Contents

1. Introduction

1.1. A brief History on SnakeKeyLogger

Snake Keylogger is a highly sophisticated malware designed to steal sensitive information by capturing keystrokes, clipboard data, screenshots, and stored credentials. First detected in **2020**, it quickly gained popularity among cybercriminals due to its stealthy nature and ability to evade detection through advanced obfuscation techniques.

Typically distributed via phishing emails containing malicious attachments —such as Microsoft Word or PDF files with embedded macros—Snake Keylogger infiltrates a system once the victim opens the infected document. Once active, it silently records user activity and exfiltrates the stolen data via email, FTP, or messaging platforms like Telegram.

Snake Keylogger gained widespread attention in **2021**, when it became available as **Malware-as-a-Service (MaaS)** on underground hacker forums. This meant that even low-skilled cybercriminals could purchase and deploy the malware easily, leading to an increase in **targeted attacks on businesses, financial institutions, and individual users**. Security researchers have observed that the malware frequently evolves with **obfuscation techniques**, making it harder to detect by traditional antivirus solutions.

Key Features & Capabilities:

- Captures **keystrokes, clipboard data, screenshots, and credentials**.
- Sends stolen data via **SMTP (email), FTP, or Telegram bots**.
- Uses **obfuscation and anti-analysis techniques** to evade detection.

1.2. Working

1. Infection & Initial Access

- The victim receives a **phishing email** with a malicious **Word, Excel, or PDF document** attached.
- **The document contains a hidden VBA macro that executes malware upon opening.**
- The user is tricked into clicking **"Enable Macros" or "Enable Content"**, allowing the malicious script to run.

2. Malware Execution & Installation

- The macro **downloads and executes Snake Keylogger** from a remote command-and-control (C2) server.
- The malware installs itself into **Windows directories** (e.g., AppData or Temp) to persist across reboots.
- It may modify **Windows Registry keys** to launch automatically on startup.

3. Data Collection & Keylogging

- Snake Keylogger begins monitoring **keystrokes**, capturing everything the user types, including:
    - Login credentials (usernames & passwords).
    - Banking and payment details.
    - Messages, emails, and private data.
- It can also steal:
    - **Clipboard data** (copied text, passwords).
    - **Stored browser credentials**.

- **Screenshots of active windows**.

4. Exfiltration of Stolen Data

- The collected data is sent to the attacker via:
  - **SMTP (email)** – sends logs to an attacker-controlled email.
  - **FTP** – uploads data to a remote server.
  - **Telegram bots** – uses encrypted communication for stealth.

5. Persistence & Evasion

- Snake Keylogger employs **anti-analysis techniques** to evade detection, such as:
  - **Code obfuscation** to bypass antivirus detection.
  - **Disabling security tools** like Windows Defender.
  - **Checking for sandbox environments** to avoid detection by researchers.

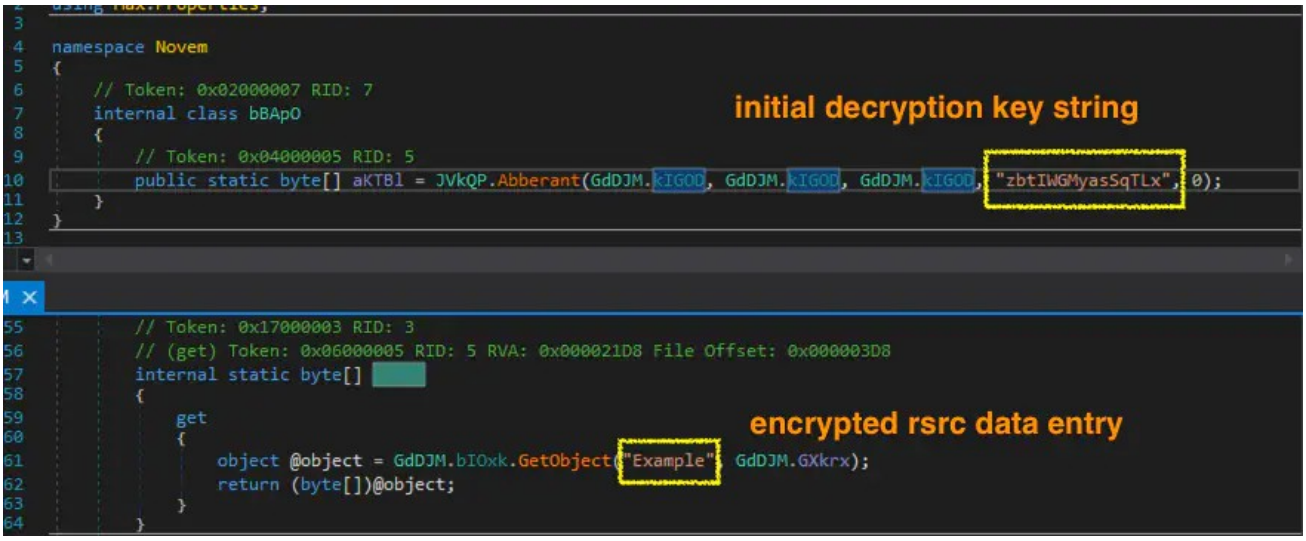
How Snake Keylogger Uses Macros[2]

