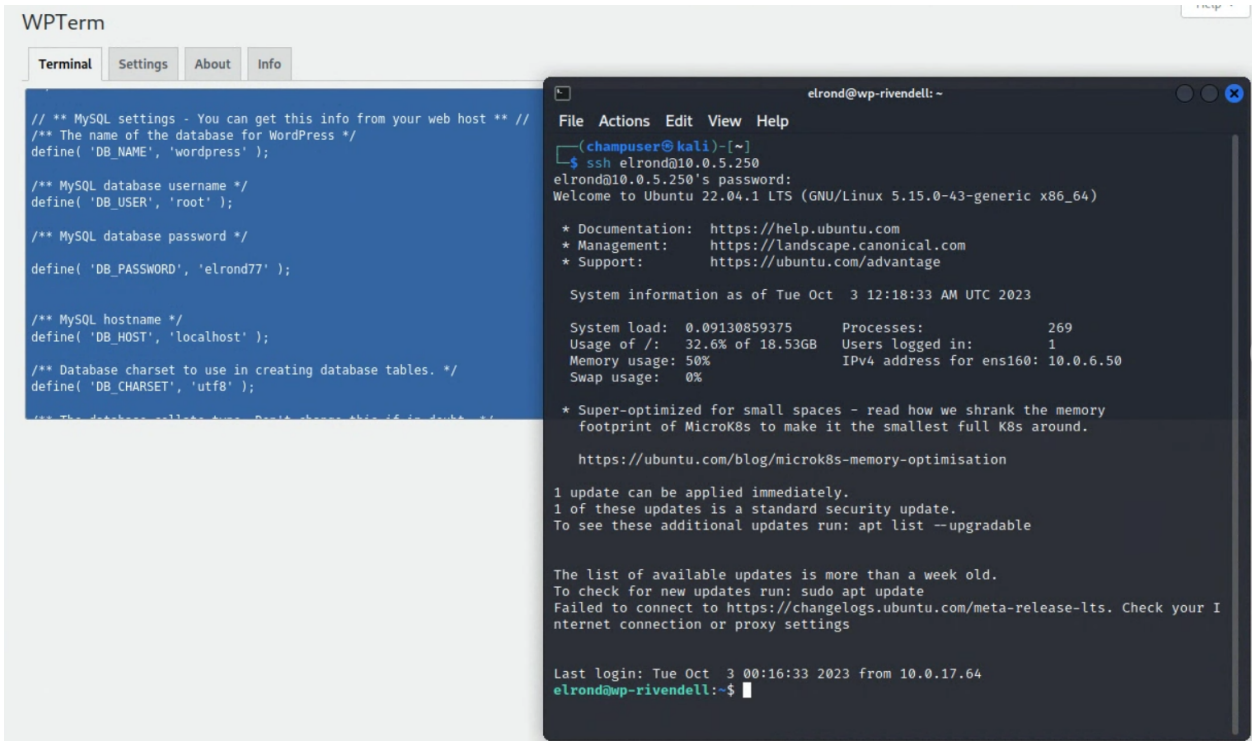


Utilizing the outdated versions of a wordpress plug in, we were able to run an exploit to create an admin user.



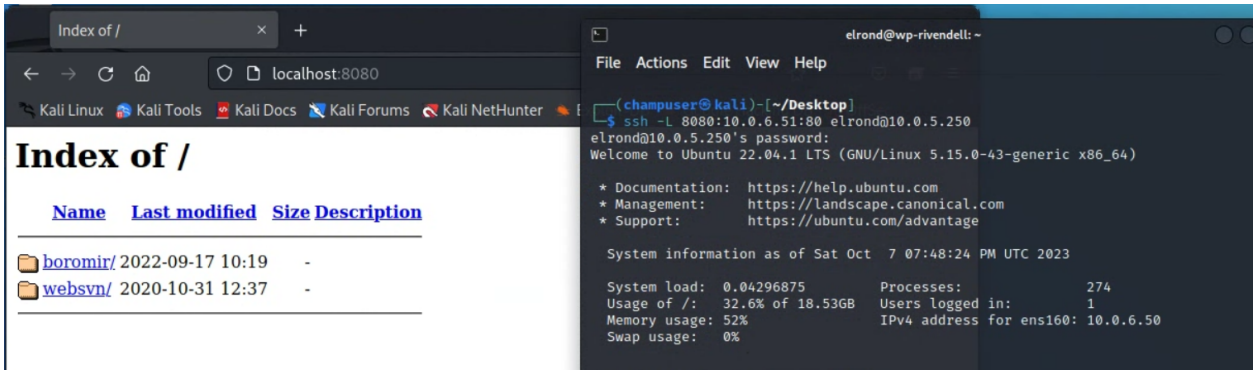
This admin account grants access to the backend of WordPress and with this, a plugin was installed to create a shell within the wordpress website, and gather credentials.



