INSTITUTO SUPERIOR TÉCNICO
Departamento de Engenharia Informática
Forensics Cyber Security
MEIC / METI 2024-2025 – 1st Period

Digital Forensics Report Lab2

Group number:	43	Name	IST Number
Student 1:		Daniel Pereira	99194
Student 2:		Luana Marques	82374
Student 3:		Sofia Du	104195

1 Acquired artifacts

Name	Type	SHA-256 Value
backup_1727365201.zip	Zip	0ff91e834b3f022b57f30200623ae4f5be25187ff6d83c77768bac37570eeca1
backup_1727365801.zip	Zip	2f5c01bcba0a0bb583ecfec86e1b940b807856f3c37d5a70c5fe280785f4beab
backup_1727366402.zip	Zip	5e615972e621d6ef3f1fb471f9e21a646c17b291e6a419b5d85f589c66b8ffa5
backup_1727367001.zip	Zip	c6ebd95a9cdf32904b204d23f3845ce5d2c21a6d7b94a60291577b987c46f5d4
backup_1727367601.zip	Zip	146b96a45aeb0beeaba4ae67d0fa12cdc6d78f6849715af2bd6a8c252fa211ad
backup_1727368201.zip	Zip	618f88dee9f34c227906d0b195b54b13867909d01bcb38530508a41c9db39791
backup_1727368801.zip	Zip	a7001fa984da7117d7474e4918984594dbf7ab8e8c17d3a44baa2315be6d7da1
converter.py	Python	a95cfe59fb2fd225e3a448582baea8898d2b57004ff6e75cb1314877ce036946
createChunks.py	Python	230db42d33b9e3c3ed9e7b0b56178352fd71515d4f94464e31e6fee43c214826

doubleEncodingAndHidingInsideElf.py	Python	10f012976912da3289b62528e3523ec0e30ef93831cda89ac1d06a87807119f2
hide_pdf.py	Python	f28b8183c12aee88d7e15f98a3856904595b342477a960b55933e4158ffc2b80
cookieSteal.htm	HTML	9a65dadabe142ceb15bbe5232c00a411571e85efedb88ab3f808e5cd7c776a53
exploit.htpl	HTPL	0f9fe89cc0108f9e1b933bd8daff74dfbde17444ce737665b7f1bac7c2df9b06
exploit.py	Python	ccbfd54f6c3fe06c214aea28927a76a7f508269f76c3e5ef3692a8ad2022ae3b
lsb.pyc	PYC	0a003a31d44d8434cfdf5dea9ac646b3d2e38f9339c9a2399d99d6e4f447a681
User_Manual.pdf	PDF	c8a2641e23014cb7fafc83f14e4e448500b141cb74fe09634f3db2f8b4ec6d20
200d463649820c65548b121e4f42ad68.png (rocket´s image)	PNG	432c77c33c498674b8848a2d0d1eaedfc50decd287677f0457ff216ae5827979
ceaf1f22d9a118942a4f7ad833c36b97.png (study rooms)	PNG	108c385564aa9ccf38a393e9a9f1a0960043e711dfeebbf04a9d3e585e34e601
#thebasement.09-26.log (IRSSI chat)	Text Document	b19ce5156507851d206f7559ed67cb308d2b8b774f096296fcfee754cb98df82
syslog		21bd644a2359c064bbaae327e178930a9e83771916ab8f22cdf94ba4d4551baa
Passwords.kdbx	KeePass DB	40b6c756d4d1b4fb85cdc4c11c6e0b8f4ca5eb1019c05f51c356a4d669c7c38a
KFP7oy1K7O5.log (keylogger)	Text Document	939b7ac1963a073f727683fd7a2ae66cf0b84b00f6157c628d324328b02c53e1
K5rb9cnL0Is.log (keylogger)	Text Document	1fa61fc25578cd81146c31116327f4c7defa5282815265b8cbb5c2662b165aae
seed.txt	Text Document	0c4a848fe5e6225ad5b41f67af0d77040e2fad2bc713b7b4938569f9255e135e
Johnnymusk (cron job)		5ecfbf9a5a2785414a461b91cf908ea24850664999d6fdd8cb716a79428ae5bd
Inbox		662fc743207d8cd70ec7e5b7f063b5fdda5e8e12f1f8d14d93f9c189234e169f
history.csv	CSV file	c26900f8cc25ad9ee5e2a85d8d37d73e4c70209820c23bdf73b5260d9785d085
.bash_history		9a656b05678b07ac0f4aa7a0fd167ea528bd395893290e09244fbe0888103a44

1. Report of all findings

The initial step of this investigation involved checking the fingerprints of the artifact files, using ***sha256sum file_name***. By doing this we confirmed that our images of the hard drives had not been manipulated.

- **JohnnyDisk.img**

The first step that we did is ***mmls johnnyDisk.img.*** This command lists all the partition table within that image.

```
┌──(kali㉿daniel-win11)-[~/johnnyDisk]
└─$ mmls johnnyDisk.img
GUID Partition Table (EFI)
Offset Sector: 0
Units are in 512-byte sectors

      Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Safety Table
001:  -------   0000000000   0000002047   0000002048   Unallocated
002:  Meta      0000000001   0000000001   0000000001   GPT Header
003:  Meta      0000000002   0000000033   0000000032   Partition Table
004:  000       0000002048   0000004095   0000002048
005:  001       0000004096   0052426751   0052422656
006:  -------   0052426752   0052428799   0000002048   Unallocated
```

Figure 1: Output of mmls johnnyDisk.img

As we can see, neither slot 000 nor slot 001 has any description. So, we decided to analyze each target partition using the ***fsstat*** command. The partition that starts at sector offset 2048, we didn´t obtain any file system information. But in the root file system, on the partition that starts at sector offset 4096, we obtain the information below about the partition.

```
└─$ fsstat -o 4096 johnnyDisk.img
FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: Ext4
Volume Name:
Volume ID: df77712f9e08d8e9149527c56cc21fd

Last Written at: 2024-10-03 18:44:22 (WEST)
Last Checked at: 2024-09-12 09:37:30 (WEST)

Last Mounted at: 2024-10-03 18:44:22 (WEST)
Unmounted properly
Last mounted on: /mnt/johnnyDisk

Source OS: Linux
Dynamic Structure
Compat Features: Journal, Ext Attributes, Resize Inode, Dir Index
InCompat Features: Filetype, Needs Recovery, Extents, 64bit, Flexible Block Groups,
Read Only Compat Features: Sparse Super, Large File, Huge File, Extra Inode Size

Journal ID: 00
Journal Inode: 8

METADATA INFORMATION
--------------------------------------------
Inode Range: 1 - 1638401
Root Directory: 2
Free Inodes: 1412670
Inode Size: 256

CONTENT INFORMATION
--------------------------------------------
Block Groups Per Flex Group: 16
Block Range: 0 - 6552831
Block Size: 4096
Free Blocks: 3521575

BLOCK GROUP INFORMATION
--------------------------------------------
Number of Block Groups: 200
Inodes per group: 8192
Blocks per group: 32768

Group: 0:
  Block Group Flags: [INODE_ZEROED]
  Inode Range: 1 - 8192
  Block Range: 0 - 32767
  Layout:
    Super Block: 0 - 0
    Group Descriptor Table: 1 - 4
    Group Descriptor Growth Blocks: 5 - 1028
```

Figure 2: Output of fsstat –o 4096 johnnyDisk.img

After using the ***fls*** command to list file and directory names from a disk image on that specific partition. We didn´t suspect any directory names and files. So, we tried to check if any important files had been deleted using the same ***fls*** command and adding ***–Frd*** flag. The output doesn´t show any relevant deleted files or directories.