A **macro** is a script or set of automated instructions embedded within a document, typically used in applications like **Microsoft Word, Excel, and PowerPoint**. Macros are written in **Visual Basic for Applications (VBA)** and are designed to automate repetitive tasks, such as formatting, calculations, and data entry.

Snake Keylogger is often delivered via phishing emails that contain **malicious Microsoft Office documents**. These documents prompt the victim to enable macros, usually by displaying a fake security warning like: *"This document is protected. Enable content to view the full text."*

If the victim enables macros, the embedded VBA script **executes malicious code** that:

- **Downloads and installs Snake Keylogger** onto the system.
- **Runs in the background**, logging keystrokes, stealing credentials, and capturing screenshots.
- **Exfiltrates stolen data** via email, FTP, or Telegram[2].



1.3 Services Compromised

The primary categories of services targeted by Snake Keylogger include the following:

1. Email Services

Snake Keylogger poses a significant threat to widely used email platforms, including **Gmail, Outlook, Yahoo Mail, and ProtonMail**. By intercepting login credentials, the malware facilitates unauthorized access, enabling threat actors to manipulate email communications, disseminate phishing campaigns, and extract sensitive correspondences.

2. Banking and Financial Services

The malware actively targets online banking systems, such as **PayPal, Bank of America, and Wells Fargo**, with the intent of capturing authentication credentials. This form of credential theft exposes victims to financial fraud, unauthorized fund

transfers, and illicit transactions.

3. Social Media Platforms

Popular social networking services, including **Facebook, Instagram, Twitter, and LinkedIn**, are frequently compromised through Snake Keylogger's credential-stealing mechanisms. The exploitation of these accounts can result in identity theft, social engineering attacks, and the propagation of further malware.

4. Cloud Storage and File-Sharing Services

Platforms such as **Google Drive, Dropbox, OneDrive, and Mega** serve as lucrative targets due to their role in storing sensitive documents and proprietary data. Unauthorized access to these repositories heightens the risk of intellectual property theft, data leaks, and corporate espionage.

5. E-Commerce and Online Payment Platforms

Snake Keylogger also targets e-commerce platforms and online payment services. By intercepting credentials used for platforms such as Amazon, eBay, and payment gateways, attackers can conduct fraudulent transactions, steal user account information, and exploit customer trust. This type of attack can lead to significant financial loss for both consumers and businesses, while also damaging the reputation of the e-commerce platform involved.

