INSTITUTO SUPERIOR TÉCNICO

Departamento de Engenharia Informática

Forensics Cyber Security

MEIC / METI 2025-2026 – 1st Period

Digital Forensics Report Lab2

Group number:	Name	IST Number
Student 1:	Olha Buts	116276
Student 2:	Vasco Conceição	106481
Student 3:	Nuno Martins	107273

1 Acquired artifacts

Name	Type	SHA-256 Value
backup_1758589201.zip	ZIP Archive	6ccd58ef578e0c2d13c5e47c9af78ab274aff059a42422872902a11f9e1689d7
backup_1758588601.zip	ZIP Archive	a9e69f5431616bfe086748105c8a23360c807635de076bedc6b1e4118ea9efd9
backup_1758588001.zip	ZIP Archive	4fbf84cd93743ba7b732f05aaf93d17f8d03eb197366a93aff23129cc18b0ade
backup_1758587401.zip	ZIP Archive	8819b97c7af26b2983b8b0130d0d578dc1b9c40e2fa19aab7aa58cc704a1c471
backup_1758586801.zip	ZIP Archive	ebbdba595a739862685e16c2e95de119de3f957b87da8d6e9ecd70612c3cf6c8
backup_1758586202.zip	ZIP Archive	8de6d689252e61c65c930762f0be8650a8ee4c1dd7931bbf5cbad2e8fdb02f5d
backup_1758585623.zip	ZIP Archive	f42a540046f87d7745c6a06fbc706d2dbdb441319de10d3196b80256ace87393
backup_1758495226.zip	ZIP Archive	eb2956ad880fd0ef02ed9b124f6635ea2dfdb53886d5b3cd88e4f8f96d107e71
Dv34.js	JS File	3cd0b50e08450c75cb3e77d1e4e3b5fd30932393f3e5c10fe30cf1fab3f0a136

entries.json	JSON File	1797ec14bcabfddac1264c5761ad3c86a2281163f076d77f87f5f819e0df67ee
N9hg.js	JS File	e4549908b21c4feec4bf73e06fd4bd96629eaba791f277701c29056e30961119
deploy.sh	Bash Script	d74a4123e4299e5aef90b74bee62bd314b5d2f437ad09ab3f31fac65da19d314
backups.sh	Bash Script	1d4d463a45f75fda04a8d5af8e0d21b4f0fa8f640ae5597ede13f19b823f46d8
#thebasement.09-23.log	IRC Log	8752362a3747ebf027b86864b2950e427b1688cc343d9257682538945
pass_gen.sh	Bash Script	83363303d40cbcef23466bb540d7a5bbe0f574d87db3655e01525a01dd4f2503
K306ao9voUZ.log	Keystroke Log	68e1ed95b0fdd91831a913fd27a9c4e314bdf46f8f3a05eac1e3ba57910bc8a4
Bash_history	Log File	c368617ef9982ae9be3f776584ac1dae533e83b9460dbc10da5c14b460153324
syslog	System Log	a639407dc14b8de805773a7decb414a0700ab8b4b1f4e98c37b71aec4e7d8fb6
history.log	APT Log	a639407dc14b8de805773a7decb414a0700ab8b4b1f4e98c37b71aec4e7d8fb6
recovered_inbox.txt	Text File	a833bceceb829721591585eae6fe3dd8a572a17452da8db6088fe2251bf940cb
source.py	Python File	c31987bce7265cdacd3329769acada11b26f8d57cc6a9676e3a6dda3b5c90200
Exploit.htpl	HTML Template	0f9fe89cc0108f9e1b933bd8daff74dfbde17444ce737665b7f1bac7c2df9b06
cookieSteal.html	HTML File	9a65dadabe142ceb15bbe5232c00a411571e85efedb88ab3f808e5cd7c776a53
reflectedXSS.html	HTML File	68b02362b9d62c67ba19ded67ca5bdc16ad78811162a8bd7cb463a79a40b6fb5
exploit.py	Python File	ccbfd54f6c3fe06c214aea28927a76a7f508269f76c3e5ef3692a8ad2022ae3b
obfuscator_decompiled.py	Python File	782d396713f7c6f85331c29c5394ac5c85e35a3167279d0401c4d4a5dac7438e
Seed.txt	Text File	98b073309a566415455c7f57b56b6d280fde814147dae61363363632e5a5fc47
Auth.log	System Log	47c8b7c3132827f578b68ed07fc90e447a3d1a4f70bad6b40945dc1e06f3e52a

Webhistory.txt	Text File	6218d229a19986930cb6fac470df641dd35e99b57b4f2aff3792c0c3cee7ea3f
obfuscator	Python Bytecode	86a89e4c96282492bbabb364b5a601a9e187d8f9365c3c1c900c79c7db244560
Places.sqlite	Database	30faeefb762d91a45a371bad4e4cecb274f13d614b75d9235cdf47b265995a8f
recently-used.xbel	XML Bookmark	6f661b31f9df96cefd9f1de6f96fc5e4247bc1c22a9f6d78614fcd9e39176c7a
Password.kdbx	Database	076f4fea500f73b89e50530482194e76867f403620c13b261e9458f53e8f9739
.gitconfig	Config File	345fdb845a85735f71775dd873d17dc6c36a98fa2b07325124bffde8ee71a983
.ssh/config	Config File	41999c1f215420945e1b5e3173d8a5db93e1e35b461376362c7e667b83677bbf

2 Report of all findings

Artifacts were extracted from the seized disk images (*johnny.dd* from the workstation and *backup.dd* from the backup server). The workstation disk was mounted read-only, and files were analyzed using *file, cat, stat, ls and "mousepad"* across directories. For the backup server, we employed *The Sleuth Kit (TSK)* tools *fls* and *icat* to extract files from the disk image. Despite anti-forensic measures (e.g., *srm*), evidence was recovered from caches, logs, and backups, with timestamps validated against */var/log/auth.log, /var/log/syslog,* and *bash history*. Below is a tabulated summary of key artifacts, including how they were located.

