Meta walkthrough

Index

List of pictures

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:

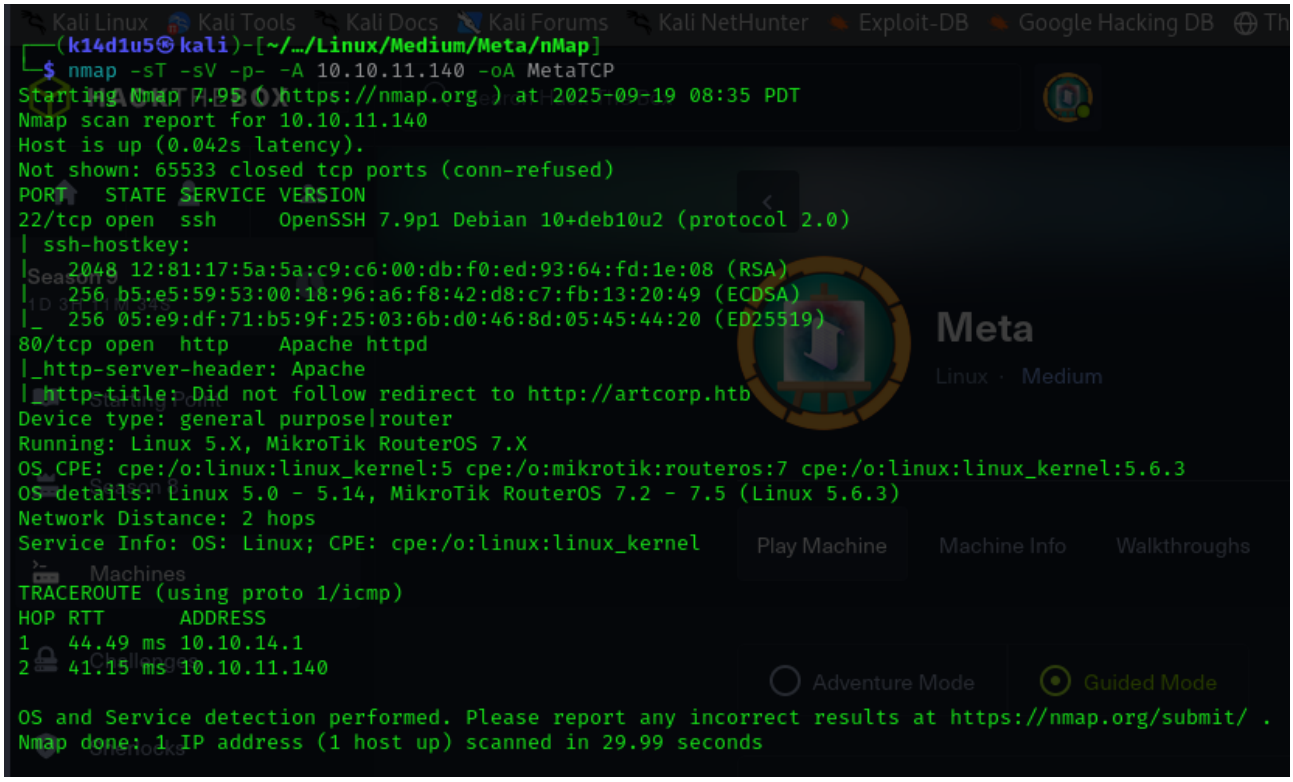

Figure 1 - nMap scan results

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 vey 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:

