DFIR PRACTICE

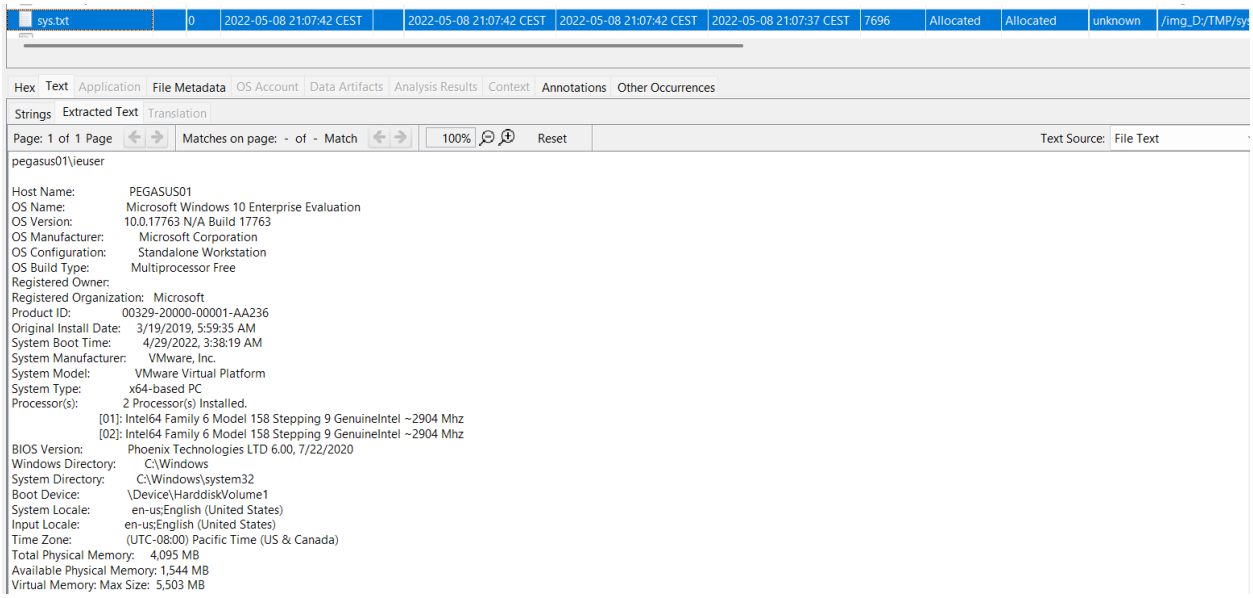
Initially, I conducted an analysis of the Windows 10 machine using tools provided throughout the course. Due to suspicions that the user was extracting company information from the machine — along with an alert from the monitoring team indicating unusual behavior — I investigated. My task was to analyze the system and determine whether there was evidence supporting these concerns.

The following is a list of the tools used, and the option chosen in each case to obtain the desired result. Sometimes I used up to three or four tools that gave me the same evidence, but only one of them is shown in this report.

- FTK Imager
- Registry Explorer/ MFTExplorer
- MFTECmd
- Anaconda / Impacket
- Hashcat
- John The Ripper
- Autopsy
-

After registering on the site ctf.sancastell.me, I found the option to download the machine to investigate **[Win10_PC001.mvdk]** and where I found a series of challenges to answer.

The screenshots and the corresponding flags for each challenge are explained below. I would like to reiterate that, in many cases, the results can be obtained using different tools and, at times, from different parts of the virtual machine image. First, general information about the Windows 10 machine to be analyzed is presented.



File Hash

As an analyst, the first step I took was to obtain the SHA-256 hash of the evidence. This procedure needed to be carried out immediately after downloading the image; otherwise, the SHA-256 hash would change, making it impossible to reproduce the same result and requiring the image to be downloaded again.

```
PS C:\WINDOWS\system32> Get-FileHash -Path "c:\Users\rabri\Downloads\Win10_PC001.vmdk" -Algorithm SHA256

Algorithm       Hash                                                                   Path
---------       ----                                                                   ----
SHA256          4446E9C42345A32FA78A8CE20834FAA047A3B161EBA986F894D2230FCF6B0CBE       C:\Users\rabri\Download
```

4446E9C42345A32FA78A8CE20834FAA047A3B161EBA986F894D2230FCF6B0CBE

Machine Name

Here are the steps I followed to obtain the name of the machine I analyzed. First, I ran the Registry Explorer tool as an administrator to gain access to the SYSTEM subdirectory. In Registry Explorer, I used the "Load Hive" option and selected the following file:

D:\Windows\System32\config\SYSTEM
Follow the next path, and it takes you directly to the file name which is the **Data** column.
SYSTEM
└── CurrentControlSet
  └── Control
    └── ComputerName
      └── ComputerName
In the right window, look for the ComputerName value.
    └── ComputerName
      └── ComputerName