Name	Discovery Method and Context	Evidentiary Value
backup_1758589201.zip	Extracted from /home/ironcaesar/backups/ on backup using fls and icat; unlocked with derived password	Contains ZYRA site snapshot, linking to phishing deployment.
backup_1758588601.zip	Extracted from /home/ironcaesar/backups/ on backup using fls and icat; unlocked with derived password	Holds hidden artifacts (e.g., Invoice.pdf) and tools, confirming concealment.
backup_1758588001.zip	Extracted from /home/ironcaesar/backups/ on	Holds hidden artifacts (e.g., diff.txt) and tools.

	backup using fls and icat; unlocked with derived password	
backup_1758587401.zip	Extracted from /home/ironcaesar/backups/ on backup using fls and icat; unlocked with derived password	Supports backup routine evidence.
backup_1758586801.zip	Extracted from /home/ironcaesar/backups/ on backup using fls and icat; unlocked with derived password	Indicates regular backup schedule.
backup_1758586202.zip	Extracted from /home/ironcaesar/backups/ on backup using fls and icat; unlocked with derived password	Indicates regular backup schedule.
backup_1758585623.zip	Extracted from /home/ironcaesar/backups/ on backup using fls and icat; unlocked with derived password	Indicates regular backup schedule.
backup_1758495226.zip	Extracted from /home/ironcaesar/backups/ on backup using fls and icat; unlocked with derived password	Early evidence of site development.
Dv34.js	Recovered from home/johhnymusk/.config/Code/User/History/34ce7173/ on workstation opened with “mousepad”	Phishing server.js remnant, shows QR code logic.
entries.json	Same directory; opened with “mousepad”	Logs phishing code modifications.
N9hg.js	Same; opened with “mousepad”	Another server.js backup, reinforces phishing evidence.
deploy.sh	Located in /home/johhnymusk/scripts/ on workstation via ls and cat; executed	Deploys ZYRA site to INESC-ID, key to attribution.
backups.sh	Found in /home/johhnymusk/backups/ on workstation; traced via rsync logs	Manages encrypted backups and links to server activity.
#thebasement.09-23.log	Identified in /snap/irssi/common/irclogs/2025/freenode/ on workstation;	Reveals “CatDucky” collaboration and motives.
pass_gen.sh	Located in /home/johhnymusk/backups/ on workstation; decompiled obfuscator	Generates zip passwords, critical for decryption.

K306ao9voUZ.log	Hidden in /var/tmp/ on workstation;	Captures “ILoveMyMomTheQueen”and code edits.
obfuscator	Found in /home/johhnymusk/backups/ on workstation; decompiled online	Implements password hashing, unlocks zips.
Bash_history	Located in /home/johhnymusk/ on workstation	Shows site edits, hiding, and deletions
syslog	Retrieved from /var/log/ on workstation; parsed	Records power-off and network activity.
history.log	Found in /var/log/apt/ on workstation; grep searched	Logs secure-delete install.
recovered_inbox.txt	Found in Firefox cache	Contains CatDucky's tip email.
source.py	Recovered from /home/johhnymusk/scripts on workstation using ls and cat	Matches ZYRA attack vector (ChaCha20).
Exploit.htpl	Recovered from /home/johhnymusk/scripts on workstation	Phishing exploit code.
cookieSteal.html	Recovered from /home/johhnymusk/scripts on workstation	Indicates credential theft mechanism.
reflectedXSS.html	Recovered from /home/johhnymusk/scripts on workstation	Reflects XSS attack vector.
exploit.py	Recovered from /home/johhnymusk/scripts on workstation	Contains RCE via pickle, ties to phishing.
obfuscator_decompiled.py	Decompiled from obfuscator.pyc; analyzed manually	Reveals password generation logic.
Seed.txt	Found in /home/johhnymusk /backups/ on workstation; file search	Provides number, seed for password derivation.
Auth.log	Retrieved from /var/log/ on workstation; parsed	Confirms sudo commands and session data.
Webhistory.txt	Extracted from places.sqlite; SQL queried	Shows GitHub and QR phishing research.
Places.sqlite	Found in Firefox profile on workstation; SQL analyzed	Web history linking to attack planning.
recently-used.xbel	Located in/home/johhnymusk / .local/share/ on workstation; XML parsed	Records access to artifacts.

Password.kdbx	Found in /home/johhnymusk/ on workstation; opened with KeePassXC	Contains "TheByteOf78" seed, critical for decryption.
.gitconfig	Located in /home/johhnymusk/ on workstation; file search	Links "muskyboi" to GitHub activities.
.ssh/config	Located in /home/johhnymusk/ .ssh/ on workstation; file search	Authorizes "muskyboi" for INESC-ID and backup server.

3 Analysis of relevant findings

3.1 Do the seized disk images contain traces of the phishing website (source code, deployment scripts, logs) and/or the hidden artifacts discovered in Lab 1? If so, identify them, their locations on disk, and their SHA-256 values as recovered from the images.

Our analysis of disk images from João's workstation and backup server confirmed that both disk images contain traces of the phishing website.

```
rm fst-gang.png
rm wordlist.txt
ls -l
cp ~/Pictures/RaceCar.png .
python3 HideFiles/lsb-complex.pyc -m hide -d diagonalup -c rgb -n 4 -o RaceCar.png -p TouristThreads.png -e png
rm TouristThreads.png
rm -rf HideFiles/
rm ~/Pictures/RaceCar.png
rm .gitkeep
deactivate
keepassxc &
code ~/Documents/ZYRA
ls -l
git add --all
git commit -m "finally finished"
git push
cd ~/scripts/
chmod +x deploy.sh
./deploy.sh
irssi
cd ~/Documents/
sudo apt install secure-delete
srm -r *
cd ..
ls -l
srm zyra.zip
```

Figure 1 - Bash history

We started by looking at user activity logs and code editors since the bash history (see Figure 1) mentioned editing files in VS Code. Although *home/johnnymusk/Documents/Zyra* was wiped, the VS Code caches and backups on the workstation kept some traces alive and the backup server had complete zipped copies of the files.