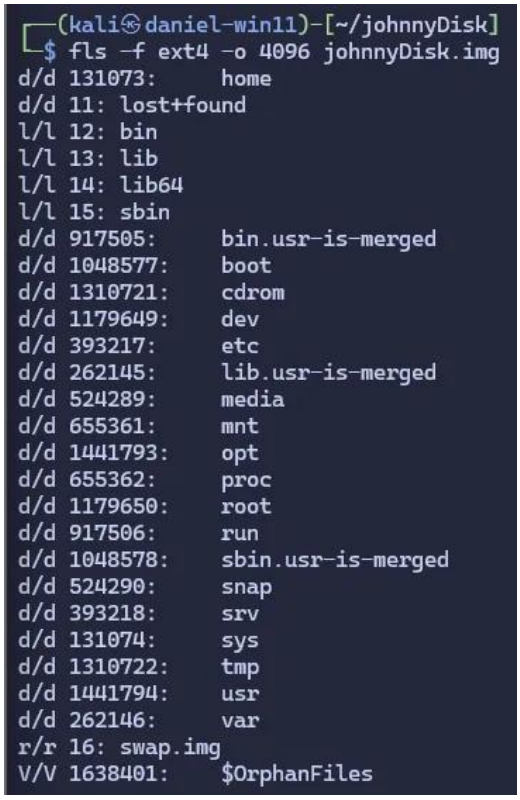

Figure 3: Output of fls –f ext4 –o 4096 johnnyDisk.img

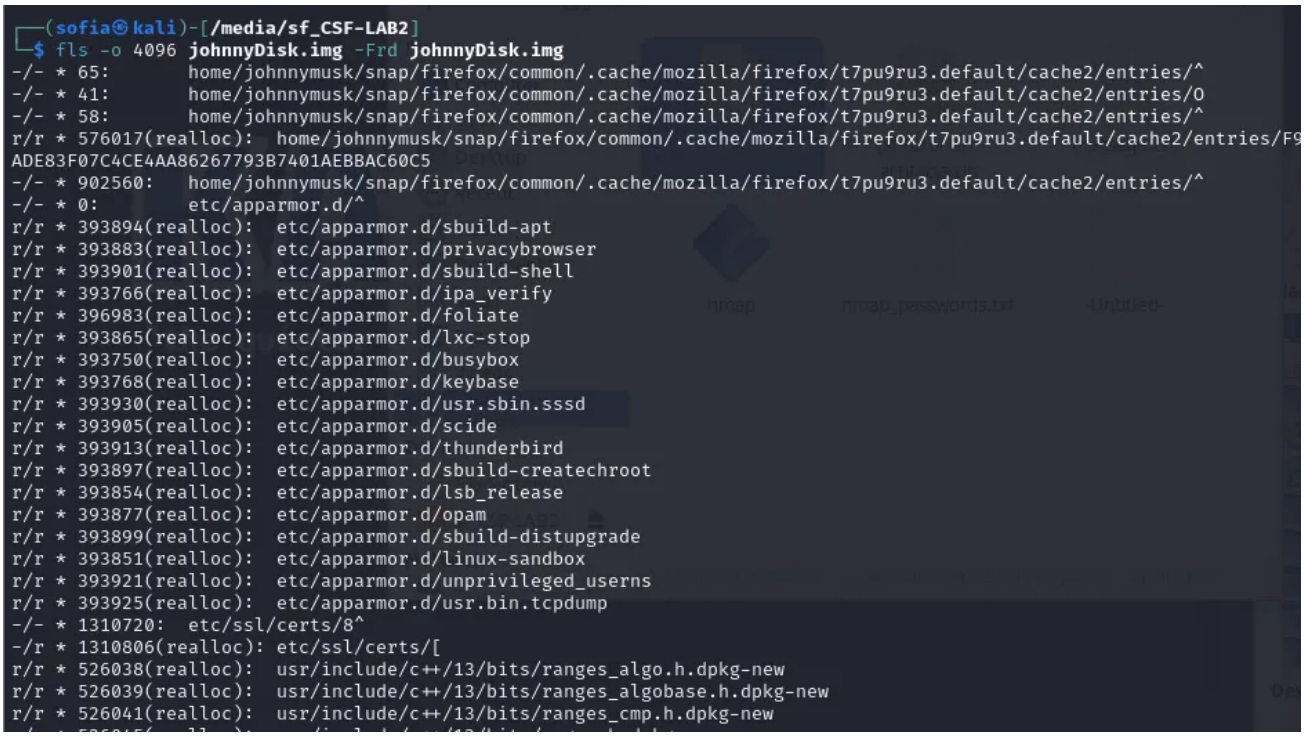

Figure 4: Output of fls –o 4096 johnnyDisk.img -Frd johnnyDisk.img

Besides using the **istat** and **icat** commands, we also used **TKF Imager.** We found the hidden artifacts and the files originally discovered in João Musk's sigma account in the below paths:

- root/home/johnnymusk/.cache/thumbnails/normal
- root/home/johnnymusk/ .cache/thumbnails/large
- root/home/johnnymusk/Documents
- root/home/johnnymusk/Music

We also found many scripts, that could be evidence of an anti-forensic activities, in the root/home/johnnymusk/s tt

- converter.py (the code converts the input file´s content to a binary code and then can read it)
- createChunks.py (the code is used to split the contents of a text file into 3 chunks and save each chunk as a separate file)
- doubleEncodingAndHidingInsideElf.py (the code encrypts the input files information, input_path = '/home/johnnymusk/Documents/hackedcredentials.txt')
- hide_pdf.py (script to hide secret.pdf after the EOF of input.wav)
- cookieSteal.html, exploit.htpl and exploit.py (with this files content it seems like the attacker want runs in the user´s browser and then capturing the user´s cookies)
- lsb.pcy (this is a compiled bytecode that shows us the python code is used to encoded diagonally payload in the specific image)

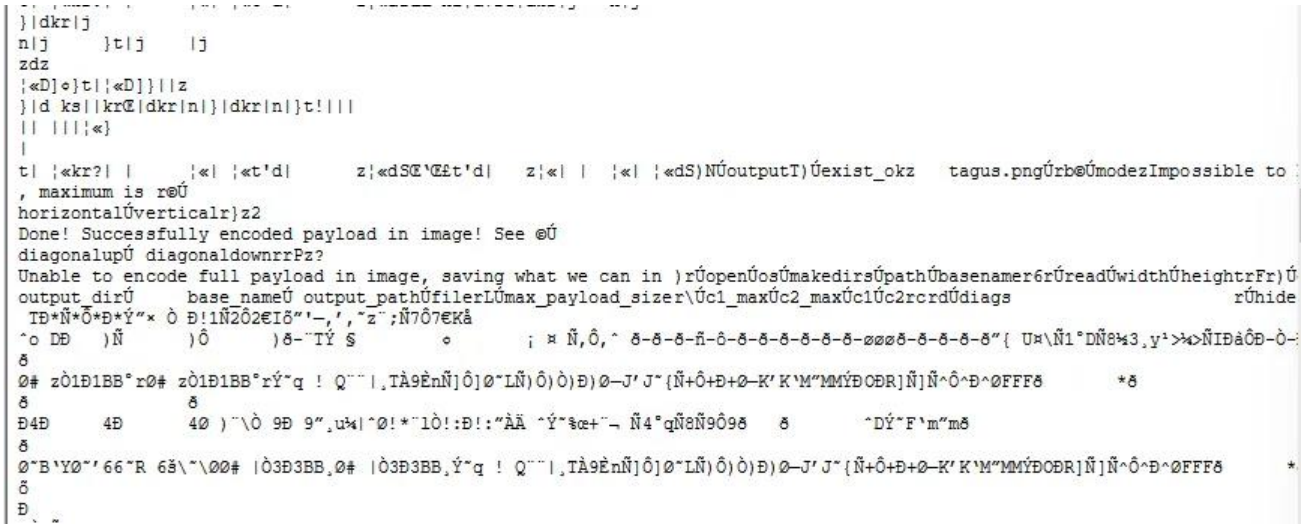

Figure 4: lsb.pcy file

We found many TikTok-like videos and noticed that one of them had been modified (Snapinsta.app_video_0A4741B6D39EC9A52A68DA4898659396_video_dashinit.mp4), so we analyzed it by checking its magic numbers. As its magic number coincided with the mp4 file, it means that this is only a simple video.