Drag a column header here to group by that column

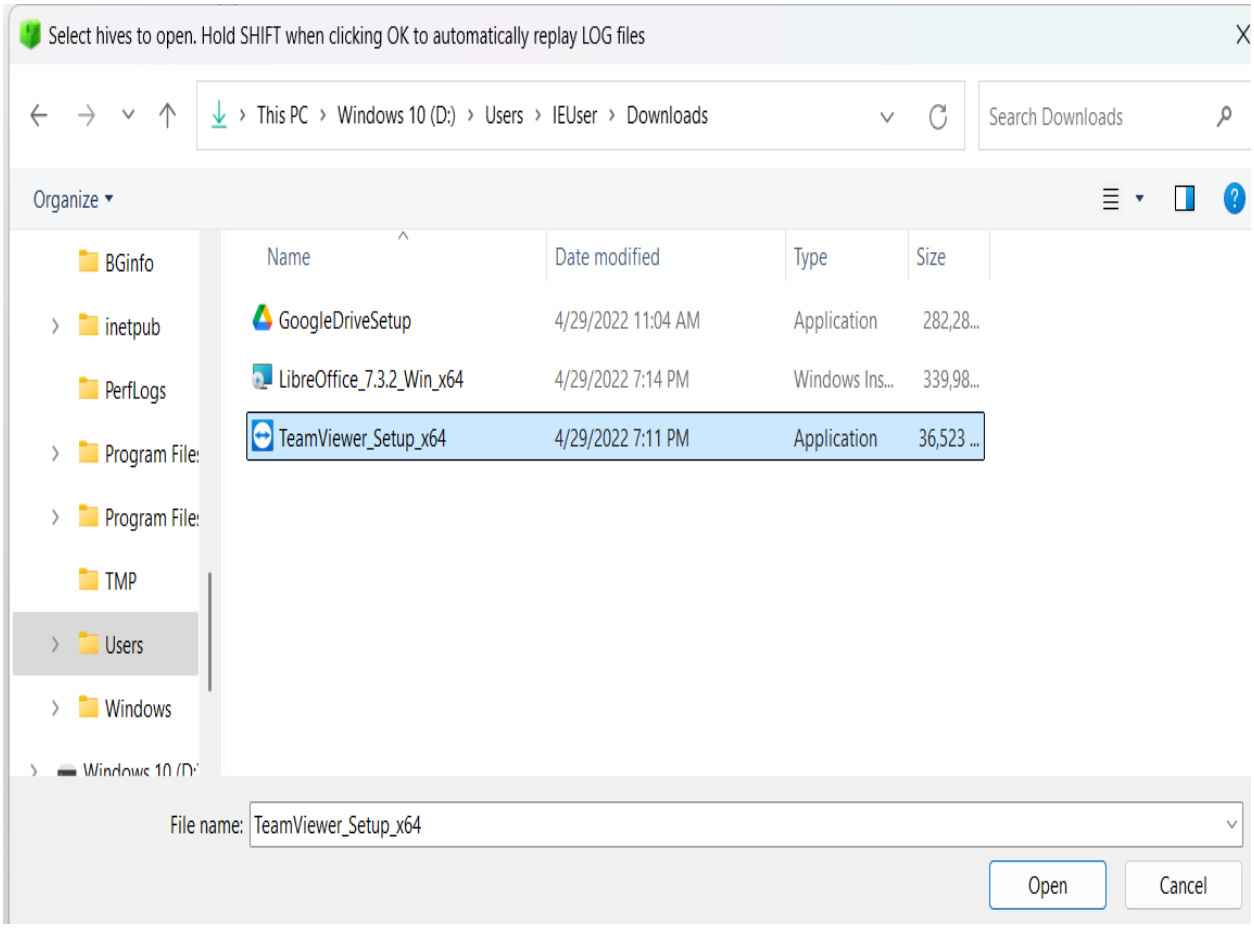
	Value Name	Value Type	Data	Value Slack	Is Deleted	Data Record Reallocated
▼	abc	abc	abc	abc	■	■
▶	(default)	RegSz	mnmsrvc	DC-00-00-00	☐	☐
	ComputerName	RegSz	PEGASUS01	30-00-00-00-53-0...	☐	☐

Machine Name: PEGASUS01

Remote Control File Download

Locating the name of the .exe file of a remote-control program downloaded by the user in the folder:

D:/Users/IEUser/Downloads/ as shown in the Screenshot below.



File Name: TeamViewer_Setup_x64.exe

Date of Remote-Control Software Download

By reviewing the information shown above, we can identify the date when the user downloaded the remote-control executable "TeamViewer_Setup_x64.exe."

Date Format: yyyy-mm-dd (EX: 2020-12-01) → **2022-04-29**

Malicious Files

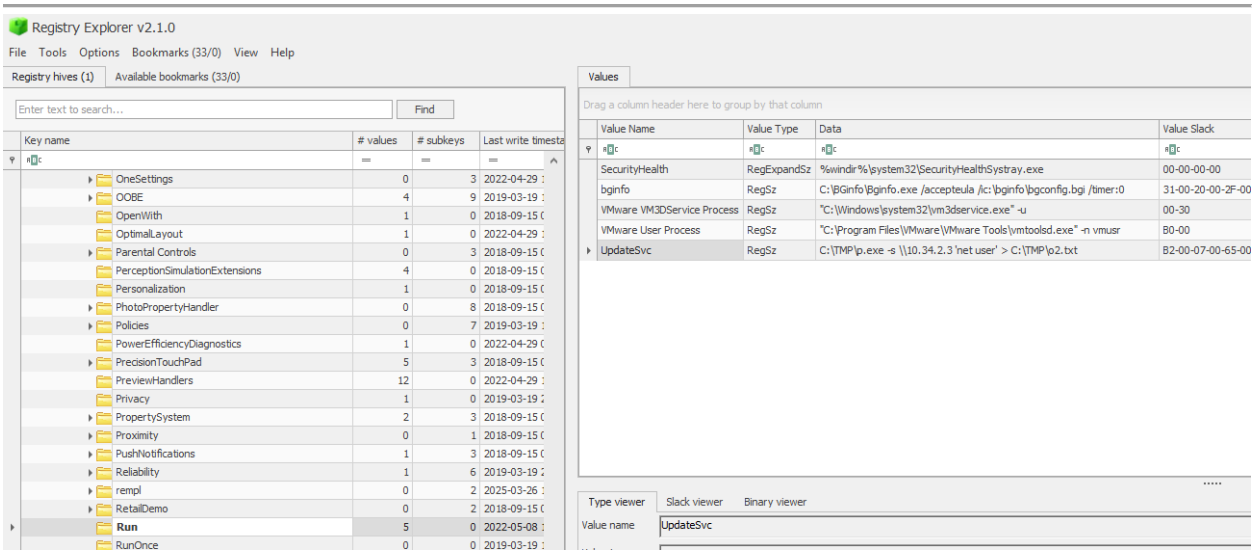
Several malicious files were found on the machine.

In which folder (only the folder name) are these files located?