We discovered files named *Dv34.js* and *N9hg.js*, which are copies of the original *server.js* by examining VS Code's history files located at */home/johhnymusk/.config/Code/User/34ce7173/History/.*

The *entries.json* file showed edits made on September 09 at 01:03 and 01:59 UTC. (SHA-256: 3cd0b50e08450c75cb3e77d1e4e3b5fd30932393f3e5c10fe30cf1fab3f0a136 - *Dv34.js*

1797ec14bcabfddac1264c5761ad3c86a2281163f076d77f87f5f819e0df67ee - *entries.json*

e4549908b21c4feec4bf73e06fd4bd96629eaba791f277701c29056e30961119 - *N9hg.js*)

```
const express = require("express");
const fs = require("fs");
const path = require("path");
const serveIndex = require('serve-index');
const QRCode = require('qrcode')

const app = express();
const PORT = 3000;

// Middleware to parse form data
app.use(express.urlencoded({ extended: true }));
app.use(express.json());

// Serve static files
app.use(express.static(__dirname));

// login get
app.get("/login", (req, res) => {
  res.sendFile(path.join(__dirname, "public", "index.html"));
})

// endpoint para gerar o QR
app.get("/qrcode", async (req, res) => {
  try{
    const url = `http://zyra.csf.syssec.dpss.inesc-id.pt/login`;
    const qr = await QRCode.toDataURL(url);
    res.send(`<img src="${qr}">`);
  } catch (err) {
    res.status(500).send("Erro a gerar QR code");
  }
});

// Handle login POST
app.post("/login", (req, res) => {
  const { email, password } = req.body;

  // Save to credentials.txt
  const filePath = path.join(__dirname, "credentials.txt");
  const line = `Email: ${email}, Password: ${password}\n`;

  fs.appendFile(filePath, line, (err) => {
    if (err) {
```

Figure 2 - Dv34.js

For the deployment, there's a script called *deploy.sh* (SHA-256 - d74a4123e4299e5aef90b74bee62bd314b5d2f437ad09ab3f31fac65da19d314), a Bash file that zips the website, sends it to the INESC-ID machine at csf*:146.193.41.109*, and runs *node server.js*. The bash history revealed commands like chmod *+x deploy.sh* and *./deploy.sh* localized in the *home/johnnymusk/scripts/* directory.

```
#!/bin/bash
set -e

echo "Zipping Zyra Website"
cd ~/Documents
zip -r ~/zyra.zip ZYRA

echo "Copying Zip To INESC machine"
scp ~/zyra.zip csf:/home/muskyboi/

echo "Initializing Zyra Website"
ssh -t csf "
    rm -rf Website/ZYRA; \
    unzip -o /home/muskyboi/zyra.zip -d Website; \
    cd Website/ZYRA; \
    npm install; \
    npm install vhost; \
```

Figure 3 - Deploy.sh

The backup process was handled by a script named *home/backup/backup.sh* (SHA-256 - 1d4d463a45f75fda04a8d5af8e0d21b4f0fa8f640ae5597ede13f19b823f46d8). This script creates a timestamp, sets up a user and host for the backup server, generates a password using *home/johnnymusk/backup/pass_gen.sh* (SHA-256 - 83363303d40cbcef23466bb540d7a5

bbe0f574d87db3655e01525a01dd4f2503), zips the *~/Documents/ZYRA/* folder with that password, syncs it to the backup server, and then removes the local zip file.

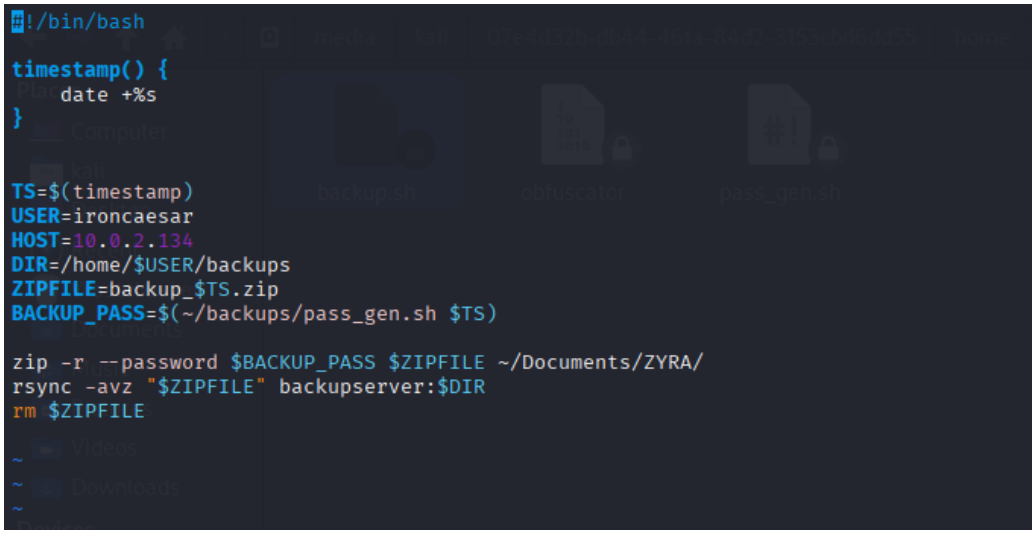


Figure 4 - Backup.sh

We also found logs related to the phishing setup. These include IRC conversations in */snap/irssi/common/irclogs/3035/freenode/#thebasement.09-23.log* (SHA-256 - 0ed38c4

8752362a3747ebf027b86864b2950e427b1688cc343d9257682538945)}, where João talks about hosting the site, and web history from *places.sqlite* (sha256 - 30faeefb762d91a45a371bad4e4cecb274f13d614b75d9235cdf47b265995a8f}, which shows searches for tools on GitHub.

In the folder */home/johnnymusk/.ssh*, we found a *config* file (SHA-256 - 41999c1f215420945e1b5e3173d8a5db93e1e35b461376362c7e667b83677bbf) containing accounts information, as well as RSA keys (both public and private) indicating authorization for the muskyboi account. We also found a .gitconfig file with the same username.

```
Host csf
    HostName 146.193.41.109
    IdentityFile ~/.ssh/muskyboi
    User muskyboi
    Port 22
    StrictHostKeyChecking no
Host github.com
    User MuskyBoi
    IdentityFile ~/.ssh/csf-johnny
    IdentitiesOnly yes
Host backupserver
    HostName 10.0.2.134
    User ironcaesar
    IdentityFile ~/.ssh/id_rsa
    IdentitiesOnly yes
```

Figure 5 - .ssh/config

During our investigation, we came across several images that played a role in hiding information that served as containers for steganography or were used on the site. Many of them were found in the .cache*/thumbnails/* directory, with subfolders *normal* and *large*. The *large* folder had various PNG images, including BarragemFagilde and BarragemOdelouca, which were used to hide the algorithm *diff.txt*.