66 74 79 70 69 73 6F 6D	ftypisom	4	mp4	ISO Base Media file (MPEG-4)

Figure 5: magic numbers for identify mp4 files

```
┌──(sofia㉿kali)-[/media/sf_CSF-LAB2]
└─$ fls -o 4096 -r johnnyDisk.img | grep '\.mp4$'
++++ r/r 576592:        Snapinsta.app_video_0F4143EB1546176E1E53B4723A5A0095_video_dashinit.mp4
+++ r/r 533494: Snapinsta.app_video_3844763E4D8D071250A9FB1CCF4A538D_video_dashinit.mp4
+++ r/r 533493: Snapinsta.app_video_7F4C59AE5584B56909DBAE9C3E9425AB_video_dashinit.mp4
+++ r/r 533490: Snapinsta.app_video_0F4143EB1546176E1E53B4723A5A0095_video_dashinit.mp4
+++ r/r 533491: Snapinsta.app_video_0A4741B6D39EC9A52A68DA4898659396_video_dashinit.mp4
+++ r/r 533520: Snapinsta.app_video_CB4741AC8D9FF45575192E4B94799AA8_video_dashinit.mp4
+++ r/r 533501: Snapinsta.app_video_25478BFC98DE275039A64F2ABBA92AAD_video_dashinit.mp4
+++ r/r 533495: Snapinsta.app_video_AQNCIIWQ5i8wTfaZ3ysCxqZdffmEEi_QeMHwpYQ6hQowY68576lQbqKm43PSDD49gTFSfN9FlkNyONxC
M6hiFZF.mp4
+++ r/r 533448: Snapinsta.app_video_AQPr38rKOynmy11fKcIDLxN6B2h_QvvWaW7VGGT6cYFf1EfLoEGhSyzFWzY_0f_j4h4DFArZQ9yaYHE
HlXoBRl.mp4
+++ r/r 533492: Snapinsta.app_video_AQMC56_VjmP5rC8-6qNkgeAomxZskLqw5PXfK0ksgp447VkOUmg2JddLqkEnV8sYKNS6c86jgX_kAHZ
AJluSEz.mp4

┌──(sofia㉿kali)-[/media/sf_CSF-LAB2]
└─$ istat -o 4096 johnnyDisk.img 533491 | less

zsh: done       istat -o 4096 johnnyDisk.img 533491 |
zsh: suspended  less

┌──(sofia㉿kali)-[/media/sf_CSF-LAB2]
└─$ icat -o 4096 johnnyDisk.img 533491  | xxd | head -n 10
00000000: 0000 0020 6674 7970 6973 6f6d 0000 0200  ... ftypisom....
00000010: 6973 6f6d 6973 6f32 6176 6331 6d70 3431  isomiso2avc1mp41
00000020: 0000 648b 6d6f 6f76 0000 006c 6d76 6864  ..d.moov...lmvhd
00000030: 0000 0000 0000 0000 0000 0000 0000 03e8  ................
00000040: 0000 752e 0001 0000 0100 0000 0000 0000  ..u.............
00000050: 0000 0000 0001 0000 0000 0000 0000 0000  ................
00000060: 0000 0000 0001 0000 0000 0000 0000 0000  ................
00000070: 0000 0000 4000 0000 0000 0000 0000 0000  ....@...........
00000080: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000090: 0000 0003 0000 4e01 7472 616b 0000 005c  ......N.trak...\

┌──(sofia㉿kali)-[/media/sf_CSF-LAB2]
└─$ icat -o 4096 johnnyDisk.img 533491  | xxd | tail -n 10
0033c0e0: d397 3f61 d805 1eec 3e8c b091 0c4a b1fb  ..?a....>....J..
0033c0f0: b653 22e1 78c0 6b82 f130 db18 558f fd61  .S".x.k..0..U..a
0033c100: 4bf9 c89e 713c e370 7e7c d08d 0eb1 efd9  K...q<.p~|......
0033c110: 25c1 cf0b dfbe 5cf9 3653 b797 f4c5 fcd5  %.....\.6S......
0033c120: 062d fecf 6787 2e3c 8ca0 1381 f735 250a  .-..g..<.....5%.
0033c130: a442 1e00 2a94 4432 26dd 010a 9450 0000  .B..*.D2&....P..
0033c140: 0000 0000 0000 0e21 1b4f ffff ff99 01e6  .......!.O......
0033c150: 03ee 6a4b 053b 0384 3fc5 52a5 1bd3 01f7  ..jK.;..?.R.....
0033c160: 3525 829d 81c2 1fe2 a952 8ded ba02 1500  5%.......R......
0033c170: 0000 0000 0000 0000 001c                 ..........

┌──(sofia㉿kali)-[/media/sf_CSF-LAB2]
└─$ icat -o 4096 johnnyDisk.img 533491  | strings
ftypisom
```

Figure 6: Terminal output showing the analysis of the inode where the suspicious video was located

The first secret we had found is the User_Manual.pdf file in the root/home/johnnymusk/Documents. This is a master´s thesis document about *Software* de telemetria do Ariane 6 by David Alexandre Ferreira da Silva.

In the root/home/johnnymusk/ .cache/thumbnails/fail/large path, we found a rocket image (200d463649820c65548b121e4f42ad68.png), we think that it is Ariana 6.

Figure 7: Ariana 6 (200d463649820c65548b121e4f42ad68.png)

In the same folder, we saw an image about IST´ study rooms map at TagusPark campus, in the ceaf1f22d9a118942a4f7ad833c36b97.png file.

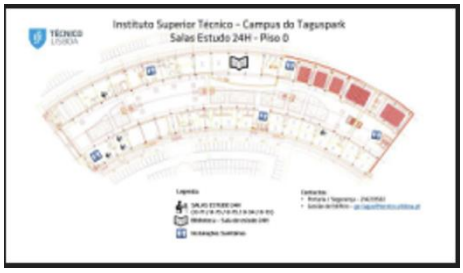

Figure 8: IST´ study rooms map at TagusPark campus

We also uncovered an important IRSSI chat history in the #thebasement.09-26.log file, located in the johnnyDisk/root/home/johnnymusk/snap/irssi/common/irclogs/2024/freenode path, between two users: johnnymusk, who appears to be the student Musk, and RootKitty, whose identity we do not yet know, maybe a colleague of João´s.