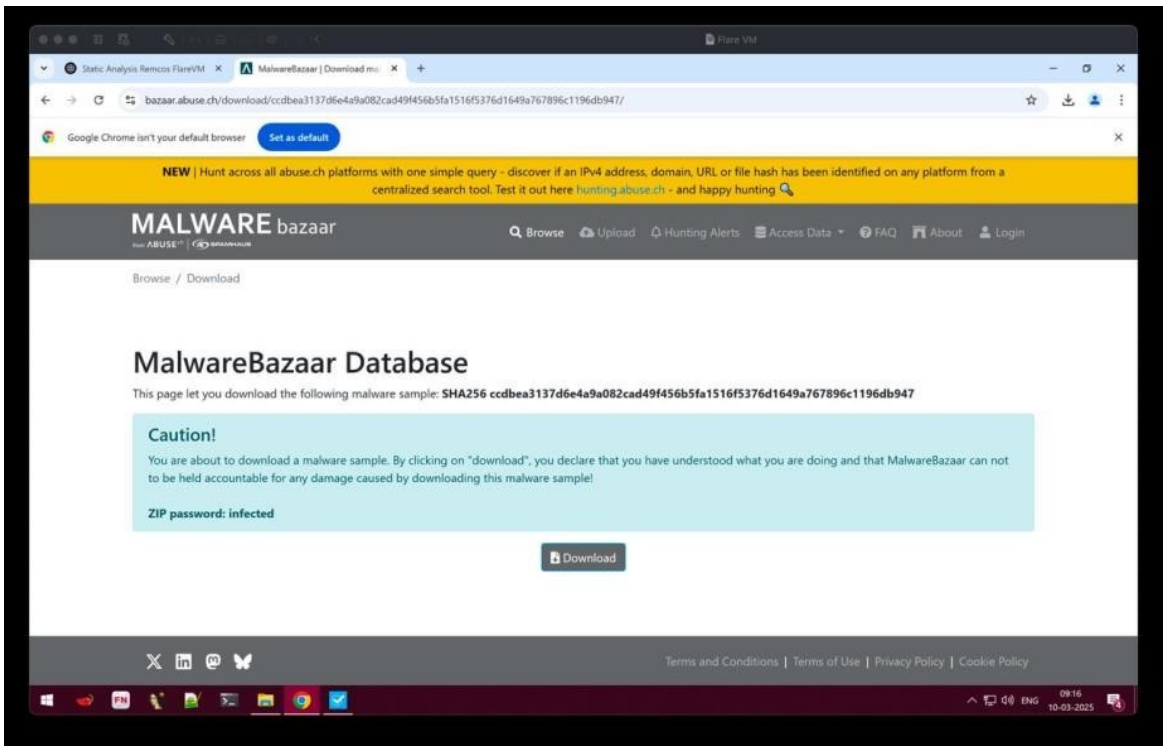
---

[2] https://www.splunk.com/en_us/blog/security/under-the-hood-of-snakekeylogger-analyzing-its-loader-and-its-tactics-techniques-and-procedures.html

[3] : https://www.fortinet.com/blog/threat-research/deep-analysis-of-snake-keylogger

2.1 Static Analysis

Static Analysis is the Analysis of malware done without actual execution inside a controlled environment such as a Virtual Machine. Here in order to do a static analysis on the SnakeKeyLogger malware we download a malware sample from www.malwarebazaar.com .

