Operation Cybershadow

David Dudas

Faculty of Mathematics and Computer Science, West University of Timisoara, Timisoara, Romania.

Corresponding author(s). E-mail(s): david.dudas03@e-uvt.ro;

Abstract

This paper present the operation CYBERSHADOW: A Digital Whodunit. This operation is about a digital forensics investigation triggered by an infiltration of the Ministry of Strategic Technologies. This paper presents four chapters that describe the investigation, the tools used, and the findings.

Keywords: Cybersecurity, Digital Forensics, Investigation, C++

1 Introduction

This paper tends to present operation CYBERSHADOW: A Digital Whoduint that started at InterSec Division HQ.

The Ministry of Strategic Technologies was inflitrated by an unknown actor and the only clues left are a compromised machine, garbled traffic logs, a suspicious USB image, and a stash of potentially synthetic media.

In the following sections I will present how I approached the problem, what tools I used, and what I found.

1.1 Chapter 1: Shadows in the File System

The investigation begins with a laptop with a compressed archive on its drive. This archive contains a web of directories and files that may contain clues. The filenames and types are misleading, so we will need a script that can recursively scan a folder and identify the actual file types.

The script should be designed for the Windows operating system and I will write it in C++.

The script will offer a command line interface with the following options:

- `-i` or `--input` followed by a path: Specify the input file path.
- `-s` or `--sigs` followed by a path: Specify the file type map path.
- `-d` or `--depth` followed by a number: Specify the search depth (when input path is a directory).
- `-h` or `--help`: Display the help message.

We can use this script to scan the archive and identify the actual file types.

1.2 Chapter 2: Listening to Ghosts

A machine started acting strange after the user installed something searching for Google Authenticator. There are two separate LAN segments, each tainted by infection. We will analyze the traffic for each.

1.2.1 Operation MOONFALL: The False Authenticator

The infection took root after accessing a fake Google Authenticator page. In order to find out **infected Windows client's IP address, MAC address, host name** and also the **likely domain name used by the fake Authenticator site and the Command and Control (C2) server IP address**, we will analyze the traffic logs.

1.2.2 Operation GREENWIRE: The Domain Breach

There are some outbound connections to obscure hosts, odd DNS behavior, and encrypted chatter in a different AD environment. Seems to be a different malware strain, but it might be linked to the previous analysis.

Here we are interested in finding out the **IP adddress and host name** of the infected Windows machine, the associated **user account names**, the responsible **malware family**, the **exact UTC timestamp** when the infection began, and the **domain name** used within this AD environment.

1.3 Chapter 3: Echoes from the Drive

An USB device is next clue. I wat profesionally wiped, but the imaging team was able to extract a complete forensic dump: IMAGE.ISO.

We also know that there is a file inside the image that was encrypted using a 2-byte XOR cipher, the archives may be nested within the image, and one file is believed to be an image (BMP, PNG, or JPG), but it also be more than just pixels.

I will try to **recover** all the extractable files from the ISO image, **classify** them by type with as much accuracy as possible, decrypt the encrypted file, and examine image files.

1.4 Chapter 4: Faces Behind the Curtain

There is also a folder that has surfaced during the investigation. It contains dozens of images and videos depicting high-ranking officials in scenarios that could ignite international crises. However, there is something suspicious about them.

I will analyze the media files and will try to **determine which images and videos are authentic and which have been manipulated**.

2 Related Work

What the heck am I supposed to write here?

3 Methodology

3.1 File Type Detection Script

In this section, I will present the C++ program used to detect the file types from the achive from chapter 1[1.1].

I will use the magic number of the file's headers to determine the file type. The magic number is a unique sequence of bytes at the beginning of a file that identifies its type. I will use Gary Kessler's magic number list[1] for this.

The program reads the json file using the nlohmann::json library[2] and populates a map where the magic number is the key and the file type details are the value. The program then recursively scans the input directory and reads the first eight bytes of each file. It checks if the magic number is present in the map and prints the file type details if it is found.

The program won't work if the input path and the signatures path are not provided. The depth is optional - if not provided, the program will scan the entire directory tree.

3.2 Network Traffic Analysis

3.2.1 The False Authenticator

3.2.2 The Domanin Breach

3.3 Digital Foresics & Hidden Data Extraction

3.4 Deepfake Detection

4 Results

4.1 File Type Detection Script Results

4.1.1 The False Authenticator Results

4.1.2 The Domain Breach Results

4.1.3 Digital Foresics & Hidden Data Extraction Results

4.1.4 Deepfake Detection Results

5 Conclusion

References

[1] Kessler, G.: File Signatures. https://web.archive.org/web/20250620072537/https://www.garykessler.net/software/index.html#filesigs (2025)

[2] Lohmann, N.: JSON for Modern C++. https://github.com/nlohmann/json (2025)