In the first part of the conversation, RootKitty informs johnnymusk that he/she successfully exploited Fenix and obtained pairs of usernames and passwords using techniques that developed in STT. In the other chat log part, johnnymusk shares a bizarre experience where he received an anonymous email instructing them to retrieve a USB drive hidden near IST. When he did, the USB contained unencrypted files with disturbing information (an Api documentation for a mind control, a blueprint of the Ariane 6 rocket, a bank statement showing a transfer to a man named Virgolino Gonçalves from a company called ERCE.LTA, which was used to buy a mind control component (MKU-2784) from MOBICARE, and Logs suggesting the mind control technology was used to influence people to visit specific restaurants). In the final conversation, the user johnnymusk decided to encrypt the USB contents in files and share them using the Sigma cluster. They also wanted to organize a protest and push for the satellite to be deactivated.

```
--- Log opened Thu Sep 26 16:30:45 2024
16:30 -!- johnnymusk [~johnnymus@freenode-995.pbo.e924c1.IP] has joined #thebasement
16:30 -!- Irssi: #thebasement: Total of 3 nicks [1 ops, 0 halfops, 0 voices, 2 normal]
16:30 -!- Irssi: Join to #thebasement was synced in 8 secs
16:30 < RootKitty> Hey João, you're not gonna believe what I managed to pull off. Using a combination of the techniques we use in STT, I was able to exploit Fenix and steal pairs of usernames and passwords.
16:31 < johnnymusk> Wait, seriously? How did you even do that? I've been poking around Fenix for ages and got nowhere!
16:31 < RootKitty> Let's just say experience comes in handy, and I got creative with a few of the methods we've been testing.
16:31 < johnnymusk> You're on another level! I can't believe you pulled it off when I couldn't even scratch the surface. What's your secret?
16:31 < RootKitty> Nice try, but I'm not giving away the whole playbook. I'll send you the credentials, though, so you can see for yourself.
16:32 < johnnymusk> You're a legend. I have to see this! I never thought Fenix could be cracked like that.
16:32 < RootKitty> Yeah, well, don't get too hyped. This stays between us, no need to broadcast it.
16:32 < johnnymusk> Don't worry, I know the rules. I'm just eager to see if those creds really work. You're seriously next-level.
16:33 < RootKitty> Appreciate it, João. Just remember to keep it quiet. We don't need extra eyes on this.
16:33 < johnnymusk> Of course, you can trust me. I wouldn't jeopardize anything. I'm just lucky I get to see how you do this stuff.
16:33 < RootKitty> Good. I'll send you the creds in a bit, and you can check it out for yourself.
16:33 < johnnymusk> I can't wait! Just, uh… make sure this doesn't backfire on us, right? IST won't take this lightly if the creds get out.
16:34 < RootKitty> I know, João. I've got this under control. You're not at the point where you need to worry about that kind of thing.
16:34 < johnnymusk> Yeah, I trust you. You always know what you're doing. Just send those creds when you're ready, I'm dying to see what you cracked.
16:35 < RootKitty> As promised, here they are https://we.tl/t-Yg5WlqlnxF.
16:36 < RootKitty> You can use the script that we made together to hide the credentials! I had a feeling you would have a use for it soon! Catch you later, João!
16:36 < johnnymusk> Thanks Kitty! You're always one step ahead.
--- Log closed Thu Sep 26 16:36:18 2024
--- Log opened Thu Sep 26 17:01:03 2024
17:01 -!- johnnymusk [~johnnymus@freenode-995.pbo.e924c1.IP] has joined #thebasement
17:01 -!- Irssi: #thebasement: Total of 3 nicks [1 ops, 0 halfops, 0 voices, 2 normal]
17:01 -!- Irssi: Join to #thebasement was synced in 8 secs
17:01 < johnnymusk> RootKitty, you're not gonna believe what just happened. I got this anonymous email telling me to retrieve a USB pen from a hidden spot near IST.
17:01 < johnnymusk> I thought it was some kind of joke, but I went anyway.
17:01 < RootKitty> Seriously?
17:02 < RootKitty> And you actually found it?
17:02 < johnnymusk> Yeah, I found it. When I plugged it into my computer, I discovered a folder with some seriously disturbing information.
17:02 < johnnymusk> No encryption, just right there for anyone to see.
17:02 < RootKitty> Okay, now you've got me intrigued. What was in the folder?
17:03 < johnnymusk> It had a bunch of documents. Among them was the API documentation for a mind control component called MKUltra, showing various parameters used for mind control.
17:03 < johnnymusk> There was also a blueprint of the Ariane 6 rocket, which specifically highlighted the position of the ISTSAT-1 satellite in its payload.
17:03 < johnnymusk> But that's not all. I found a bank statement belonging to someone named Virgolino Gonçalves, showing that he received a massive transfer from a company called ERCE.LTA. He then used that money to make a payment to MOBICARE to purchase a component
called MKU-2784.
17:04 < RootKitty> Wait, MKU-2784?
17:04 < RootKitty> That sounds like it could be part of the mind control tech.
17:04 < RootKitty> What else was in the folder?
17:04 < johnnymusk> Exactly what I was thinking. There were also logs showing that this mind control component has been actively used.
17:04 < johnnymusk> The logs indicate it's been influencing people to visit four specific restaurants in the Oeiras area. And, if that wasn't strange enough, there was a DECO report included, showing that those four restaurants particularly Pombalino and Caçoila are now
super trendy.
17:05 < johnnymusk> Their revenues have almost doubled compared to last year, right after ISTSAT-1 was launched.
17:05 < RootKitty> Hold on, you're telling me the mind control tech is being used to send people to restaurants and boost their business? That's crazy, but it actually sounds like it's all connected. What do you think?
17:05 < johnnymusk> That's exactly what I'm thinking.
17:05 < johnnymusk> I'm convinced it's all linked. The satellite, the MKUltra API, the payment trail from ERCE.LTA to Virgolino and then to MOBICARE for that MKU-2784 component, and the sudden surge in popularity and revenue for those restaurants it all lines up.
17:06 < johnnymusk> The mind control tech is being used to manipulate people and drive them to these specific places.
17:06 < RootKitty> Wow. This is huge.
17:06 < RootKitty> If you're right, we're looking at a direct application of mind control tech being used for profit. What are you going to do with this?
17:07 < johnnymusk> I want to organize a protest and push for the satellite to be deactivated. This kind of technology shouldn't be operational, especially with such invasive and unethical uses.
17:07 < RootKitty> I agree, but we need to present this in a way that grabs attention. How about creating a poster? Something visual that really highlights the key details and gets people questioning what's going on?
17:07 < johnnymusk> That's a solid idea.
17:07 < johnnymusk> A simple poster could definitely help get the word out. I'll start working on it now.
17:08 < RootKitty> Great
17:08 < RootKitty> Make sure the poster is compelling but not too flashy to avoid raising alarms. We don't want the people behind this to catch on.
17:08 < johnnymusk> Will do.
17:09 < johnnymusk> I'll keep it straightforward and clear, but subtle enough to avoid drawing unwanted attention.
17:09 < johnnymusk> Also, I'll hide the sensitive details in some files using tactics I've learned from CTFs. Then, I'll save them in my private account on the Sigma cluster for safekeeping.
17:09 < RootKitty> Sounds like a good plan. Keeping the sensitive info secure is crucial.
17:09 < johnnymusk> Thanks for the support, RootKitty. The files will be well-protected.
17:09 < RootKitty> Be cautious with this
17:09 < RootKitty> If the information is accurate, we might be dealing with some powerful people.
17:09 < johnnymusk> I'll be careful. I'll make sure everything is secure.
--- Log closed Thu Sep 26 17:10:01 2024
```

Figure 9: IRSSI chat history between RootKitty and johnnymusk

Due to the nature of chat history, we began searching for evidence of when the USB was plugged into the computer. We found relevant information in this file /var/log/syslog, where if we search for the usb, we find the times where it was inserted a usb device in the computer, manufacturer, serial number, etc.

```
2024-09-26T16:51:44.045184+01:00 mainframe kernel: usb 1-3: new high-speed USB device number 3 using xhci_hcd
2024-09-26T16:51:44.460092+01:00 mainframe kernel: usb 1-3: New USB device found, idVendor=058f, idProduct=6387, bcdDevice= 1.06
2024-09-26T16:51:44.460197+01:00 mainframe kernel: usb 1-3: New USB device strings: Mfr=1, Product=2, SerialNumber=3
2024-09-26T16:51:44.460200+01:00 mainframe kernel: usb 1-3: Product: Mass Storage
2024-09-26T16:51:44.460201+01:00 mainframe kernel: usb 1-3: Manufacturer: Generic
2024-09-26T16:51:44.460204+01:00 mainframe kernel: usb 1-3: SerialNumber: 9AD32EC0
2024-09-26T16:51:44.486986+01:00 mainframe mtp-probe: checking bus 1, device 3: "/sys/devices/pci0000:00/0000:00:02.1/0000:02:00.0/usb1/1-3"
2024-09-26T16:51:44.487294+01:00 mainframe mtp-probe: bus: 1, device: 3 was not an MTP device
2024-09-26T16:51:44.495782+01:00 mainframe kernel: usb-storage 1-3:1.0: USB Mass Storage device detected
2024-09-26T16:51:44.495791+01:00 mainframe kernel: scsi host6: usb-storage 1-3:1.0
2024-09-26T16:51:44.496308+01:00 mainframe kernel: usbcore: registered new interface driver usb-storage
2024-09-26T16:51:44.502196+01:00 mainframe kernel: usbcore: registered new interface driver uas
2024-09-26T16:51:44.507364+01:00 mainframe mtp-probe: checking bus 1, device 3: "/sys/devices/pci0000:00/0000:00:02.1/0000:02:00.0/usb1/1-3"
2024-09-26T16:51:44.507484+01:00 mainframe mtp-probe: bus: 1, device: 3 was not an MTP device
2024-09-26T16:51:45.544519+01:00 mainframe kernel: scsi 6:0:0:0: Direct-Access     Generic  Flash Disk       8.07 PQ: 0 ANSI: 4
2024-09-26T16:51:45.545176+01:00 mainframe kernel: sd 6:0:0:0: Attached scsi generic sg1 type 0
```

Figure 10: syslog file

As we could see, it has 15.7 GB of storage, not write protected, mounted at /media/johnnymusk/72B5-E8E1, and opening the document BankSt.

```
2024-09-26T16:51:50.261575+01:00 mainframe systemd[1]: Started systemd-hostnamed.service - Hostname Service.
2024-09-26T16:52:19.279395+01:00 mainframe dbus-daemon[1584]: [session uid=1000 pid=1584] Activating service name='org.
gnome.evince.Daemon' requested by ':1.437' (uid=1000 pid=30598 comm="/usr/bin/evince /media/johnnymusk/72B5-E8E1/BankSt"
label="/usr/bin/evince (enforce)")
2024-09-26T16:52:19.289547+01:00 mainframe dbus-daemon[1584]: [session uid=1000 pid=1584] Successfully activated service
'org.gnome.evince.Daemon'
```