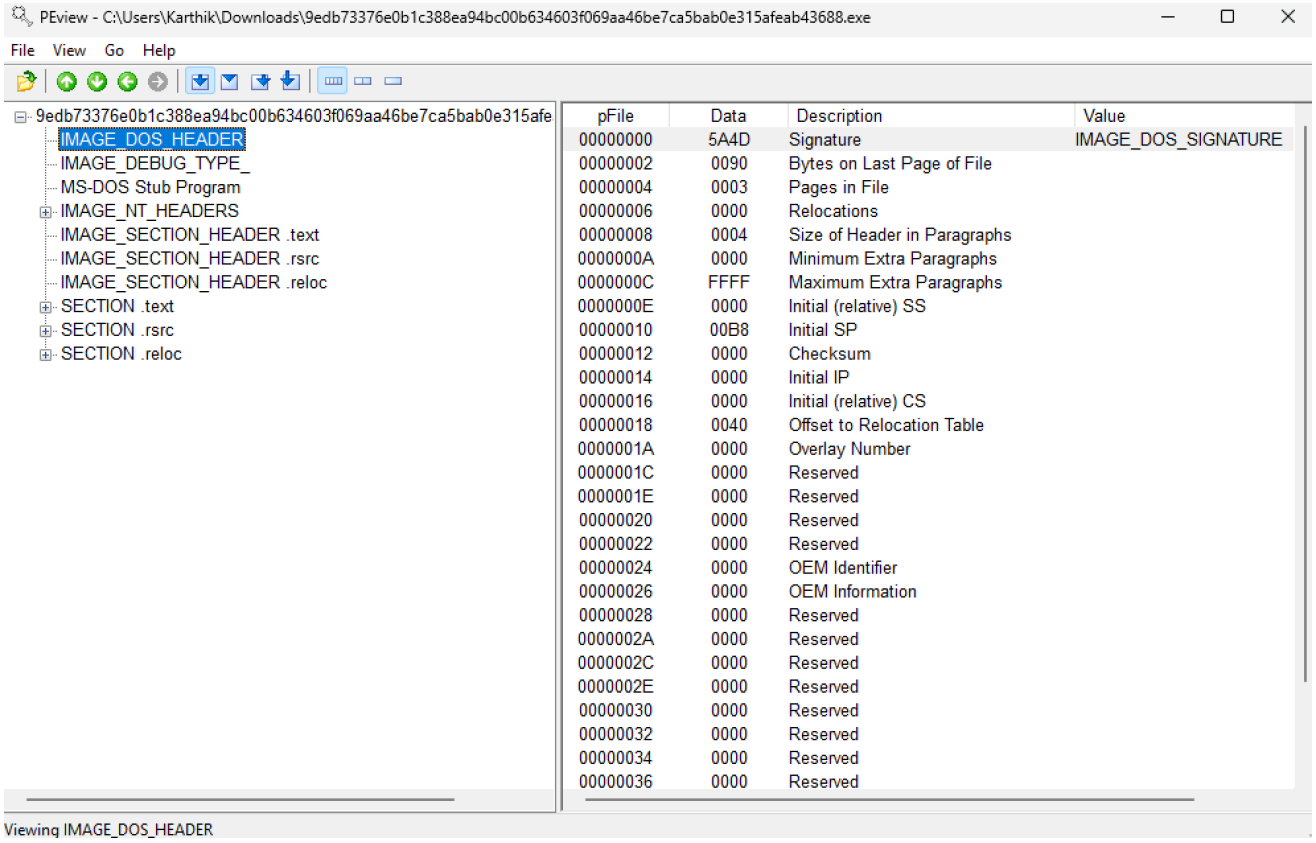


Here we employ a variety of tools using our flareVM to conduct a static analysis on our downloaded malware sample from the Internet.

1. PEView

Indicates that the file is a Windows executable with sections containing encrypted or packed code. Magic number is 5A4D which is .exe.ed

The PE file contains a valid DOS header, starting with the MZ signature (IMAGE_DOS_SIGNATURE).