Phishing Website (ZYRA clone): Backup server has zipped versions in /home/ironcaesar/backups/ (e.g., backup_1758589201.zip). SHA-256 of backup_1758589201.zip: 6ccd58ef578e0c2d13c5e47c9af78ab274aff059a42422872902a11f9e1689d7.

```python
# Visit https://www.lddgo.net/en/string/pyc-compile-decompile for more information
# Python 3.8.0 (3413)

import hashlib, sys
SEED_PATH = "/tmp/seed.txt"
with open(SEED_PATH, "r") as f:
    line = f.readline()
if len(line.split("\t")) == 2:
    n = int(line.split("\t")[0])
    seed = line.split("\t")[1].strip()
    if n == 0:
        print("For the first run, please just place your password in the " + SEED_PATH + " file.")
        exit(1)
else:
    n = 0
    seed = line.strip()
pw = hashlib.sha256(str(seed + str(sys.argv[1])).encode("utf-8"))
print(pw.hexdigest())
next_seed = hashlib.sha256(str(seed).encode("utf-8"))
with open(SEED_PATH, "w") as f:
    f.write(str(n + 1) + "\t" + next_seed.hexdigest())
```

Figure 6 - Obfuscador_decompiled

The **artifacts** were hidden using steganography from the *HideFiles* GitHub repository. We found them as thumbnails on the workstation and on the backup server. The passwords for the zips were dynamically generated using *pass_gen.sh*, which calls the *obfuscator* script with timestamps. We uncovered the master password "ILoveMyMomTheQueen" in the keystroke log, which was captured while *lsb-complex.pyc* was running at */var/tmp/K306ao9voUZ.log* (SHA-256 - 68e1ed95b0fdd91831a913fd27a9c4e3

14bdf46f8f3a05eac1e3ba57910bc8a4, where it was typed while using KeePassXC (noting the shift key presses). Opening the *Passwords.kdbx* file with KeePassXC revealed "TheByteOf78" as the initial seed. Using the *seed.txt* file, which showed n = 76 and a seed hash, and decompiling *obfuscator.pyc* with an online tool, we simulated password generations in python. This process helped us unlock the zips. For example, the password for *backup_1758586801.zip* was calculated as 78b4d2bd8a5a270605194e5f610094e51a4a2ce008710510bb00c0c59fc16988, derived from hashing "TheByteOf78" iterated 74 times with the timestamp 1758586801. The SHA-256 sum of *backup_1758586801.zip is* a9e69f5431616bfe086748105c8a233

60c807635de076bedc6b1e4118ea9efd9.

```python
#!/usr/bin/env python3
import hashlib

seed0 = "TheByteOf78"
seed76 = "6b508266ad2ab9fd7fef0ecfca4b9d874ed3e0362bbf92cf2b269b0f632f452b"
timestamp = "1758588601"
n_zip = 74
s = seed0
for i in range(1, 77):
    s = hashlib.sha256(s.encode("utf-8")).hexdigest()
    if i == n_zip:
        pw = hashlib.sha256((s + timestamp).encode("utf-8"))
        print(f"pw: {pw.hexdigest()}")

print("seed_76 matches target?", s == seed76)
```

Figure 7 - Screenshot of code used to discover ZIP password

The **FinancialReport.pdf** file was referenced in the *recently-used.xbel* (sha256 - 6f661b31f9df96cefd9f1de6f96fc5e4247bc1c22a9f6d78614fcd9e39176c7a) bookmark XML, confirming it existed at */home/johhnymusk/Documents/ZYRA/resources/FinancialReport.pdf* and was opened with Evince on September 09 at 00:34:52 UTC. The bash history shows it was later deleted with *srm*, which removed its traces from the filesystem. However, this bookmark

reference remains as proof of its existence after deletion. On the backup server, we could extract it from *backup_1758586801.zip* in the */home/johhnymusk*/ZYRA*/resources/FinancialReport*.pdf

(SHA-256 - fde6e8582b5b4763e4e574c5eb293876f5894339215d12bffd224f929e559cc9)

after unlocking zip with the calculated password.

Similarly, the **Invoice.pdf** file, a Futuro Inteligente invoice, was opened with Evince on September 09 at 00:35:43 UTC, as noted in *recently-used.xbe*l. The bash history indicates it was hidden using *hide_base64.sh Invoice.pdf ~/scripts/steghide*, then deleted along with *~/scripts/steghide* and *Invoice*. We recovered the unmodified *steghide* file from */home/johhnymusk/. cache/vmware/drag_and_drop/N2JdwH/steghide*, with a SHA-256 value of eb98a185541c4be01f523a264c923944cd47a21fb50c190aacc1abce334b9db9.

On the backup server, Invoice.pdf (SHA256 -748e7358ccbba0b769b94ecb3c96c161f

d8366454acf076a6f95b5f404515e1d) is extractable from the backup_1758586801.zip.

The **parking.db** database extract was initially verified on September 09 at 00:37:52 UTC, then embedded in MingleGame.wav and deleted with *srm*. Recovering remnants on the workstation was challenging due to the secure deletion, but the original is recoverable from */backup_1758588001.zip* on the backup server (SHA256 -23108d6552bbd633ed80c

3de50449e0698294c0da3a049a61bbaed41e0edcea0). The unmodified MingleGame.wav can still be found at */home/johhnymusk/.cache/vmware/drag_and_drop/DJ6jmh/MingleGame.wav*. (SHA-256 - cfaad93ccd6e2891b5c1fb1f4d8fa2cb876797daa595ecc3e1f231a85a1ccd27).

The **TouristThreads.png** (Reddit thread) and **wpp.png** (WhatsApp conversation) were opened on September 09 at 00:37:37 and 00:39:07 UTC, respectively, according to *recently-used.xbel*. Despite being deleted later, the system saved thumbnails located at */home/johhnymusk/. cache/thumbnails/normal*. These thumbnails have different SHA-256 values due to resizing: 3cc8df3b8781ad72490ecb32c5e29dff0c772bd5928da809f190

a090094b6730 for *TouristThreads.png* and b90aec208015897d5740dd87629f97d3d789

361ac84d3a64902fcce40aafe7ac for *wpp.png*. The originals are located on the backup server at

/backup_1758586801.zip/home/johhnymusk/ZYRA*/resources/. Their SHA-256 hashes are c938a05b84f2d749fc6ac65a88f57ccc0c860d4169577947425cc01aa1bbd1c3 for* wpp.png and ff56d339f5affe0279530214dec0ef01950828beb4af6bfa83fe8465a71da7d8 for TouristThreads.png