Figure 11: evidence at /media/johnnymusk/72B5-E8E1

Also in the syslog file, we found the creation and running of cron jobs, scheduled to run a script.

```
2024-09-23T19:00:01.356259+01:00 mainframe systemd[1]: Starting sysstat-collect.service - system activity accounting tool...
2024-09-23T19:00:01.358758+01:00 mainframe CRON[6985]: (johnnymusk) CMD (sh ~/backups/backup.sh)
2024-09-23T19:00:01.361372+01:00 mainframe systemd[1]: sysstat-collect.service: Deactivated successfully.
```

Figure 12: syslog file

The cron job is shown in /var/spool/cron/crontabs

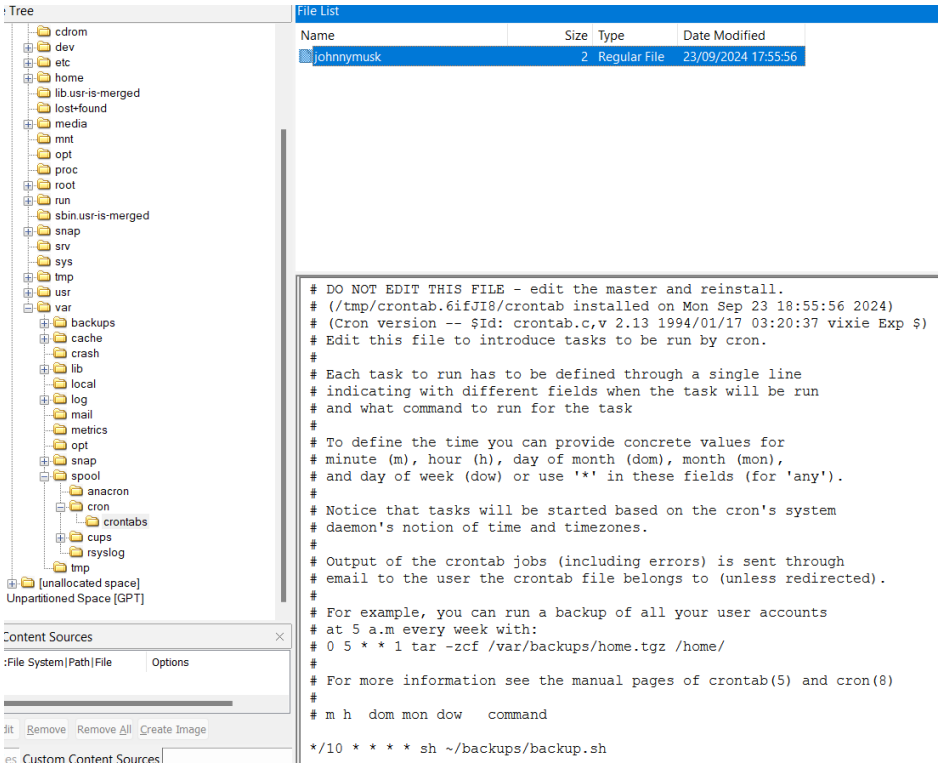

Figure 13: cron job

The backups folder contains 3 files:

- backup.sh – it zips the folder ~/Desktop/TVShows with a password generated with pass_gen.sh and a timestamp of the current date

```
#!/bin/bash

timestamp() {
    date +%s
}


TS=$(timestamp)
USER=johnnymusk
HOST=10.0.2.147
DIR=~
ZIPFILE=backup_$TS.zip
BACKUP_PASS=$(~/backups/pass_gen.sh $TS)

zip -r --password $BACKUP_PASS $ZIPFILE ~/Desktop/TVShows
rsync -avz -e "ssh -i ~/.ssh/id_rsa" ./$ZIPFILE $USER@$HOST:$DIR
rm $ZIPFILE
```

Figure 14: backup.sh

- pass_gen.sh – Runs the obfuscator
- obfuscator – obfuscated python file

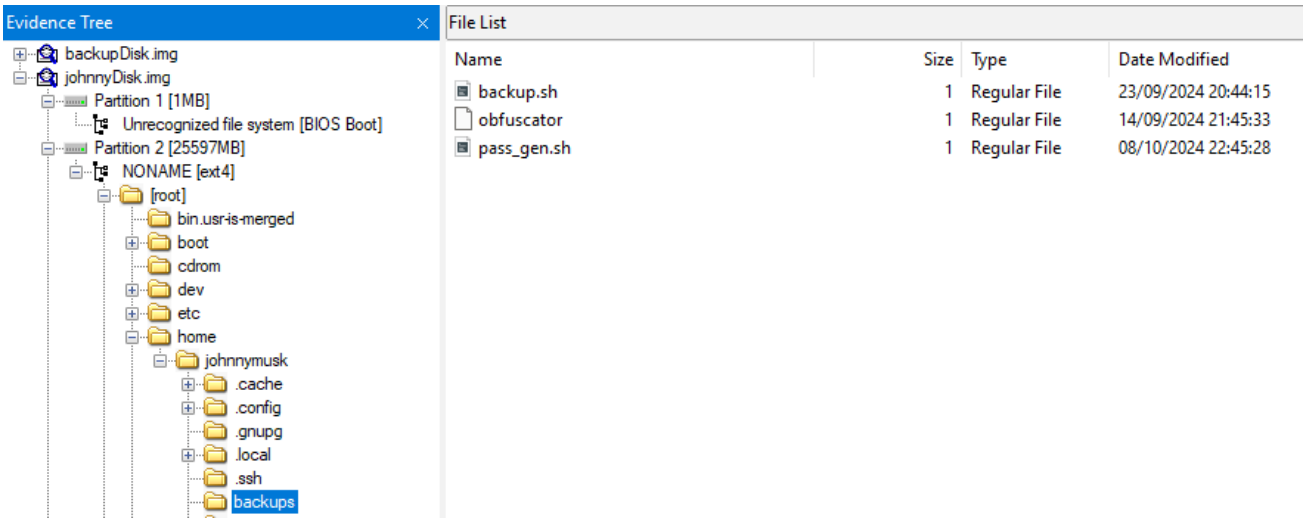

Figure 15: contents in the backups folder

To unzip the backup zips found in the backupDisk.img, we needed to find their password.

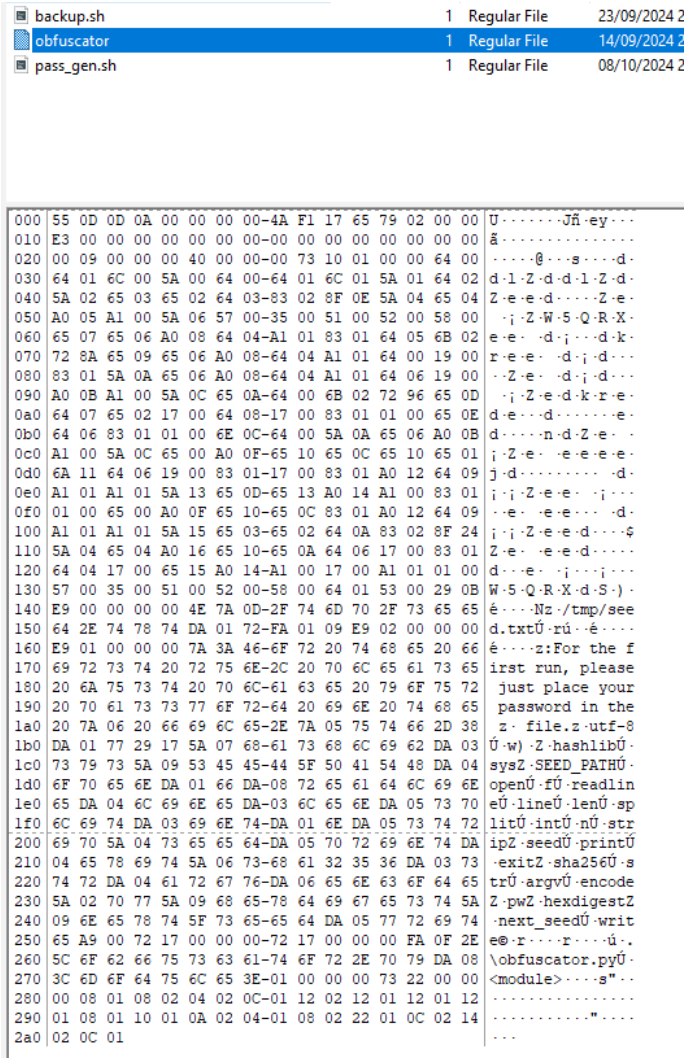

Figure 16: obfuscator file

As the file above was obfuscated, we had to reconstruct to get a runnable python script. In the file content, we know that it saves the seed in /tmp/seed.txt and in the first run, the seed was simply "1 <password>", so we needed to find this password.