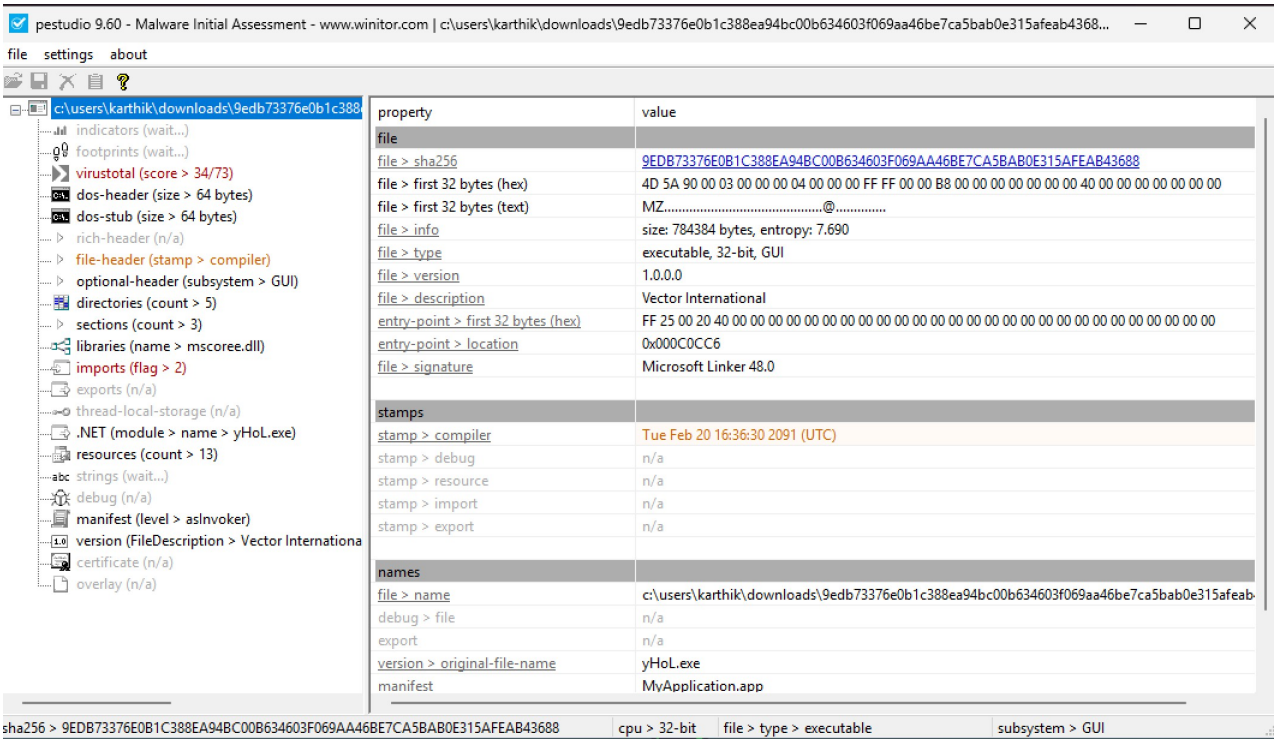
2. PEStudio

High VirusTotal Detection Rate (34/73)

Multiple AV engines flagged it as **malicious**, including detections like Win32:MalwareX-gen [Trj], Malicious, and Kryptik.ANKV.The presence of W32.AIDetectMalware.CS suggests AI-based detection of malware behaviors.

Suspicious Compilation Timestamp (2091)

Malware authors often manipulate timestamps to evade detection by security tools that rely on behavioral analysis.



High Entropy (7.690)

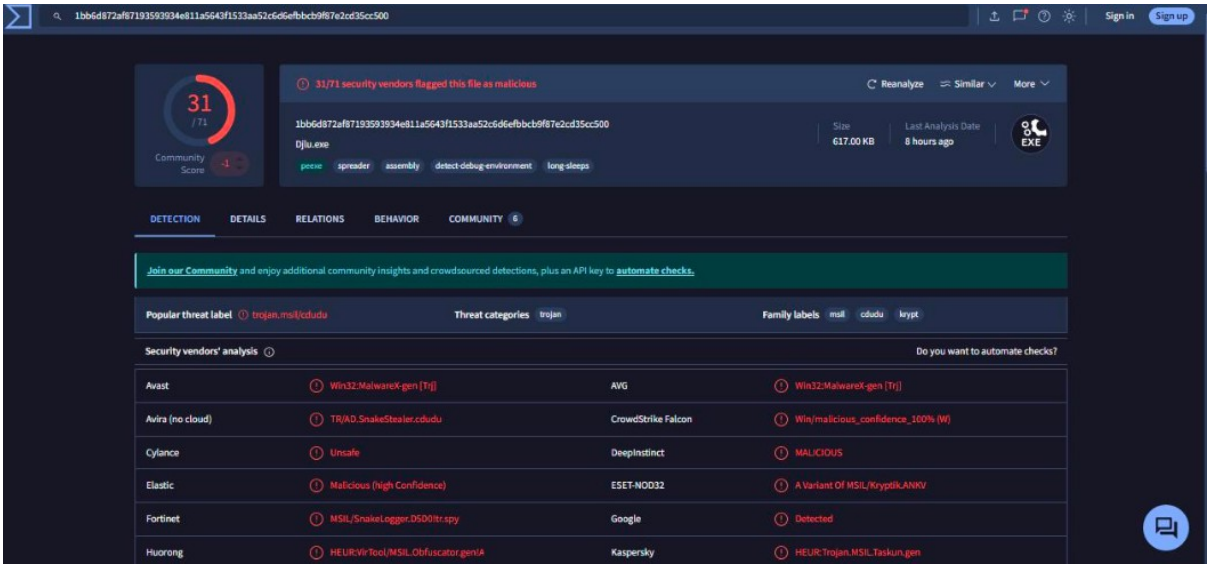
High entropy suggests encryption or packing, which is a common obfuscation technique used by keyloggers to hide their real behavior.