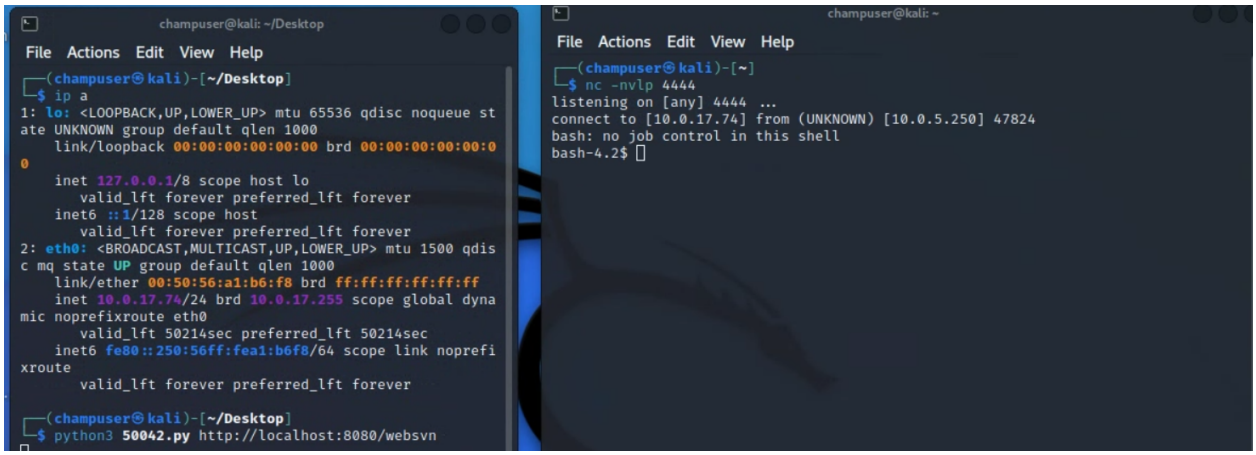

WPTerm
Terminal
Settings
About
Info
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');
/** MySQL database username */
define('DB_USER', 'root');
/** MySQL database password */
define('DB_PASSWORD', 'elrond77');
/** MySQL hostname */
define('DB_HOST', 'localhost');
/** Database charset to use in creating database tables. */
define('DB_CHARSET', 'utf8');
elrond@wp-rivendell: ~
File Actions Edit View Help
(champuser㉿kali)-[~]
$ ssh elrond@10.0.5.250
elrond@10.0.5.250's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-43-generic x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
System information as of Tue Oct 3 12:18:33 AM UTC 2023
System load: 0.09130859375 Processes: 269
Usage of /: 32.6% of 18.53GB Users logged in: 1
Memory usage: 50% IPv4 address for ens160: 10.0.6.50
Swap usage: 0%
* Super-optimized for small spaces - read how we shrank the memory
footprint of MicroK8s to make it the smallest full K8s around.
https://ubuntu.com/blog/microk8s-memory-optimisation
1 update can be applied immediately.
1 of these updates is a standard security update.
To see these additional updates run: apt list --upgradable
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your I
nternet connection or proxy settings
Last login: Tue Oct 3 00:16:33 2023 from 10.0.17.64
elrond@wp-rivendell:~$

Exploitation of Boromir

Target: boromir.rivendell.org // 10.0.6.51:

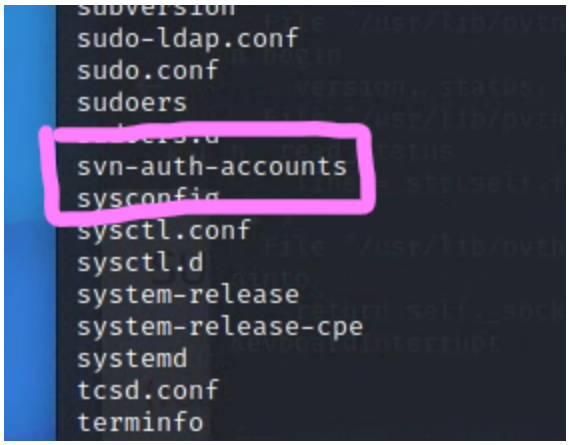


With this SSH, It was possible to create a proxy in order to access the web server from the shire network. (Curling the webpage was possible but granted limited access to any information). This webpage granted further access to the websvn page, which was vulnerable to this exploit that creates a reverse shell on the boromir server.



After further exploration of this shell, there was a file labeled svn-auth-accounts which included credentials to boromir's account. This file was not protected and the hash was able to be cracked using hashcat as well as the rockyou.txt wordlist.