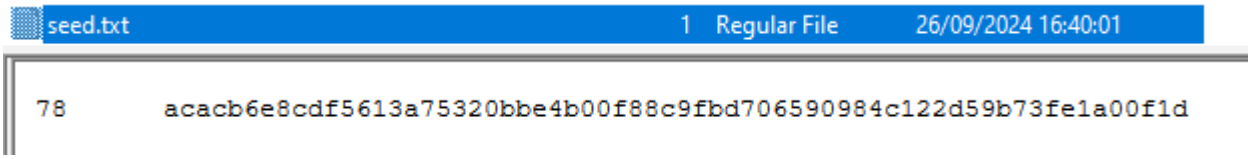

Figure 17: seed.txt

If we found the initial password as the seed, we could generate all the seeds in between the first and the last one (78 generations). If we found a password, but the generation of the 78th seed is not equal to the one in /tmp/seed.txt, the password was incorrect.

```python
# Function to generate the next seed from the password
def generate_seed(seed):
    # Convert the password to bytes and hash it using SHA-256
    sha256_hash = hashlib.sha256(seed.encode("utf-8")).hexdigest()
    print(f"SHA-256 hash of the password: {sha256_hash}")
    # next seed

    return sha256_hash
    # return next_seed
```

Figure 18: code to generate the next seed

Searching the disk, we found a keylogger file, where it shows the user typed "keepassxc", which is a password manager, and then typed "ilovemydadthegoat" where that could be the password of keepass.

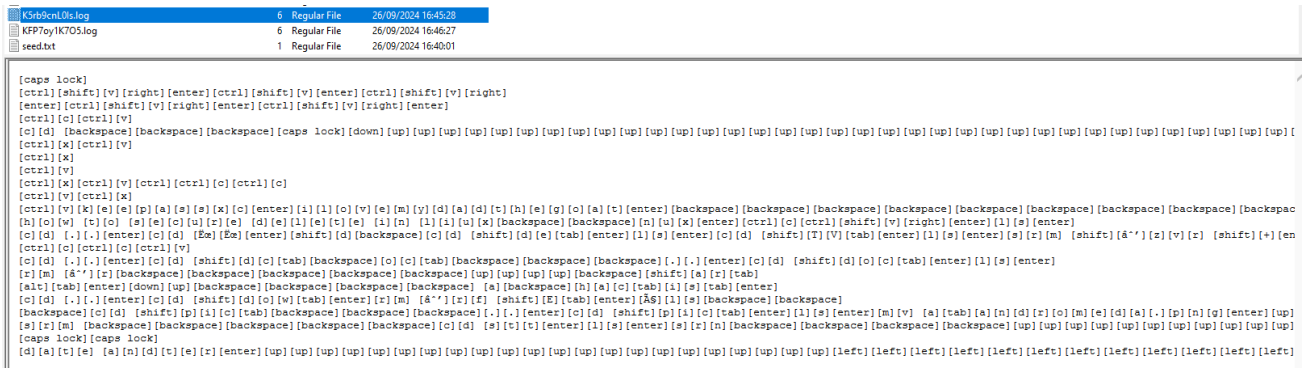

Figure 19: keylogger file

In /home/johnnymusk there is a file called Passwords.kdbx, where keepass stores all the passwords.

Opening this file online in the keepass website with the password "ilovemydadthegoat", we found the password of the zips, which is "**TheBiteOf87**"

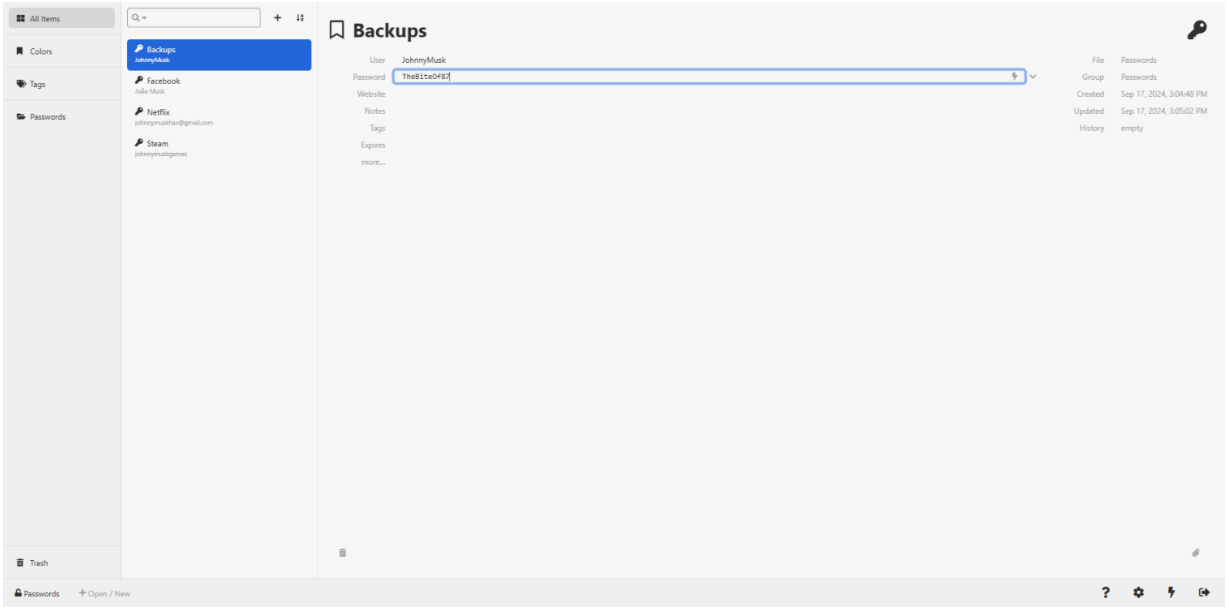

Figure 19: Screenshot showing the password being entered.

To generate the password, the obfuscator appends the timestamp to the seed, so for each zip, we need to test every seed + the zip timestamp found in their name.

```python
def create_password(seed, ts):
    combined = seed + ts
    password = hashlib.sha256(combined.encode("utf-8")).hexdigest()
    return password
```

Figure 20: code to create passord

```python
# Main function that runs the script
def main():

    seed = "TheBiteOf87"

    for i in range(78):
        # Generate the next seed based on the password
        seed = generate_seed(seed)
        print(f"Next seed generated: {i, seed}")

        # unzip the backup files with the seed
        zip_ts = [
            "1727365201",
            "1727365801",
            "1727366402",
            "1727367001",
            "1727367601",
            "1727368201",
            "1727368801",
        ]

        for ts in zip_ts:
            extract_folder = f"backup_{ts}"

            # Create the extraction folder if it doesn't exist
            if not os.path.exists(extract_folder):
                os.makedirs(extract_folder)

            try:
                password = create_password(seed, ts)
                with zipfile.ZipFile(f"backup_{ts}.zip", "r") as zip_ref:
                    zip_ref.extractall(path=extract_folder, pwd=bytes(password, "utf-8"))
                    print(f"{f"backup_{ts}.zip"} successfully unzipped with seed gen {i}.")
            except Exception:
                continue
```

Opening all the now unzipped folders, we found all the files from lab1. Many of them also sent the nmap file to ist90834 to 10.0.2.166 which is inside the IP range of the sigma cluster's services.