Folder Name: **Temp**

Here is a summary of the steps I took to obtain the answer above, along with additional findings.
Using the Registry Explorer tool with the hive loaded, I searched for programs with suspicious names or located in unusual directories, such as:

C:\Users\Public\
C:\Windows\Temp\
C:\Users\Usuario\AppData\

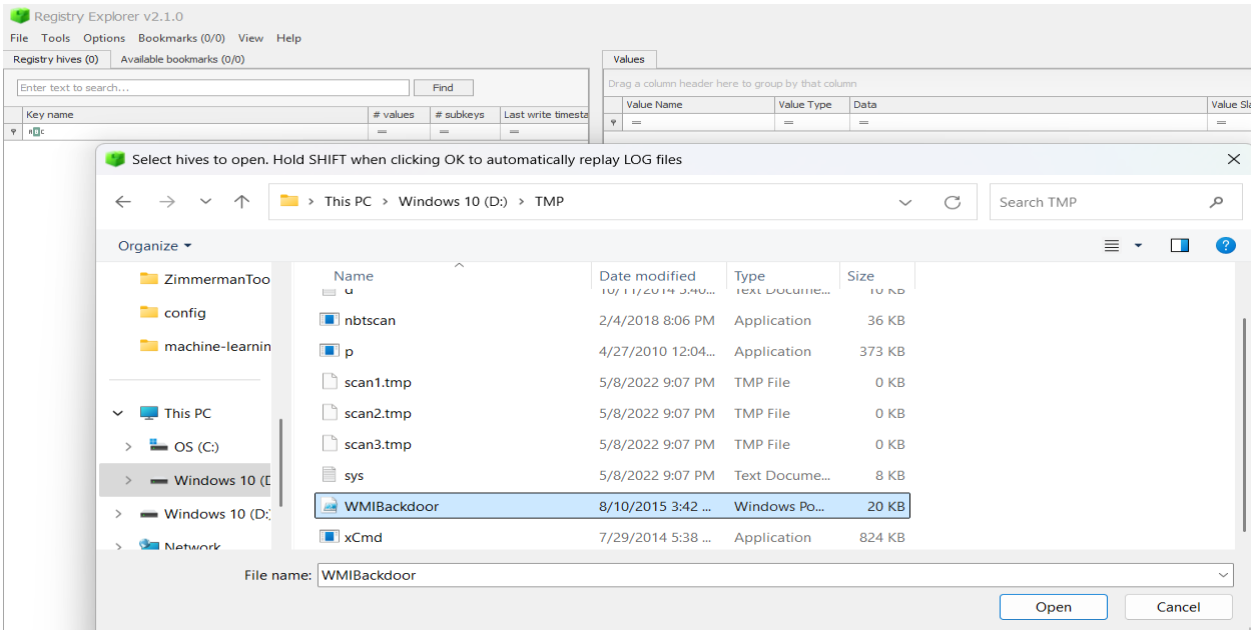


🔎 Analysis of the suspicious key:
C:\TMP\p.exe -s \\10.34.2.3 'net user' > C:\TMP\o2.txt
Reasons why it's suspicious:

1. Executable in a suspicious folder (C:\TMP\p.exe):
   - C:\TMP\ is not a standard location for system files.
   - An executable file (p.exe) in this folder is suspicious.

2. Use of a remote command (-s \\10.34.2.3 'net user'):
   - p.exe is running with parameters including -s, suggesting an automated or remote action.
   - \\10.34.2.3 indicates a possible connection to another machine in the network.
3. Command "net user" redirected to a file: (> C:\TMP\o2.txt):
   - net user is used to list user accounts in the system.
   - The output of the command is saved in o2.txt, which may indicate credential collection or system information.

Another finding in d:\TMP

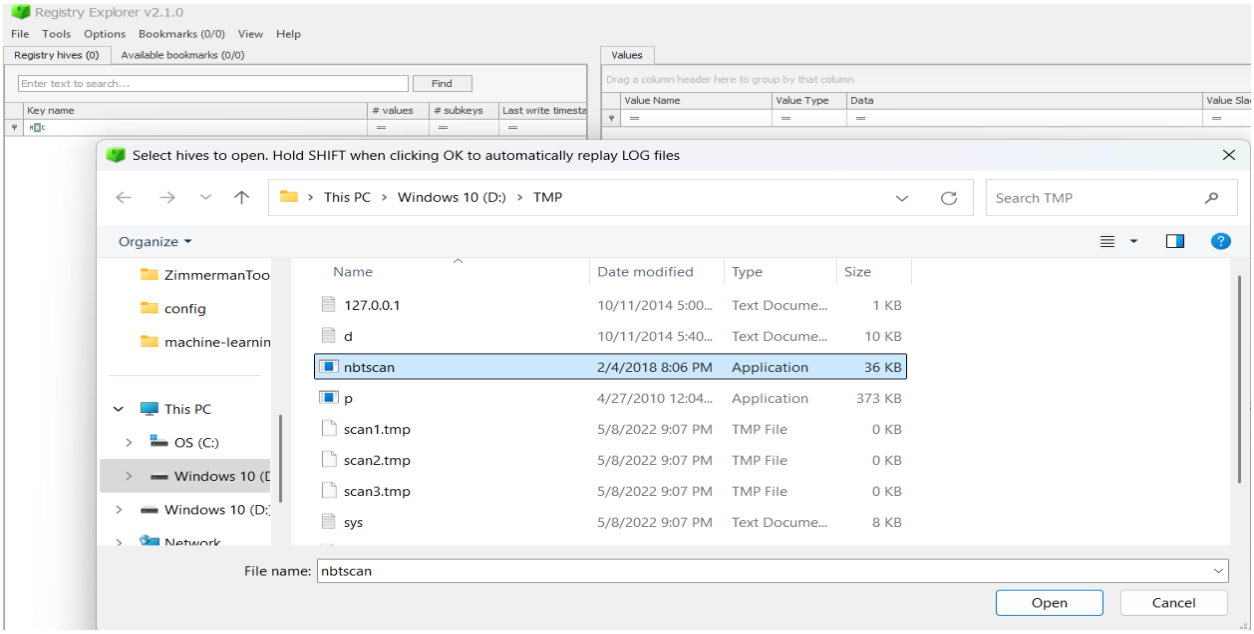


The file **wbmibackdoor.ps1**, a 20 KB Windows PowerShell script (.ps1), is highly suspicious. Finding a PowerShell script with the name **wbmibackdoor.ps1** in the **TMP** folder is a strong indicator of malicious activity. The term "backdoor" in the filename suggests that it may be designed for remote access or persistence on the system.

What is the significance of this finding?

1 PowerShell is a tool commonly used by attackers because it allows commands to be executed without generating detectable files on disk.
2 The name “wbmibackdoor” suggests the use of WMI (Windows Management Instrumentation), which could indicate a method of persistence or remote execution.
3. The 20 KB size suggests that it has extensive code, probably for executing commands in the background.