Password Hash:




```
bash-4.2$ cat /etc/svn-auth-accounts
cat /etc/svn-auth-accounts
boromir:$apr1$/dPEVRIP$33jd0o1KAzXVVJaSPDwCV/
bash-4.2$
```

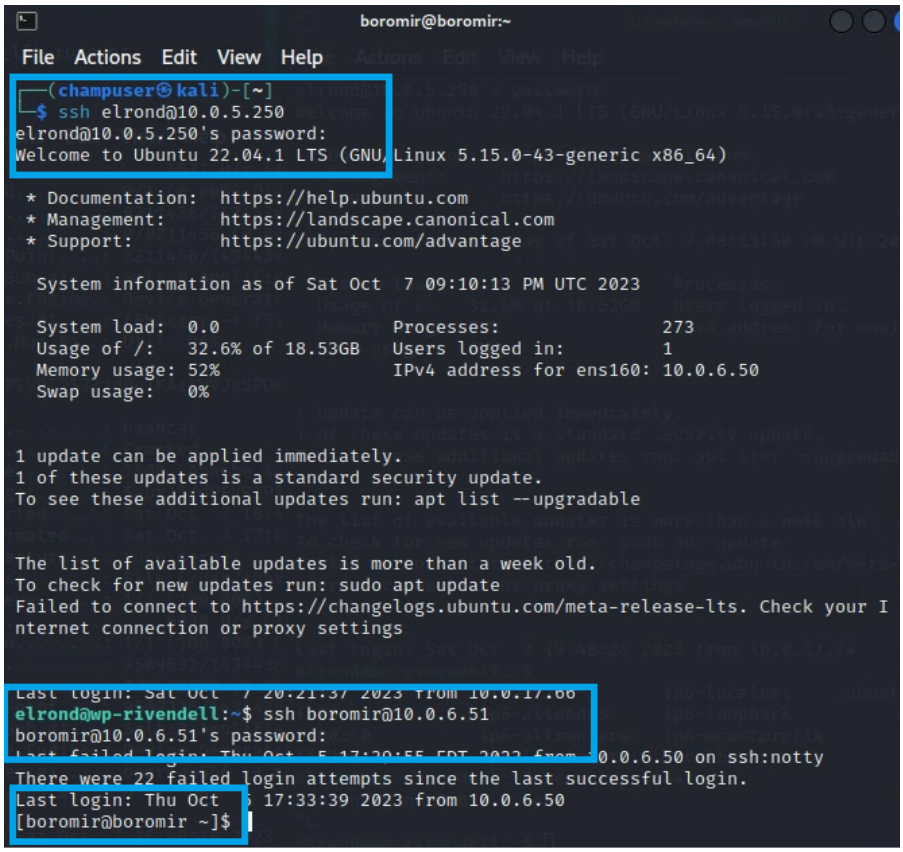
Hashcat:

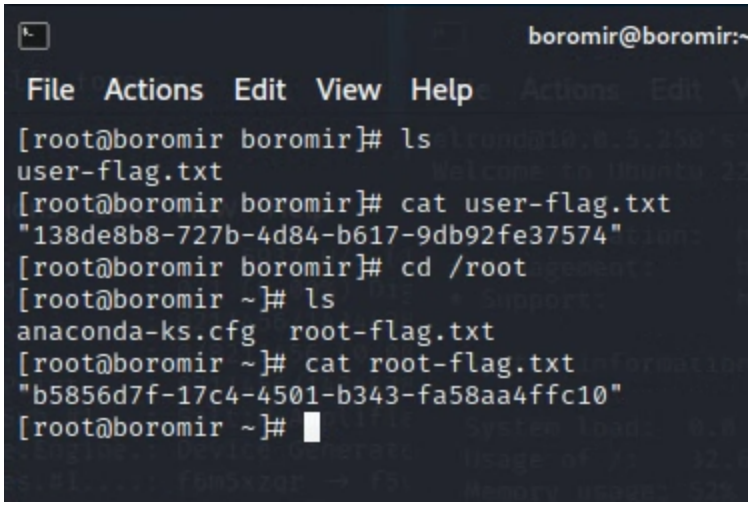
```
┌──(champuser㉿kali)-[~]
└─$ hashcat boromir.txt /usr/share/wordlists/rockyou.txt.gz
```

```
Candidates.#1....: f6m5xzqr → f5v4k6
Hardware.Mon.#1..: Util: 96%

$apr1$/dPEVRIP$33jd0o1KAzXVVJaSPDwCV/:boromir1984

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 1600 (Apache $apr1$ MD5, md5apr1, MD5 (APR))
```

Privilege Escalation:

Due to the sharing of passwords already noted in Elrond's SSH password and the DB_PASSWORD, once the password was cracked, the first vulnerability to check was for improperly configured root credentials. This had granted immediate access to the root user.





Gathering both of these flags with root credentials was a standard procedure.