In the /home/johnnymusk directory, we also have the .bash_history file, where it has all the commands used in the bash terminal. Some important ones are opening the irssi chat, send of files to a remote user via scp (scp -r nmap ist90834@10.0.2.166), opening the usb device, running the anti-forensic python files, etc.

```
irssi
mv ~/Downloads/hackedcredentials.txt ~/Documents
xdg-open ~/Documents/hackedcredentials.txt
rm ~/Desktop/TVShows/
rm -r ~/Desktop/TVShows/
ls
cd Desktop/
ls
cd TVShows/
ls
cd ..
cd stt
ls
nano doubleEncodingAndHidingInsideElf.py
source .venv/bin/activate
python3 doubleEncodingAndHidingInsideElf.py
mv nmap ~/Desktop/TVShows/
cd ~/Desktop/TVShows/
ls
strings nmap
scp -r nmap ist90834@10.0.2.166
scp -r nmap ist90834@10.0.2.166:~
cd /media/johnnymusk/72B5-E8E1
```

Figure 22: .bash._history file

In /home/johnnymusk/snap/thunderbird/common/.thunderbird/iw2y9jr6.default/Mail/pop.gmail.com we found the file Inbox, with an email sent by somebodysupercool@protonmail.com on 26 September 2024.

The Inbox file has the email contents encoded in base64, so we created a python script to decode those parts to read the contents of this email.

The sender of this email is talking about the mind control program, MKUltra and where to find a USB with all the information about this project, and to maintain secrecy.

```
Date: Thu, 26 Sep 2024 15:36:08 +0000
To: "johnnymuskhax@gmail.com" <johnnymuskhax@gmail.com>
From: somebodysupercool <somebodysupercool@protonmail.com>
Subject: SUPER IMPORTANT
Message-ID: <0oq_Awt6_HxcDu-9h3SQ_AhnUopzcCHpTfCmW4mIHDB7UZG4QEXvx-c1fRa89ly2rzSv
Feedback-ID: 120342430:user:proton
X-Pm-Message-ID: c144957813a6e46bca09967d82226f9f91bc9d4a
MIME-Version: 1.0
Content-Type: multipart/alternative;
 boundary="b1_dmpQ3KyKBug53VEs6WHrak6MOBWvKmSRzGWh01x7Q"

This is a multi-part message in MIME format.

--b1_dmpQ3KyKBug53VEs6WHrak6MOBWvKmSRzGWh01x7Q
Content-Type: text/plain; charset=utf-8
Content-Transfer-Encoding: base64

TGlzdGVuIHVwLAoKSeKAmW0gZHJvcHBpbmcgdGhpcyBvbiB5b3UgYmVjYXVzZSB5b3UgbmVlZCB0
byBrbm93LiBUaGVyZeKAmXMgc29tZXRoaW5nIGJpZyBnb2luZyBvbuKAlHNvbWV0aGluZyBubyBv
bmXigJlzIHRhbGtpbmcgYWJvdXQuIEV2ZXIgaGVhcmQgb2YgTUtVbHRyYT8gSXTigJlzIGEgbWlu
ZCBjb250cm9sIHByb2dyYW0uIFNvdW5kcyBjcmF6eSwgcmlnaHQ/IFdlbGwsIGl04oCZcyByZWFs
LCBhbmQgaXTigJlzIHRpZWQgd8gdGhlIEFyaWFuZS02IHByb2plY3QgYW5kIGl0IGlzIHRhcmdl
dGluZyB0aGUgT2VpcmFzJyBwb3B1bGF0aW9uLgoKSeKAmXZlIGdvdCB0aGUgcHJvb2YuIERvY3Vt
ZW50cywgZmlsZXMsIHRoZSB3a69sZSBkZWFsLiBUaGlzIGlzbuKAmXQgc3R1ZmYgeW914oCZbGwg
ZmluZCBhbnl3aGVyZSBlbHNlLiBJdOKAmXMgYWxsIGJlZW4ga2VwdCBxdWlldCwgYnV0IG5vdCBh
bnltb3JlLiBJ4oCZdmUgc3Rhc2hlZCBpdCBmb3IgeW91IHRvIGNoZWNrIG91dCB5b3Vyc2VsZi4K
CkhlcmXigJlzIHdoZXJlIHlvdSBjYW4gZmluZCB0aGUgVVNCIHdpdGggZXZlcnl0aGluZzoKTG9j
a2VyTG9ja3kgQ1RUIElTVCBMaXNib2EKQXYuIFJvdmlzY28gUGFpcyAxClBvc3RhbCBDb2RlOiAx
MDAwLTI2NwpMb2NrZXIgMDMKQ29kZTogNjY2CgpHZXQgdGhlcmUsIHBpY2sgdXAgdGhlIFVTQi4g
SXQncyBjbG9zZSB0byB5b3VyIGhvbWUgYW5kIHdhaXRpbmcgZm9yIHlvdS4gRGlnIHRocm91Z2gg
dGhlIGZpbGVzIEFTQVAuIEJ1dCBtb3ZlIGNhcmVmdWxseS4gVGhpcyBpc27igJl0IGluZm8gdGhl
eSB3YW50IGdldHRpbmcgb3V0Lg==
```

Figure 23: email contents

```python
def decode_base64_from_file(file_path):
    with open(file_path, "r") as file:
        content = file.read()

    # Regular expression to find potential Base64 strings
    base64_pattern = re.compile(r"([A-Za-z0-9+/=]{20,})")
    base64_strings = base64_pattern.findall(content)

    decoded_strings = []
    for b64_string in base64_strings:
        try:
            decoded_data = base64.b64decode(b64_string).decode("utf-8")
            decoded_strings.append(decoded_data)
        except Exception as e:
            print(f"Failed to decode string: {b64_string}, Error: {e}")

    return decoded_strings
```

Figure 24: python script to decode email contents

```
Listen up,

I'm dropping this on you because you need t
o know. There's something big going on—something no o
ne's talking about. Ever heard of MKUltra? It's a min
d control program. Sounds crazy, right? Well, it's real
, and it's tied to the Ariane-6 project and it is targe
ting the Oeiras' population.

I've got the proof. Docum
ents, files, the whole deal. This isn't stuff you'll
find anywhere else. It's all been kept quiet, but not a
nymore. I've stashed it for you to check out yourself.


Here's where you can find the USB with everything:
Loc
kerLocky CTT IST Lisboa
Av. Rovisco Pais 1
Postal Code: 1
000-267
Locker 03
Code: 666

Get there, pick up the USB.
It's close to your home and waiting for you. Dig through
the files ASAP. But move carefully. This isn't info the
y want getting out.
```

Figure 25: email contents after decode

The browser history was found in /home/johnnydisk/snap/firefox/common/.mozilla/firefox/t7pu9ru3.default, if we run "sqlite3 places.sqlite" we can query the data and save it to a csv.

```
┌──(kali㉿daniel-win11)-[/mnt/johnnyDisk/home/johnnymusk/snap/firefox/common/.mozilla/firefox/t7pu9ru3.default]
└─$ sqlite3 places.sqlite
SQLite version 3.46.0 2024-05-23 13:25:27
Enter ".help" for usage hints.
sqlite> SELECT url, title FROM moz_places;
```