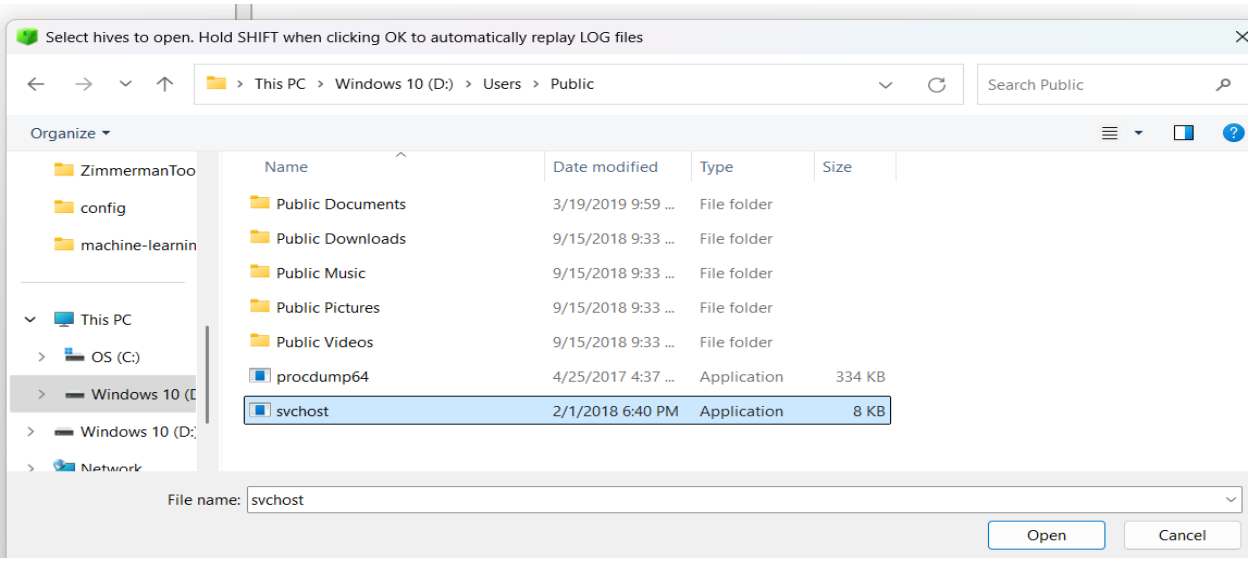
Another dangerous file in D:\TMP



nbtscan in D:\TMP\ is malicious.
My antivirus casually detected it which confirms that this file is suspicious. The name “nbtscan” suggests that it could be a network scanner, potentially used by an attacker to map devices on the network and find vulnerabilities.
🔎 Why is it dangerous?

1. "nbtscan" sounds like tools like nbtscan, which is used to scan NetBIOS networks (computer names on a Windows network).
2. The file is in D:\TMP\, a temporary folder where attackers often leave malware before executing or deleting it.
3. The antivirus flagged it as malware, indicating that it likely has a recognized signature associated with a known malicious tool.

Another possibly dangerous file [svchost].

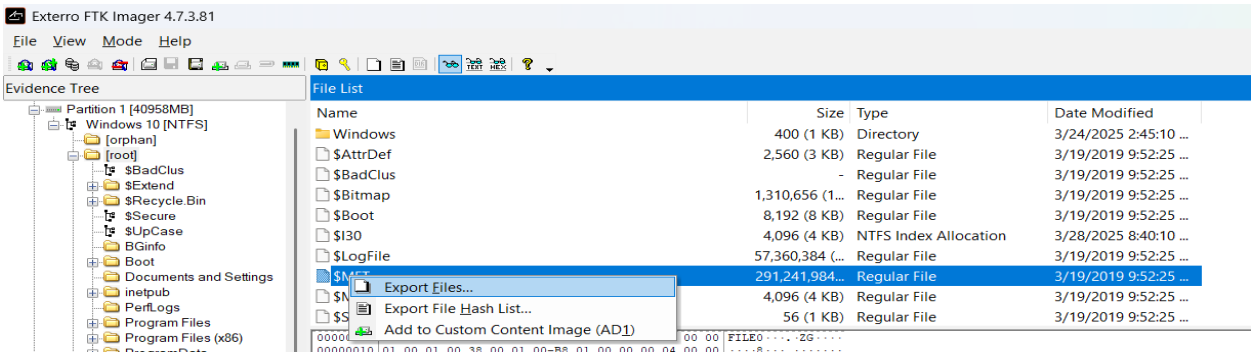


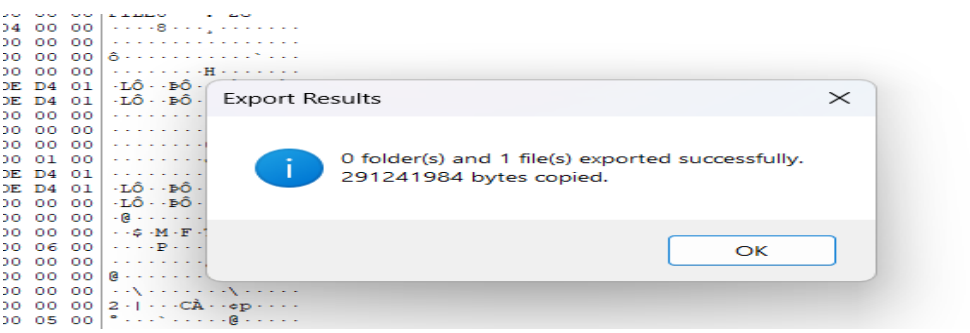
The file svhost.exe in D:\Users\Public\ is highly suspicious. The legitimate system file called svchost.exe is always located at: C:\Windows\System32\svchost.exe.

This type of file left in a publicly accessible folder is typically where attackers leave malware to facilitate its execution. The size of 8 kb suggests that it could be a dropper.

Deleted Files

It was suspected that a deleted zip file existed. I was able to find the name of this file in two different ways: directly through Autopsy, and by exporting the MFT file using FTK Imager and analyzing it with Eric Zimmerman's MFTECmd tool. I then opened the results in Microsoft Excel and searched for zip files. The file's distinctive name served as the clue that it should be used as the flag for this CTF challenge.