Finally, the **diff.txt** was hidden in /Pictures/BarragemFagilde.tiff and /Pictures/BarragemOdelouca.jpg. Like the other artifacts, **diff.txt** (SHA256 – 80373bc7994731717f3fe753685848541bb81cbf906a85c3f42415e5ed05bd86) is recoverable from the backup server (*/backup_1758586801.zip/home/johhnymusk/*ZYRA*/resources/diff.txt.*

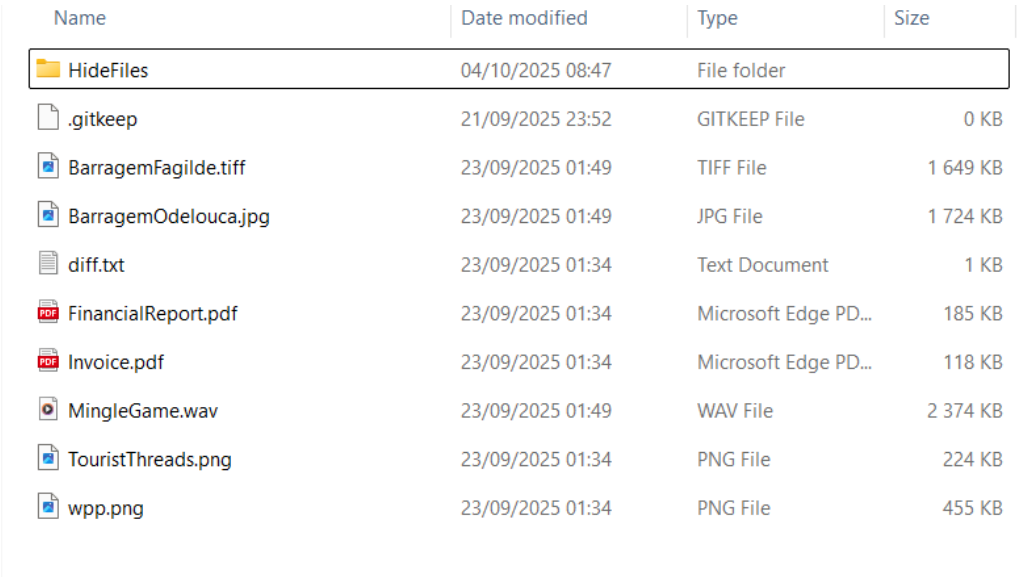


Figure 8 - Folder structure of Zyra/resource

3.2 Can you reconstruct the provenance and handling of these files? Provide a timeline of relevant events that shows how the phishing site was built and operated, and how the hidden artifacts were managed on the seized machines.

Time	Description	Source
2025-09-04 01:11:25	Boot: VMware Tools services are initialized, the kernel detects the “VMware Hypervisor”, creates virtual devices, and enables drag-and-drop functionality	Var/log/syslog (2025-09-18T00:11:25.563805+00:00 johnnymusk systemd-modules-load[397]: Inserted module ‘lp’ 2025-09-18T00:11:25.566303+00:00 johnnymusk systemd-modules-load[397]: Inserted module 'ppdev' , …)
2025-08-11 17:34	On the host, several scripts are created exploit.py, exploit.htpl, cookieSteal.html, reflectedXSS.html,source.py, wordlistFromPdf.py, deploy.sh.	home/johhnymusk/scripts (stat -> Modify)
2025-09-02 23:20–23:43	Several documents and images with original timestamps (e.g. Teorica_6_2017_2018.pdf, Teorica_7_2017_2018.pdf, .jpg/.gif)	Stat
2025-09-07 21:00–21:01	Drag and Drop import of scripts (created 08/25) + PDFs and images.	drag_and_drop/ (stat -> Modify)
2025-09-07 23:50	deploy.sh created	home/johhnymusk/scripts/deploy.sh
2025-09-07 23:50	Creating backup.sh in home/johhnymusk /backup	home/johhnymusk/backups/backup.sh
2025-09-07 22:53:55	Git clone: from https://github.com/MuskyBoi/ZYRA.git	.git/logs/refs/main (backup server)
2025-09-09 01:03:16	Johnny opened *VSCode* using the command “*code .*” to edit the phishing site’s code	Var/log/syslog (2025-09-23T00:03:16.446497+00:00 johnnymusk systemd[2762]: Started snap. code.code-6315aa92-b94c-4baf-b102-5808a1b306b5.scope.), Bash_history (code.) and .config/Code/logs/20252923T010317/editSes

		sions.log (2025-09-23 01:03:23.991 [info] Prompting to enable ..)
2025-09-09 01:03:39	Johnny opened Firefox and started browsing official Formula 1 ticket sites, exploring options for upcoming races and grandstands. Afterward, he shifted gears to YouTube, searching for jazz music (ambient studying aid.)	Places.sqlite (extracted timestamps and URLs; converted from local 01:03:39)
2025-09-09 01:05:17	Johnny joins IRC channel #thebasement as "johhnymusk" and converses with “CatDucky”. He discusses a broken leg from a ZYRA bike incident, expresses intent to create a phishing site to steal ZYRA client credentials, and mentions hosting it on a “Syssec” machine (zyra.csf.syssec.dpss.inesc-id.pt) under Professor Nuno's supervision for his thesis. Cat warns against illegal actions.	#thebasement.09-23.log (--- Log opened Tue Sep 23 01:05:17 2025 01:05 -!- johhnymusk [~johhnymus@freenode-uip.cj2.jjp4uo.IP] has joined #thebasement)
2025-09-09 01:10:00	“CatDucky” receives an anonymous email (from totallyaveragedude@tutamail.com via Proton Mail) revealing MELE's “smart parking” system is rigged (tourists overcharged, allies spared via premium zones and ghost company “Futuro Inteligente, Lda.”). The email directs her/him to retrieve proof files from a physical locker at CTT IST Lisbon (Locker 03, Code: 666).	Home/johhnymusk/thunderbird/common/.thunderbird/kl1zswcm.default/Mail/pop.gmail.com/inbox
2025-09-09 01:10:00 - 02:00:00	Periodic execution via CRON of the backup script located at home/johhnymusk /backups/backup.sh	Var/log/syslog (2025-09-23T00:10:02.013942+00:00 johnnymusk CRON [23244]: (johhnymusk) CMD (sh ~/backups/backup.sh) 2025-09-23T00:20:01.836349+00:00 johnnymusk CRON [23537]: (johhnymusk) CMD (sh ~/backups/backup.sh), ..)