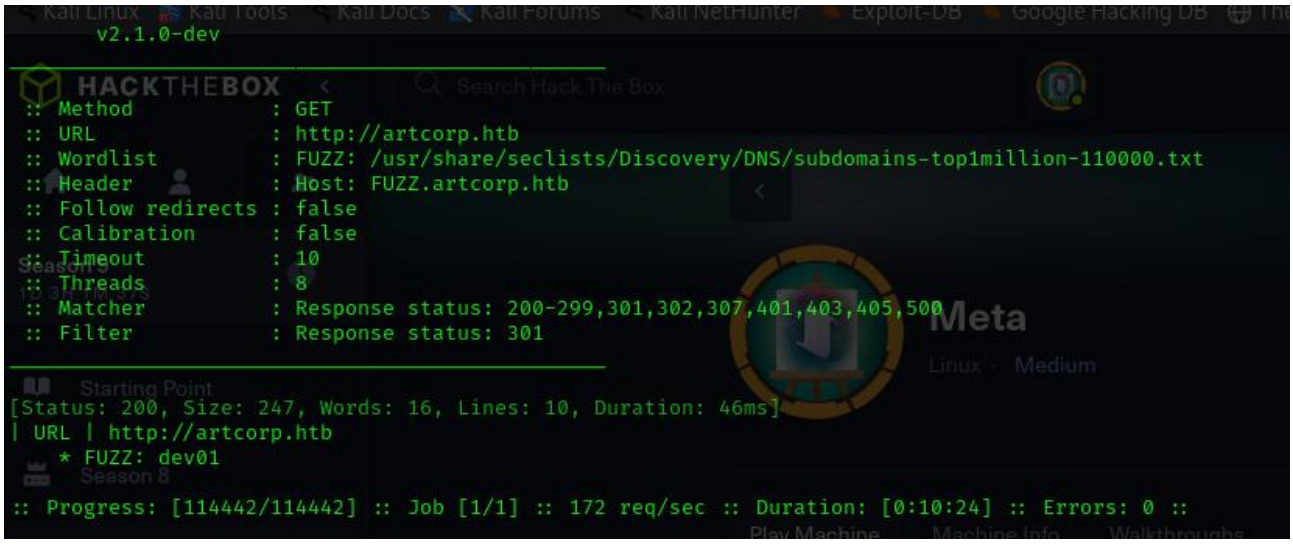

Figure 2 - Subdomain found

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:

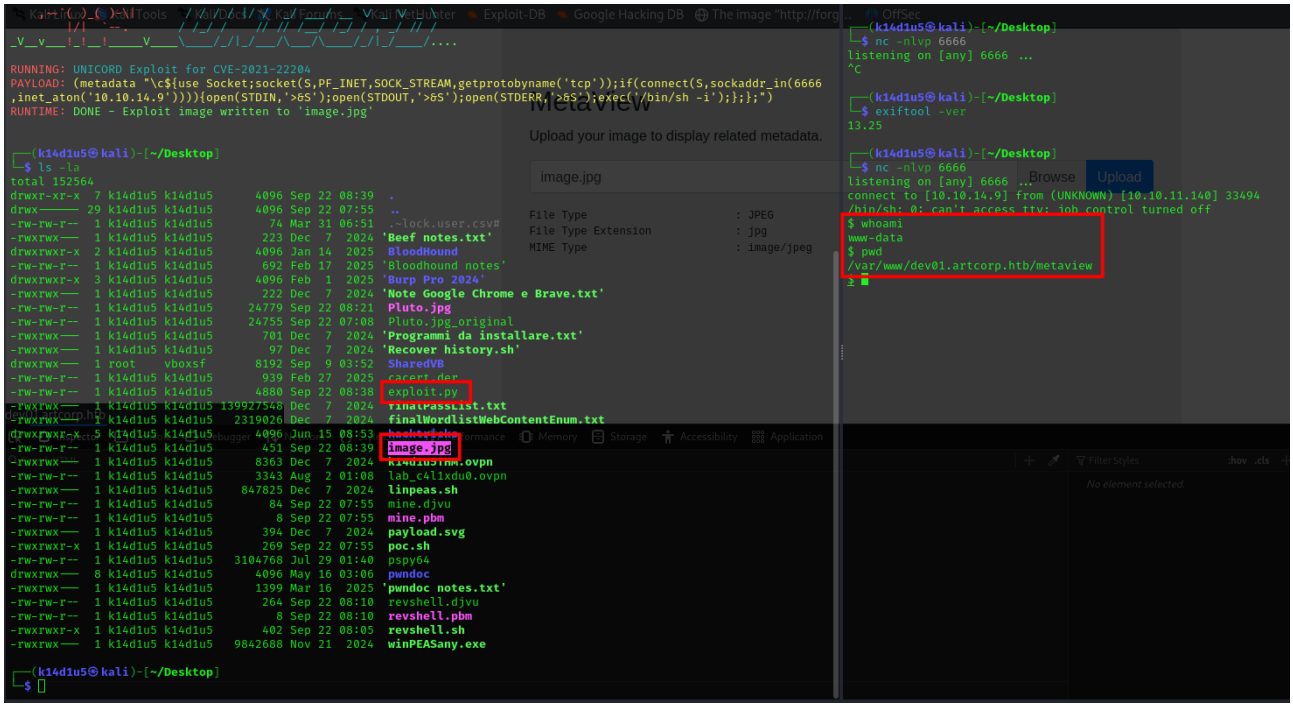

Figure 3 - Shell as www-data user

At this point, I need to find a way to became 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:

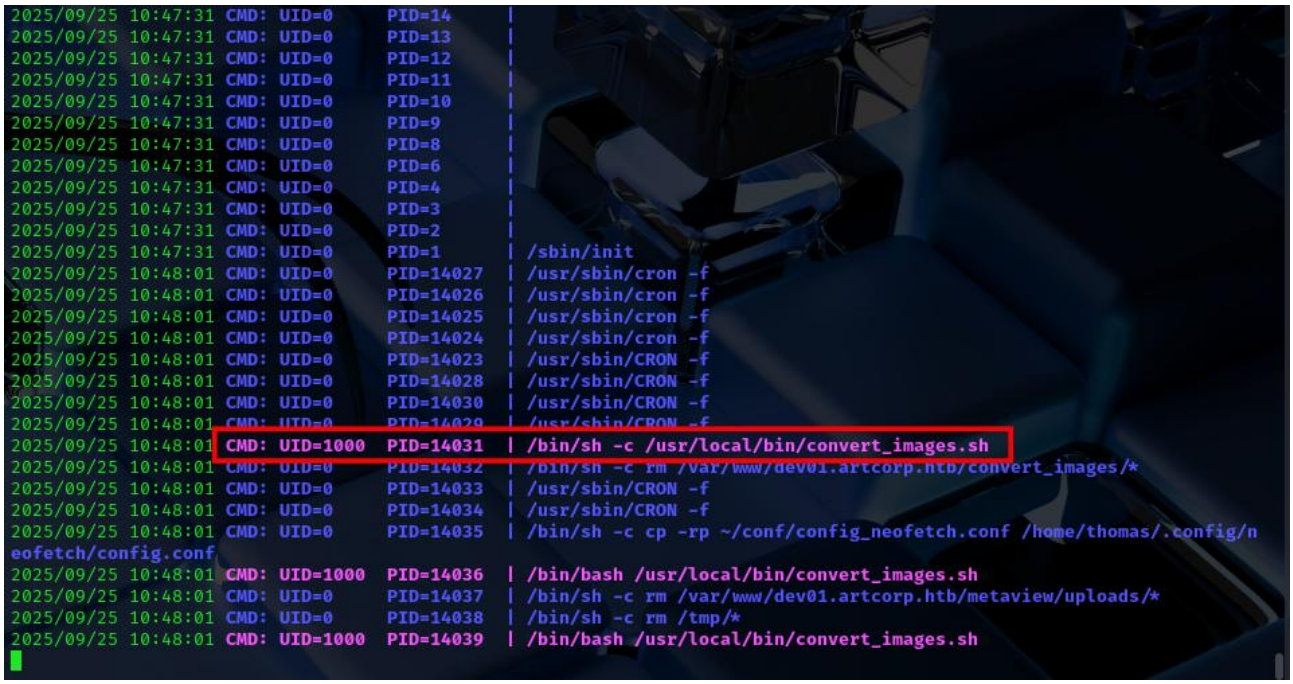

Figure 4 - Scheduled script

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:

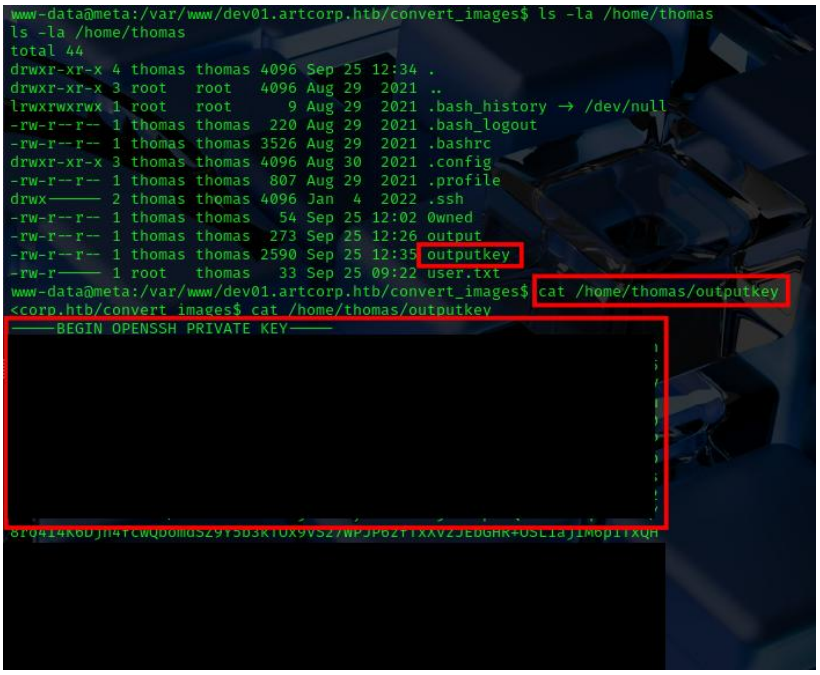

Figure 5 - Copying the Thomas private SSH keys

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:

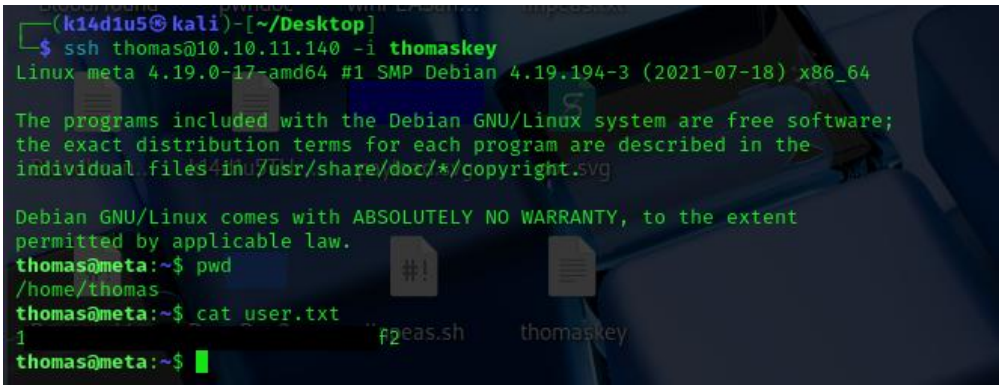

Figure 6 - User flag

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:

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:

Figure 8 - Exploitation

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

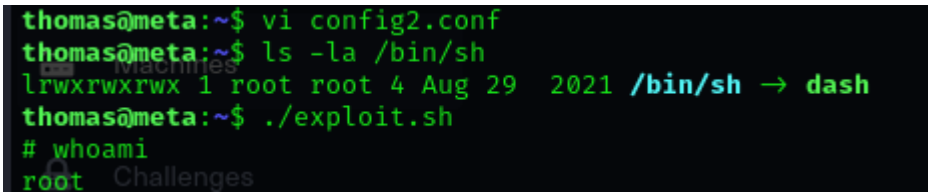

Figure 9 - Root shell

Finally, I retrieved the root shell:

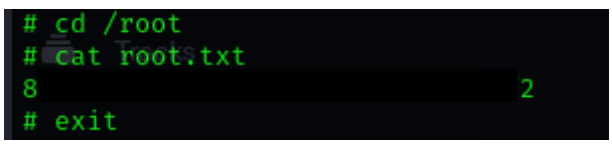

Figure 10 - Root flag

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.