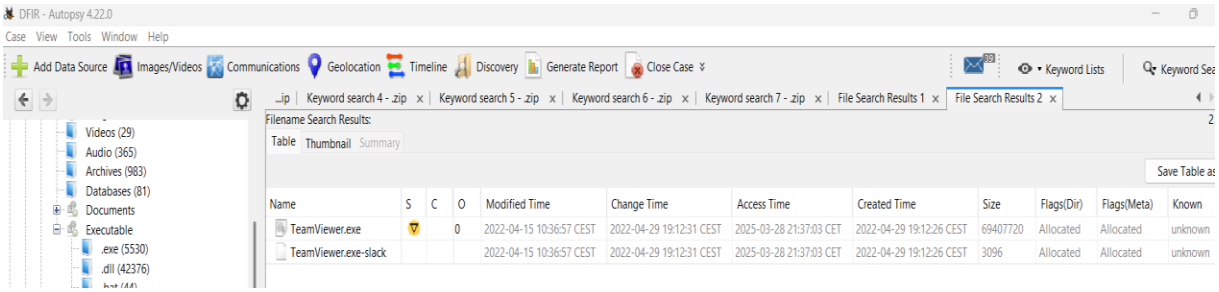


psy_Deleted_files.csv	Autopsy_Deleted_files	A1315	amd64_microsoft-windows-zipfldr_31bf3856ad364e35_10.0.17763.107_none_ecfc1396bad03aba.manifest
psy_Deleted_files.csv	Autopsy_Deleted_files	J1315	/img_D:/Windows/servicing/LCU/Package_for_RollupFix~31bf3856ad364e35~amd64~~17763.379.1.11/amd64_microsoft-windo
psy_Deleted_files.csv	Autopsy_Deleted_files	A2139	SDelete.zip
psy_Deleted_files.csv	Autopsy_Deleted_files	J2139	/img_D:/Windows/Temp/SDelete.zip
psy_Deleted_files.csv	Autopsy_Deleted_files	N2139	zip
psy_Deleted_files.csv	Autopsy_Deleted_files	A4993	amd64_microsoft-windows-zipfldr.resources_31bf3856ad364e35_10.0.17763.1_en-us_043da35bd675bef6.manifest
psy_Deleted_files.csv	Autopsy_Deleted_files	J4993	/img_D:/Windows/WinSxS/Manifests/amd64_microsoft-windows-zipfldr.resources_31bf3856ad364e35_10.0.17763.1_en-us_043d
psy_Deleted_files.csv	Autopsy_Deleted_files	A6498	cosas.zip

Date of Execution of Remote-Control Program

We know that the TeamViewer program has been run on the computer, and we can determine the date it was executed. Format: dd/mm/yyyy.

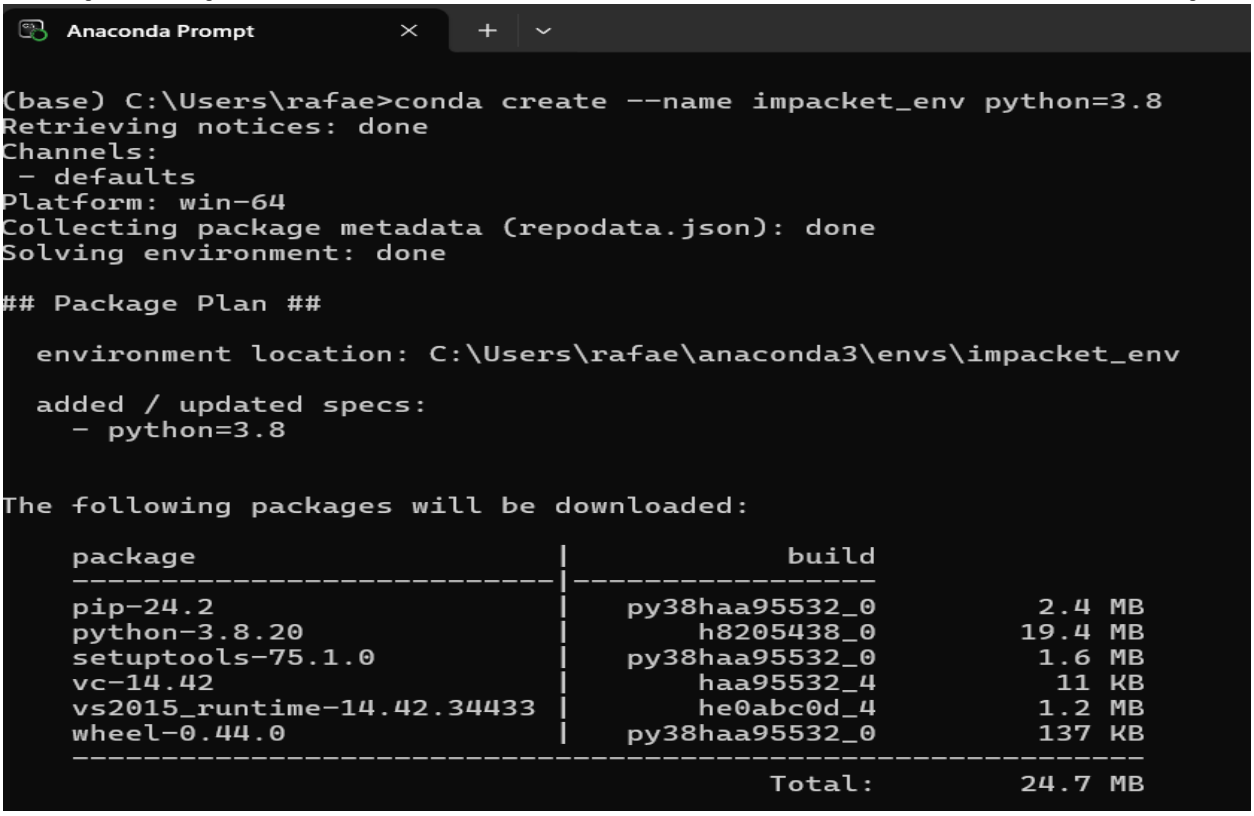
Date: **29/04/2022** [Appears in the following screenshot]



Weak Passwords

There are suspicions that the password for the user **IEUser** is weak, which may have allowed the attacker to gain access. I initially attempted to use the Mimikatz tool to obtain the password, but I chose to use my host for this practice. However, I found it too labor-intensive to install Mimikatz due to the multiple antivirus programs on my host, and I decided it would be better to install it at a later time. Instead, I opted for an alternative approach, and below are the steps I followed.