2025-09-09 01:20:00	An anonymous email originally sent to itscatducky@proton.me was subsequently forwarded to sussyjohnnyhacker.	Home/johhnymusk/thunderbird/common/.thunderbird/kl1zswcm.default/Mail/pop.gmail.com/inbox
2025-09-09 01:21:44	Johnny rejoins the IRC channel #thebasement and questions the email he received.	Home/johhnymusk/irssi/common/irssilogs/2025/freenode / #thebasement.09-23.log
2025-09-09 01:33:49	Johnny retrieves a USB drive labeled “Intenso” with serial number *3D32E2CD* from the locker. The device is mounted at /media/johhnymusk/Intenso. He copies several sensitive files to the directory ~/Documents/ZYRA/resources/, including: diff.txt, FinancialReport.pdf, Invoice.pdf, parking.db, TouristThreads.png and wpp.png.	Var/log/syslog (2025-09-23T00: 33:49.838174+00:00 johnnymusk kernel: usb 2-1: New USB device found, idVendor=058f, idProduct=6387, bcdDevice= 1.03) and bach_history (cp /media/johhnymusk/Intenso/* .)
2025-09-09 01:34:52-01:39:07	He views artifacts using commands like cat diff.txt, xdg-open for PDFs/images, and code parking.db (likely to inspect contents).	home/johhnymusk/.local/share/recently-used.xbel (<bookmark href="file:///home/johhnymusk/Documents/ZYRA/resources/FinancialReport.pdf" added="2025-09-23T00:34:52.935690Z" modified="2025-09-23T00:34:52.935784Z" visited="2025-09-23T00:34:52.935691Z">) and bash_history (cat diff.txt)
2025-09-09 01:40:08	Browsed the Portugal Smart Cities Summit 2025 website (turismodeportugal.pt)	Places,sqlite(2025-09-23 01:40:08|https://business.turismodeportugal.pt/pt/Agenda/Eventos/Paginas/portugal-smart-cities-summit.aspx|Portugal Smart Cities Summit 2025|2)
2025-09-09 01:40:20	Visited the Barracuda blog discussing phishing attacks using QR codes.	Places.sqlite (https://blog.barracuda.com/2024/10/22/threat-spotlight-evolving-qr-codes-phishing-attacks|Threat Spotlight: The evolving use of QR codes in phishing attacks | Barracuda Networks Blog)
2025-09-09 01:41:00	Johnny rejoins IRC #thebasement, informs Cat he has proof of MEMEL rigging (preferential plates, unpaid parking, money funneled to	Home/johhnymusk/irssi/common/irssilogs/2025/freenode / #thebasement.09-23.log

	Futuro Inteligente). Cat suggests hiding files using a GitHub repo like "shadowexploit" (FileHider/HideFiles).	
2025-09-09 01:44:24	Johnny searches for 'HideFiles' repositories (matching Cat's suggestion; likely identifies shadowexploitthunder-web/HideFiles.git).	Places.sqlite:(2025-09-23 01:44:36|https://github.com/search?q=HideFiles&type=repositories|Repository search results · GitHub|1)
≈2025-09-09 01:46:00 - 02:00:00	Johnny clones the HideFiles repo (git clone https://github.com/shadowexploitthunder-web/HideFiles.git), views its code (code .), makes scripts executable (chmod +x), and hides the leaked files using steganography: - parking.db hidden in ~/Music/MingleGame.wav via wav_hider.py. diff.txt hidden in TIFF/JPG images via hide_in_comments.sh. - Invoice.pdf hidden in ~/scripts/steghide via hide_base64.sh (venv activated). wpp.png hidden in fst-gang.png via lsb.py. FinancialReport.pdf converted to PNG and hidden. TouristThreads.png hidden in RaceCar.png via lsb-complex.pyc. Additional steps: Generates wordlist from Modern_Cars.pdf, hides in ZIP via create_zip.sh. Deletes originals with rm, removes HideFiles repo (rm -rf HideFiles/), deactivates venv.	bash_history: (full command sequence post-git clone; inferred timing between GitHub browse and next edit).
2025-09-09-02:01:01	Final edits to server.js; git commit/push.	entries.json. Keystroke log (typing code like const qr = await QR..., git commit -m [shift]@finally finished[shift]@, git push[enter][shift]MUsky[shift]Boi). recovered_bash_history.txt (git commands).
2025-09-09-02:03:01	Johnny deploys the phishing site via ~/scripts/deploy.sh.	bash_history: (cd ~/scripts/, chmod +x deploy.sh, ./deploy.sh)

2025-09-09-02:04:00	Johnny rejoins IRC, tells Cat “All is ready” and that “ZYRA is gonna suffer.” Cat suggests using srm for secure deletion.	Home/johhnymusk/irssi/common/irssilogs/2025/freenode / #thebasement.09-23.log
2025-09-09 02:06:00	Johnny install secure-delete package, entering sudo password “squidGamer”.	var/log/auth.log (sudo command at 2025-09-23T01:06:00+00:00). Keystroke log (secu[tab]-delete[enter]squid[shift]Gamer[enter])
≈2025-09-09 02:40:00	Johnny deletes files: srm -r * in ~/Documents/, then srm zyra.zip.	bash_history. (final commands). Keystroke log (srm -r [shift]{[enter], srm [shift]zy[tab][enter])
2025-09-09 03:43:31	System shutdown initiated, ending session.	Var/log/syslog (power off messages at 2025-09-23T02:43:31+00:00) and auth.log (session closed at same time).

3.3 Did you find any evidence of anti-forensic activity? If so, describe the techniques observed and the evidence that supports your claim.

Yes, evidence of anti-forensic activity was found.

Secure Deleting Tool

The workstation had the “secure-delete” package installed and with “srm” command was used. This tool erases data fully, evading simple recovery.