Figure 26: output of sqlite3 places.sqlite

history.csv

history.csv > data

```
1  url,title,visit_count,last_visit_date
2  https://support.mozilla.org/products/firefox,,0,
3  https://support.mozilla.org/kb/customize-firefox-controls-buttons-and-toolbars?utm_source=firefox-browser&utm_medium=defa
4  https://www.mozilla.org/contribute/,,0,
5  https://www.mozilla.org/about/,,0,
6  https://www.mozilla.org/firefox/?utm_medium=firefox-desktop&utm_source=bookmarks-toolbar&utm_campaign=new-users&utm_conte
7  https://www.mozilla.org/privacy/firefox/,,1,1727114989270193
8  https://www.mozilla.org/en-US/privacy/firefox/,"Firefox Privacy Notice — Mozilla",1,1727114989403804
9  https://www.google.com/search?client=ubuntu-sn&channel=fs&q=github,"github - Pesquisa Google",2,1727367636763304
10 https://github.com/,GitHub,3,1727367640109876
11 https://github.com/login,"Sign in to GitHub · GitHub",1,1727114999735158
12 https://www.youtube.com/,YouTube,2,1727115995586678
13 https://accounts.google.com/ServiceLogin?service=youtube&uilel=3&passive=true&continue=https%3A%2F%2Fwww.youtube.com%2Fsi
14 https://accounts.google.com/InteractiveLogin?continue=https://www.youtube.com/signin?action_handle_signin%3Dtrue%26app%3D
15 https://accounts.google.com/v3/signin/identifier?continue=https%3A%2F%2Fwww.youtube.com%2Fsignin%3Faction_handle_signin%3
16 https://accounts.google.com/v3/signin/challenge/pwd?TL=APps6eZSUWLpJTAM3cqDNruglHTcKK_y8ihaKZe3xvXS0a5I5uvGtTmOKwJJWfcu&c
17 https://accounts.google.com/CheckCookie?continue=https://www.youtube.com/signin?action_handle_signin%3Dtrue%26app%3Ddeskt
18 https://accounts.youtube.com/accounts/SetSID?ssdc=1&sidt=ALWU2csI2pYROCXyt%2BvtMPUxzWJq3Z972MwLomjkLBXkqDxJdQ%2BxvM5BqNDz
19 https://accounts.google.pt/accounts/SetSID?ssdc=1&sidt=ALWU2cvrRL8LCInK0FQZxWzq9w6nN3tom1KRCxWroJ5ZbiHlzWeqHpaYhPc4tNwXIe
20 https://www.youtube.com/signin?action_handle_signin=true&app=desktop&hl=en&next=https://www.youtube.com/,,1,1727115995450
21 https://www.youtube.com/watch?v=TKuQ8Zkatd8,"Game Theory: FNAF, Golden Freddy NEVER Existed! - YouTube",1,1727116009166
```

Figure 27: history.csv

We found multiple searches of Oeiras restaurants, searches about mind control and hacking tools.

- **backupDisk.img**

The first command that we executed is ***mmls backupDisk.img.***

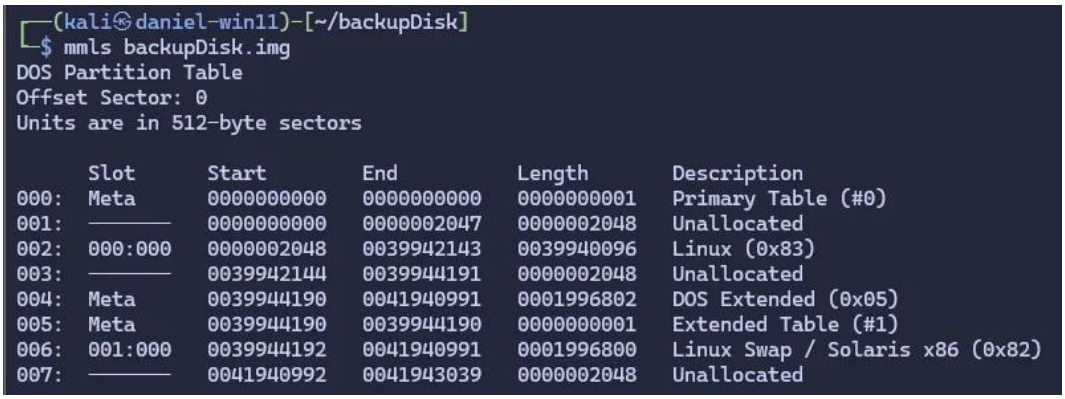

Figure 28: output of mmls backupDisk.img

As we can see here, all of these slots have descriptions. So, we decided to analyze the root partition using the ***fsstat -o 2048 backupDisk.img | less*** command. The partition that starts at sector offset 2048.

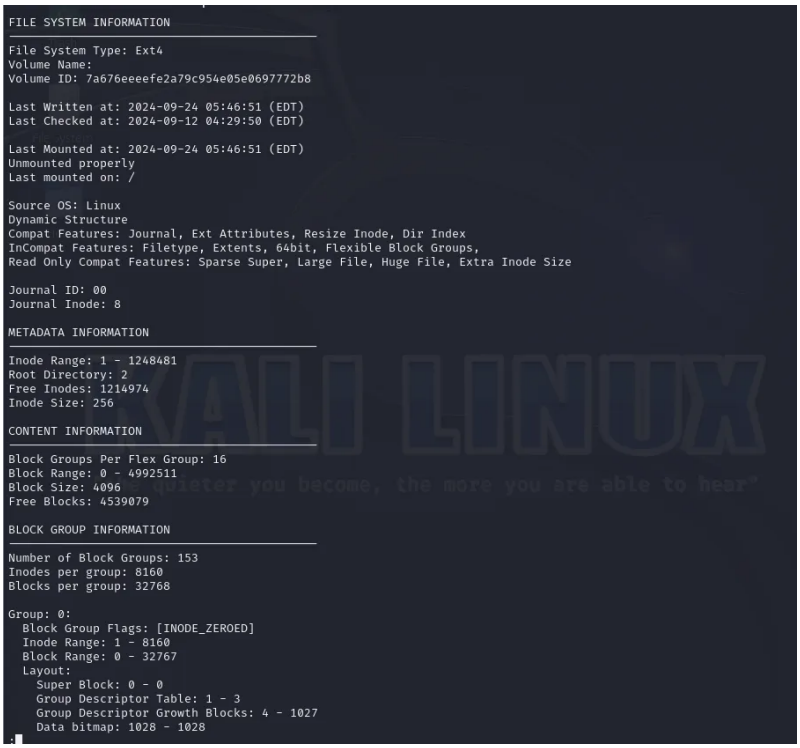

Figure 29: output of fsstat -o 2048 backupDisk.img | less

Using **FTK Imager**, we also found the hidden artifacts and the files originally discovered in João Musk's sigma account in the below paths:

- BackupDisk.img/Partition 1/ext4/root/home/johnnymusk/backup_1727368801.zip
- BackupDisk.img/Partition 1/ext4/root/home/johnnymusk/backup_1727367601.zip
- BackupDisk.img/Partition 1/ext4/root/home/johnnymusk/backup_1727367001.zip
- BackupDisk.img/Partition 1/ext4/root/home/johnnymusk/backup_1727366402.zip
- BackupDisk.img/Partition 1/ext4/root/home/johnnymusk/backup_1727365801.zip
- BackupDisk.img/Partition 1/ext4/root/home/johnnymusk/backup_1727365201.zip

2 Analysis of relevant findings

2.1 Did you find any traces of the hidden artifacts and/or the files originally discovered in João Musk's sigma account on his computers?

Yes, we did. They are located in the folder **"backup_1727368201.zip/home/johnnymusk/Desktop/TVShows"** as well as in **"root/home/johnnymusk/.cache/thumbnails/fail/normal", "root/home/johnnymusk/.cache/thumbnails/fail/large","root/home/johnnymusk/Documents", "root/home/johnnymusk/Music".**

1. If so, can you trace the origin of these files and how they were processed over time? Construct a timeline of relevant events.

2. Did you uncover any evidence of anti-forensic activities?

Yes, João Musk's backups compressed files in the backup disk image were password protected.

Also, we see in the keylogger log files **"johnnyDisk\Partition 2\ext4\[root]\tmp\K5rb9cnL0Is.log"** and **"johnnyDisk\Partition 2\ext4\[root]\tmp\ KFP7oy1K7O5.log"** that he used the command **srm**, which is used for securely deleting files and directories on Unix OS to manipulate the andromeda.png and cartwheel.tiff files.

The entire folder on the path **"johhnyDisk\ Partition 2\[root]\home\johnnymusk\stt"** is comprised of scripts and files used to perform anti-forensic activities, e.g., **exploit.py** and **createChunks.py.**

2.2 What new discoveries can you report that might clarify the plot or identify other relevant actors?

The **User_Manual.pdf**, might indicate that David Alexandre Ferreira da Silva was also possibly involved in the Ariana 6 project. There is another unidentified IST student whose student number is **ist90834**, that appears in multiple backup folders as a log file of connection to address 10.0.2.166, which is one of sigma cluster's server, which could be the person with username **RootKitty** with whom João Musk communicates with in the IRSSI chat room, evidenced by the log file in **"johnnyDisk\Partition2\ext4\[root]\home\johnnymusk\snap\irssi\common\irclogs\2024\freenode\#thebasement.09-26.log".** The conversation in this chat indicates that that both of them are against the MK-Ultra program.

3 Appendices