El primer paso fue crear un ambiente dentro de Anaconda de impacket.



```
Anaconda Prompt
  pip                 pkgs/main/win-64::pip-24.2-py38haa95532_0
  python              pkgs/main/win-64::python-3.8.20-h8205438_0
  setuptools          pkgs/main/win-64::setuptools-75.1.0-py38haa95532_0
  sqlite              pkgs/main/win-64::sqlite-3.45.3-h2bbff1b_0
  vc                  pkgs/main/win-64::vc-14.42-haa95532_4
  vs2015_runtime      pkgs/main/win-64::vs2015_runtime-14.42.34433-he0abc0d_4
  wheel               pkgs/main/win-64::wheel-0.44.0-py38haa95532_0


Proceed ([y]/n)? conda activate impacket_env
Invalid choice: conda activate impacket_env
Proceed ([y]/n)?


Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate impacket_env
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```

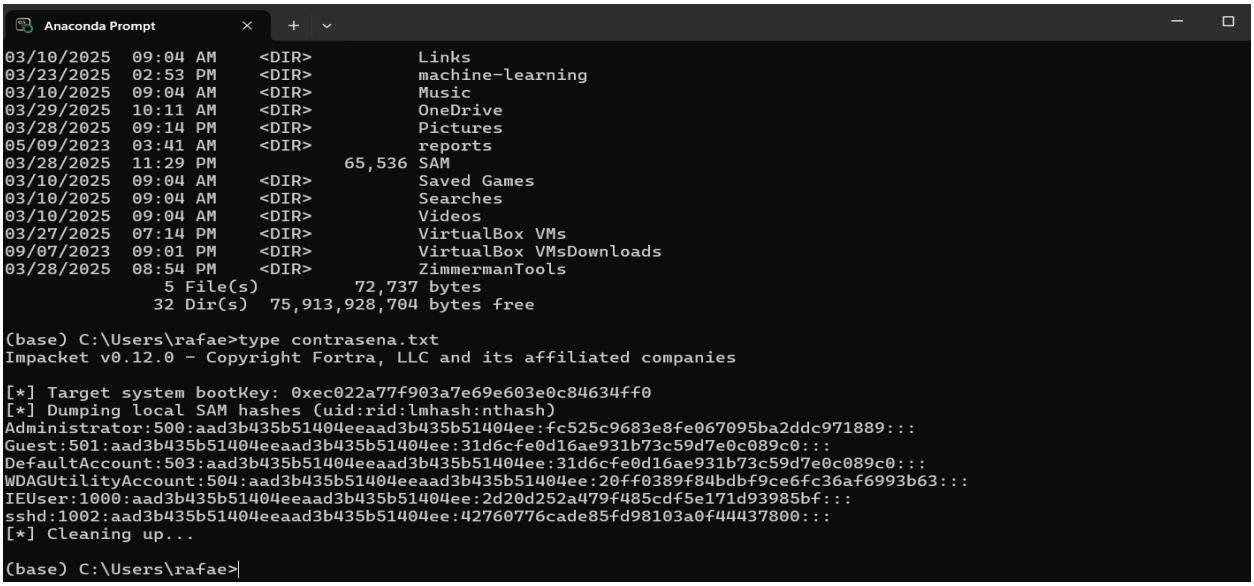
Using the secretsdump.py command, as shown below, I obtained the hashes and selected the one needed to run **John the Ripper**. The highlighted hash below corresponds to the password I aimed to crack.