Evidence:

- Installed via “sudo apt install secure-delete” (auth.log, history.log).
- Commands like “srm -r *” in bash history and keystrokes.

```
Start-Date: 2025-09-23  02:06:01
Commandline: apt install secure-delete
Requested-By: johhnymusk (1000)
Install: secure-delete:amd64 (3.1-8)
End-Date: 2025-09-23  02:06:03
```

Figure 9 - history.log

Steganography Tools

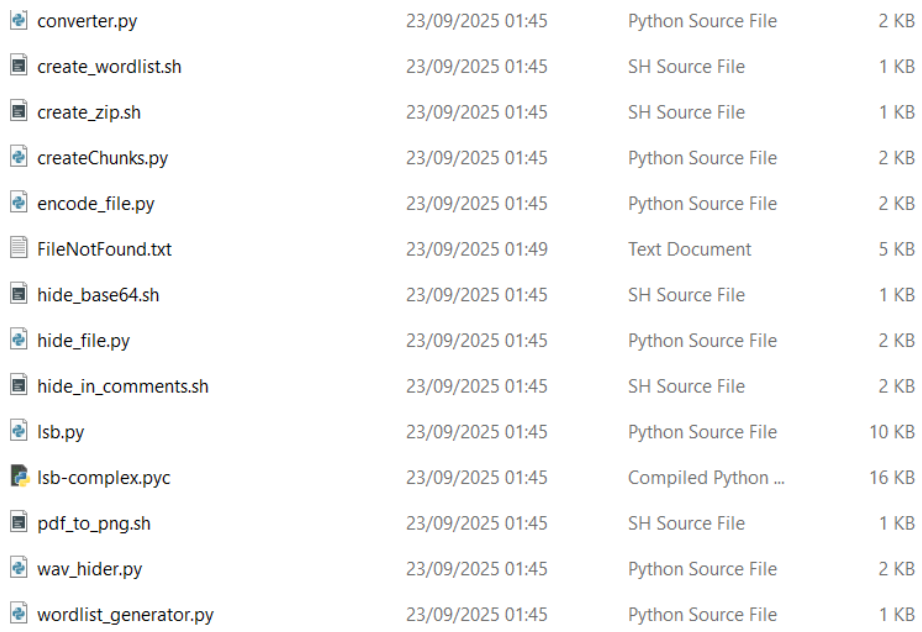
João cloned “HideFiles” repo from GitHub. Used scripts to hide files in images, audio, and more.

Techniques:

- LSB in images (lsb.py, lsb-complex.pyc).
- Base64 encoding (hide_base64.sh).
- Comments in images (hide_in_comments.sh).
- Audio hiding (wav_hider.py).
- PDF to PNG and ZIP hiding.

Evidence:

- Bash history shows cloning, chmod, and running scripts.
- Recovered hidden files from backup_1758588601.
- Keystrokes confirm deletions after hiding.



converter.py	23/09/2025 01:45	Python Source File	2 KB
create_wordlist.sh	23/09/2025 01:45	SH Source File	1 KB
create_zip.sh	23/09/2025 01:45	SH Source File	1 KB
createChunks.py	23/09/2025 01:45	Python Source File	2 KB
encode_file.py	23/09/2025 01:45	Python Source File	2 KB
FileNotFound.txt	23/09/2025 01:49	Text Document	5 KB
hide_base64.sh	23/09/2025 01:45	SH Source File	1 KB
hide_file.py	23/09/2025 01:45	Python Source File	2 KB
hide_in_comments.sh	23/09/2025 01:45	SH Source File	2 KB
lsb.py	23/09/2025 01:45	Python Source File	10 KB
lsb-complex.pyc	23/09/2025 01:45	Compiled Python ...	16 KB
pdf_to_png.sh	23/09/2025 01:45	SH Source File	1 KB
wav_hider.py	23/09/2025 01:45	Python Source File	2 KB
wordlist_generator.py	23/09/2025 01:45	Python Source File	1 KB

Figure 10 - backup_1758588601/HideFiles

QR Codes in Phishing Attacks

The suspect searched "The evolving use of QR codes in phishing attacks" on Barracuda blog. Used ideas for phishing site.

From blog: QR codes evade email filters by hiding URLs. Dynamic codes change links, hard to trace. Example: Fake login pages via QR.

Evidence:

- Web history: Visited https://blog.barracuda.com/2024/10/22/threat-spotlight-evolving-qr-codes-phishing-attacks.
- Code in server.js generates QR for fake login.

```
2025-09-23 01:44:39|https://github.com/search?q=HideFiles&type=repositories&p=2||1
2025-09-23 01:44:36|https://github.com/search?q=HideFiles&type=repositories|Repository search results · GitHub|1
2025-09-23 01:44:26|https://github.com/|GitHub · Build and ship software on a single, collaborative platform · GitHub|1
2025-09-23 01:44:24|https://www.google.com/search?client=ubuntu-sn&channel=fs&q=github|github - Pesquisa Google|2
2025-09-23 01:40:20|https://blog.barracuda.com/2024/10/22/threat-spotlight-evolving-qr-codes-phishing-attacks|Threat Spotlight: The evolving use of QR codes in phishing attacks | Barracuda Networks Blog|2
```

Figure 11 - web history

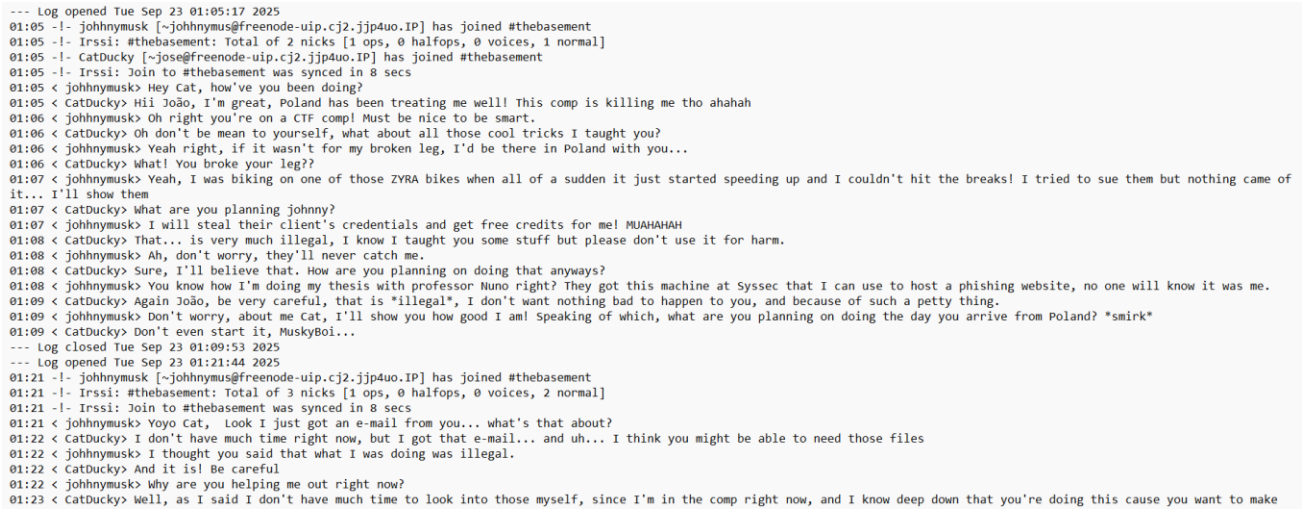
3.4 What new discoveries emerge from the seized images that were not visible in Lab 1? Do these findings strengthen the attribution of the phishing attack to João Musk, or point to other relevant actors (collaborators, external servers)? Based on your analysis, what hypotheses can you form about the larger scheme hinted at by the hidden artifacts, and what recommendations would you make for further investigative steps by Prof. Alexandre Magno and the Polícia Judiciária?