```
(base) c:\>cd c:\Users\rafae\

(base) c:\Users\rafae>cd Desktop

(base) c:\Users\rafae\Desktop>secretsdump.py -sam SAM -system SYSTEM LOCAL
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Target system bootKey: 0xec022a77f903a7e69e603e0c84634ff0
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:fc525c9683e8fe067095ba2ddc971889:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:20ff0389f84bdbf9ce6fc36af6993b63:::
IEUser:1000:aad3b435b51404eeaad3b435b51404ee:2d20d252a479f485cdf5e171d93985bf:::
sshd:1002:aad3b435b51404eeaad3b435b51404ee:42760776cade85fd98103a0f44437800:::
[*] Cleaning up...
```

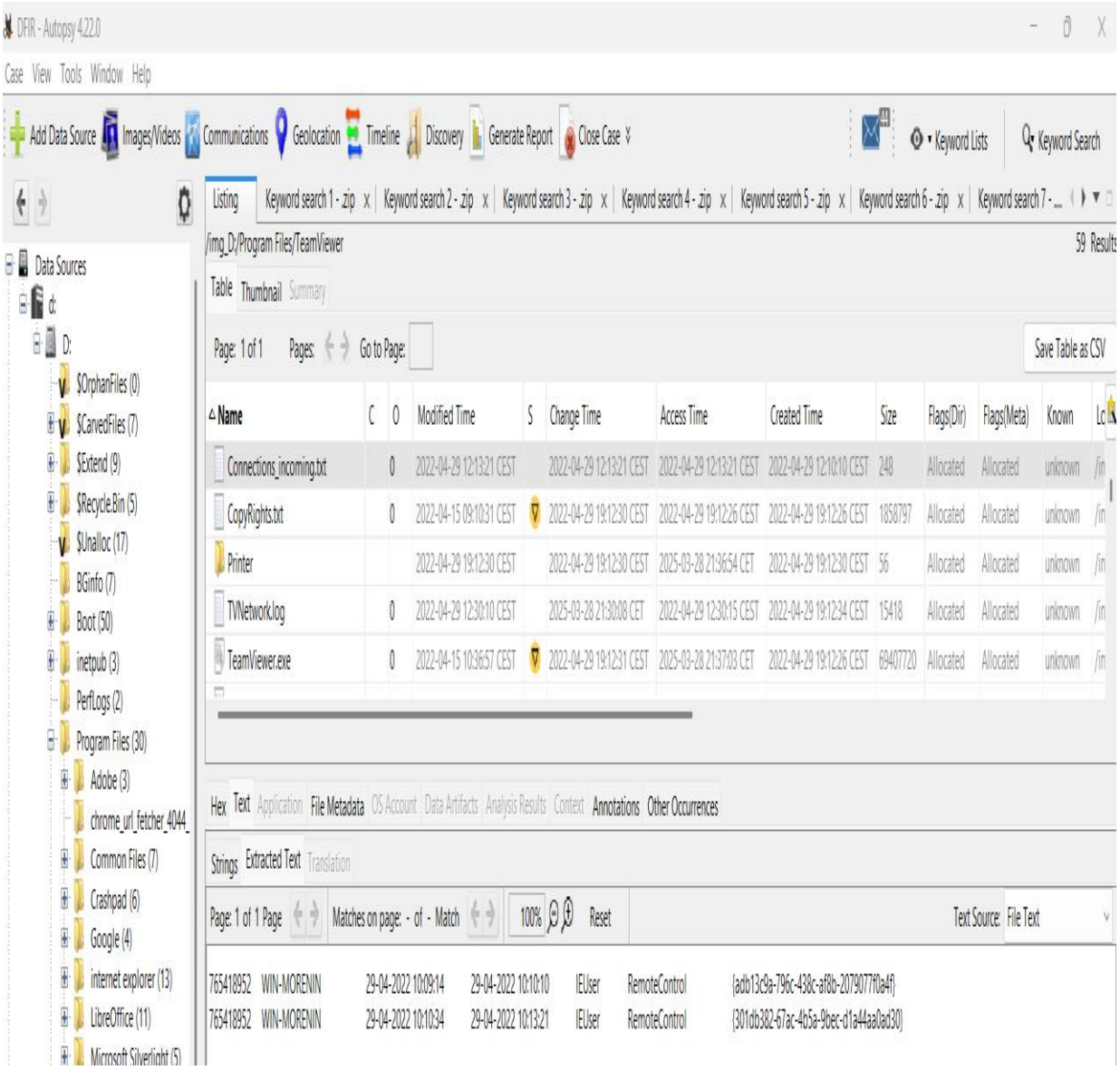


In this last screenshot you can see the password: **qwerty**

Remote Control Program Connection

There is suspicion that the attacker has connected to the computer using a remote-control program. We can identify the ID from which the attacker connected. To locate this ID, I used Autopsy, and the screenshot showing this ID is provided below.

ID = 765418952



RDP Connection

Suspicious activity has been detected on the network, you could indicate the IP from which you have connected to the machine via RDP.

Habiendo encontrado que un equipo se ha conectado desde un programa de control remoto y localizado el ID desde el que se conectó el atacante y en ese misma pagina de información que nos brinda Autopsy aparece la fecha y la hora exacta en que se realizaron esas dos conexiones pude rastrear el momento exacto con una precisión de aproximadamente dos minutos entre ambos eventos.

The IP that was utilized: **192.168.183.134**

5302,5302,2022-04-29 10:08:16.2309769,4648,LogAlways,Microsoft-Windows-Security-Auditing,Security,640,4384,PEGASUS01,66,,A logon was attempted using explicit credentials,PEGASUS01\IEUser,**192.168.183.134:445**,Target: PEGASUS01\user1,TargetServerName: dev,PID: 0x4,TargetInfo: dev,,,,False,C:\Users\rafae\Documents\Security.evtx,Audit success,0,"{""EventData"":{""Data"":[{""@Name"":""SubjectUserSid"",""#text"":""S-1-5-21-321011808-3761883066-353627080-1000""},{""@Name"":""SubjectUserName"",""#text"":""IEUser""},{""@Name"":""SubjectDomainName"",""#text"":""PEGASUS01""},{""@Name"":""SubjectLogonId"",""#text"":""0x707C1""},{""@Name"":""LogonGuid"",""#text"":""00000000-0000-0000-0000-000000000000""},{""@Name"":""TargetUserName"",""#text"":""user1""},{""@Name"":""TargetDomainName"",""#text"":""PEGASUS01""},{""@Name"":""TargetLogonGuid"",""#text"":""00000000-0000-0000-0000-000000000000""},{""@Name"":""TargetServerName"",""#text"":""dev""},{""@Name"":""TargetInfo"",""#text"":""dev""},{""@Name"":""ProcessId"",""#text"":""0x4""},{""@Name"":""ProcessName""},{""@Name"":""IpAddress"",""#text"":""192.168.183.134""},{""@Name"":""IpPort"",""#text"":""445""}]}}"

Below, I detail the steps taken to obtain the IP using the Zimmerman EvtxECmd tool.

```
C:\Users\rafae\Downloads\EvtxECmd\EvtxeCmd>.\EvtxECmd.exe -f "C:\Users\rafae\Documents\Security.evtx" --csv "C:\Users\ra
fae\Documents\evtx_output" --csvf salida_seguridad.csv --inc 4624,4648,4672
EvtxECmd version 1.5.2.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/evtx
```

```
C:\Users\rafae\Downloads\EvtxECmd\EvtxeCmd>.\EvtxECmd.exe -f "C:\Users\rafae\Documents\Security.evtx" --csv "C:\Users\ra
fae\Documents\evtx_output" --csvf salida_seguridad.csv --inc 4624,4648,4672
EvtxECmd version 1.5.2.0
```

```
CSV output will be saved to C:\Users\rafae\Documents\evtx_output\salida_seguridad.csv

Maps loaded: 453
```

Once the information was downloaded into Microsoft Excel, I searched for the relevant data, focusing on the event associated with **PEGASUS01\IEUser** and the timestamp **2022-04-29 10:08:16**, which was linked to the events where I obtained the remote access ID.

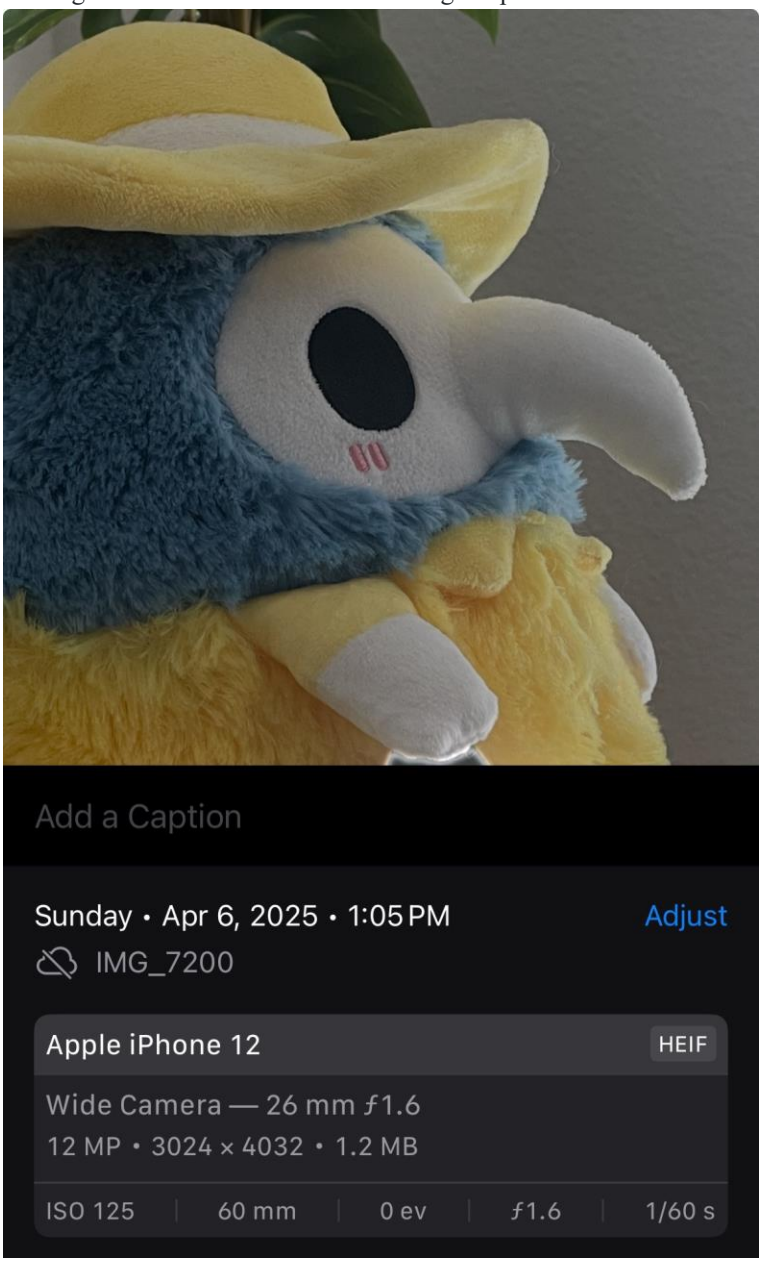
Attacker's Connection Port

As we know, there is a connection to the machine with IP 192.168.183.134 via RDP, but we suspect that there are more connections to different ports. Indicate the port on which the connection occurs.

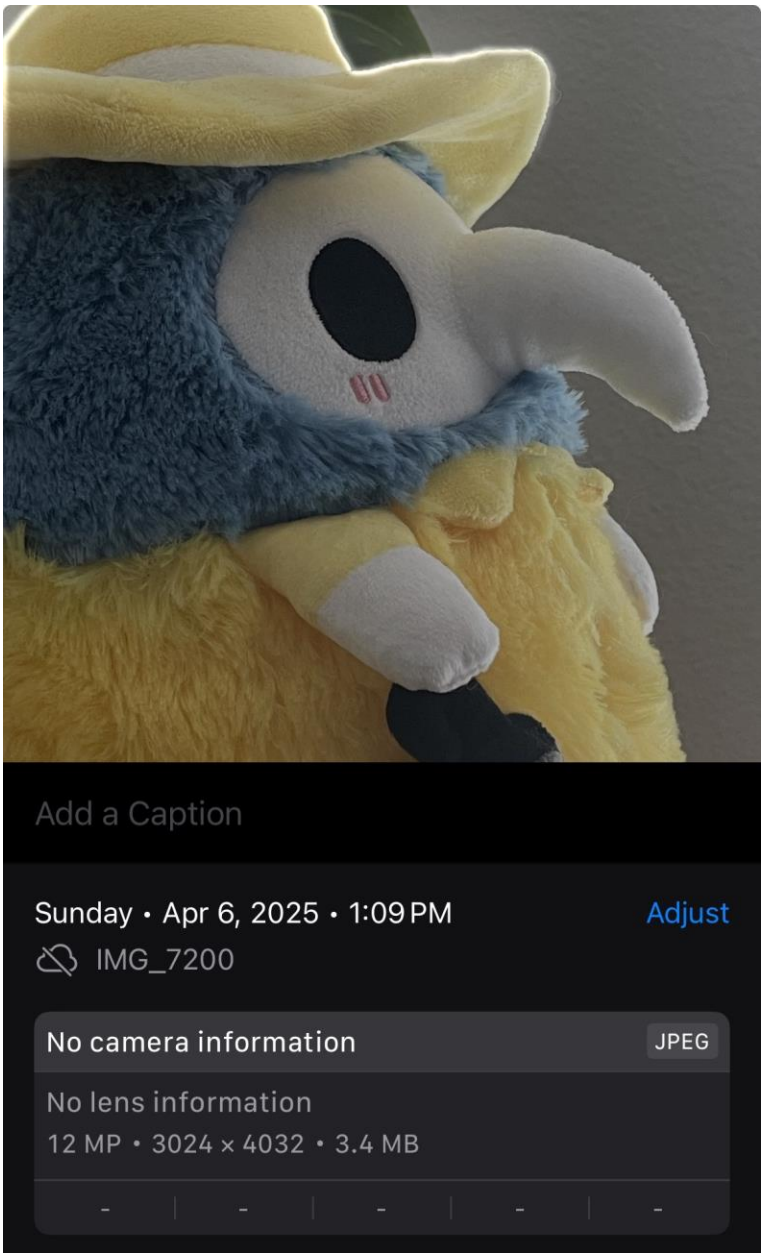
The connection port is 445 which appears in the information downloaded in the salida_seguridad.csv file and a copy of this is shown above.

Metadata

This is the original photo in question. This photo was taken on my own personal iPhone and below is a screenshot of the original metadata included with taking the photo.



After sending the image through discord, a messaging service, it goes through discord's compression/delivery process, and the metadata is changed.



You can see that the original image resolution was retained after downloading despite possible compression, but the file size has increased. Also, not only the format of the image has changed from HEIF to JPEG, but the information on the camera lens and device used to take the picture has been lost. This is important to recognize in digital forensics.