New findings:

- *IRC* logs reveal motive (ZYRA bike accident), collaboration (CatDucky aids hiding/deletion), shift to MEMEL exposure.
- Email (recovered_inbox.txt): Cat forwards anonymous tip (totallyaveragedude@tutamail.com) about fraud associated with Futuro Inteligente.
- *Web history* links to phishing research (QR code attacks) and GitHub searches for hiding tools.
- APT history shows recent installs like secure-delete, aligning with cleanup.
- Relevant PDFs (csf2526-*) are course materials on forensics, suggesting João studied evasion techniques.
- Keystroke log (K306ao9voUZ.log) captures phishing site code edits (e.g., QR code for ZYRA login).

- deploy.sh shows deployment to “csf.syssec.dpss.inesc-id.pt" - a university server, tying to his thesis.
- backups of the Zyra site are located on the backup server.
- Bash history.
- Phishing code in “source.py” and “exploit.py” mirrors ZYRA attack vectors (e.g., ChaCha20 encryption, RCE via pickle).

These strengthen attribution to João Musk. Additionally, nicks/emails match (johhnymusk, sussyjohnnyhacker@gmail.com). *IRC* logs and email highlights collaborators—CatDucky advised on hiding/deletion; an anonymous tipster (totallyaveragedude@tutamail.com) via forwarded email.



```
--- Log opened Tue Sep 23 01:05:17 2025
01:05 -!- johhnymusk [~johhnymus@freenode-uip.cj2.jjp4uo.IP] has joined #thebasement
01:05 -!- Irssi: #thebasement: Total of 2 nicks [1 ops, 0 halfops, 0 voices, 1 normal]
01:05 -!- CatDucky [~jose@freenode-uip.cj2.jjp4uo.IP] has joined #thebasement
01:05 -!- Irssi: Join to #thebasement was synced in 8 secs
01:05 < johhnymusk> Hey Cat, how've you been doing?
01:05 < CatDucky> Hii João, I'm great, Poland has been treating me well! This comp is killing me tho ahahah
01:06 < johhnymusk> Oh right you're on a CTF comp! Must be nice to be smart.
01:06 < CatDucky> Oh don't be mean to yourself, what about all those cool tricks I taught you?
01:06 < johhnymusk> Yeah right, if it wasn't for my broken leg, I'd be there in Poland with you...
01:06 < CatDucky> What! You broke your leg??
01:07 < johhnymusk> Yeah, I was biking on one of those ZYRA bikes when all of a sudden it just started speeding up and I couldn't hit the breaks! I tried to sue them but nothing came of
it... I'll show them
01:07 < CatDucky> What are you planning johnny?
01:07 < johhnymusk> I will steal their client's credentials and get free credits for me! MUAHAHAH
01:08 < CatDucky> That... is very much illegal, I know I taught you some stuff but please don't use it for harm.
01:08 < johhnymusk> Ah, don't worry, they'll never catch me.
01:08 < CatDucky> Sure, I'll believe that. How are you planning on doing that anyways?
01:08 < johhnymusk> You know how I'm doing my thesis with professor Nuno right? They got this machine at Syssec that I can use to host a phishing website, no one will know it was me.
01:09 < CatDucky> Again João, be very careful, that is *illegal*, I don't want nothing bad to happen to you, and because of such a petty thing.
01:09 < johhnymusk> Don't worry, about me Cat, I'll show you how good I am! Speaking of which, what are you planning on doing the day you arrive from Poland? *smirk*
01:09 < CatDucky> Don't even start it, MuskyBoi...
--- Log closed Tue Sep 23 01:09:53 2025
--- Log opened Tue Sep 23 01:21:44 2025
01:21 -!- johhnymusk [~johhnymus@freenode-uip.cj2.jjp4uo.IP] has joined #thebasement
01:21 -!- Irssi: #thebasement: Total of 3 nicks [1 ops, 0 halfops, 0 voices, 2 normal]
01:21 -!- Irssi: Join to #thebasement was synced in 8 secs
01:21 < johhnymusk> Yoyo Cat,  Look I just got an e-mail from you... what's that about?
01:22 < CatDucky> I don't have much time right now, but I got that e-mail... and uh... I think you might be able to need those files
01:22 < johhnymusk> I thought you said that what I was doing was illegal.
01:22 < CatDucky> And it is! Be careful
01:22 < johhnymusk> Why are you helping me out right now?
01:23 < CatDucky> Well, as I said I don't have much time to look into those myself, since I'm in the comp right now, and I know deep down that you're doing this cause you want to make
```

Figure 12 - IRC (#thebasement): Musk discusses ZYRA phishing revenge

Hypotheses: João Musk initiated a phishing attack against ZYRA as revenge for a bike accident that broke his leg. During this, he uncovered evidence of MEMEL's parking scam: an AI algorithm rigged to overcharge tourists while favoring allies with free spots, with profits laundered through Futuro Inteligente, Lda. Musk then shifted to whistleblower mode by stashing the evidence in files. “CatDucky” aided by sending files and advising on hiding and deleting them.

Recommendations for Prof. Magno and Polícia Judiciária:

- Raid locker for physical artifacts.
- Cross-check MEMEL/EMEL databases and logs against extracted artifacts (parking.db, diff.txt) for rigged AI evidence.

- Interview "CatDucky".
- Trace the anonymous Tutanota email (via subpoena if possible) for sender metadata and IP origins.
- Subpoena INESC logs (csf server), GitHub (MuskyBoi).
- Conduct network forensics.
- Interview ZYRA breach victims.
- Audit finances of Futuro Inteligente, Lda., including invoices and ties to MEMEL/EMEL.