Presence of .NET Module (yHoL.exe)

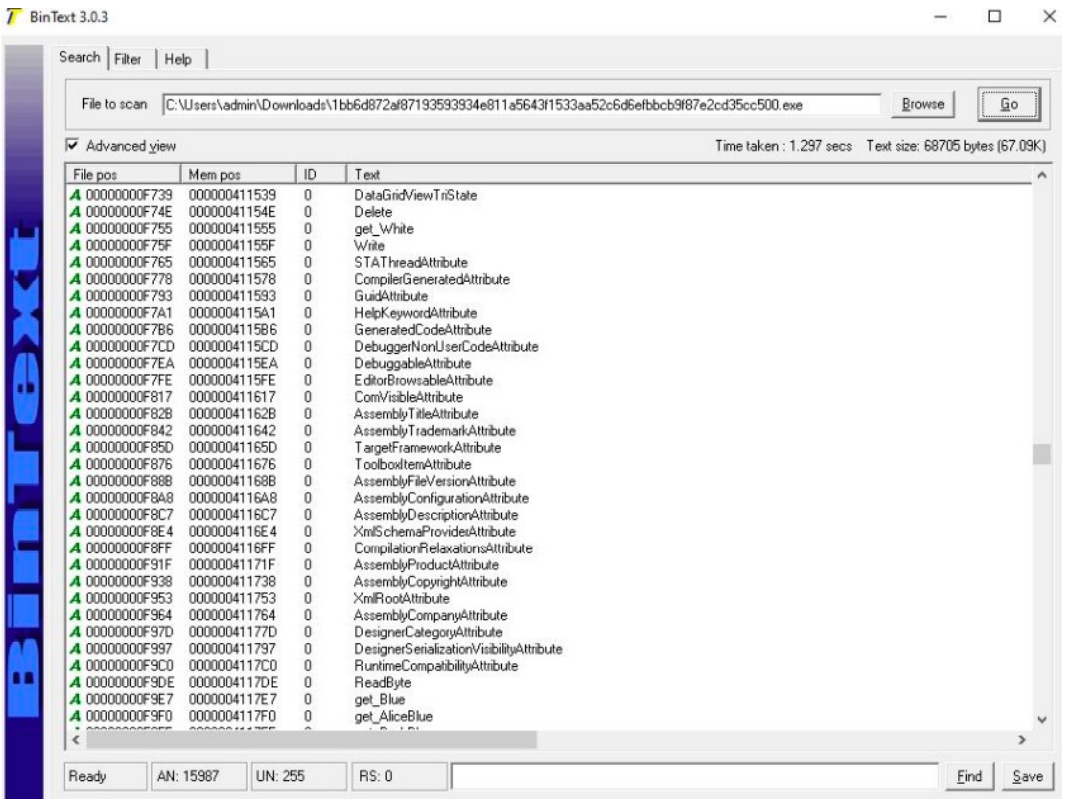
Snake Keylogger is known to be written in **.NET**, and the presence of a .NET module aligns with its structure.

3. VirusTotal

Uploading the malware sample to VirusTotal reveals multiple detections from antivirus engines, confirming its malicious nature.Detection Rate: 31/73 AV engines flagged it as malicious.

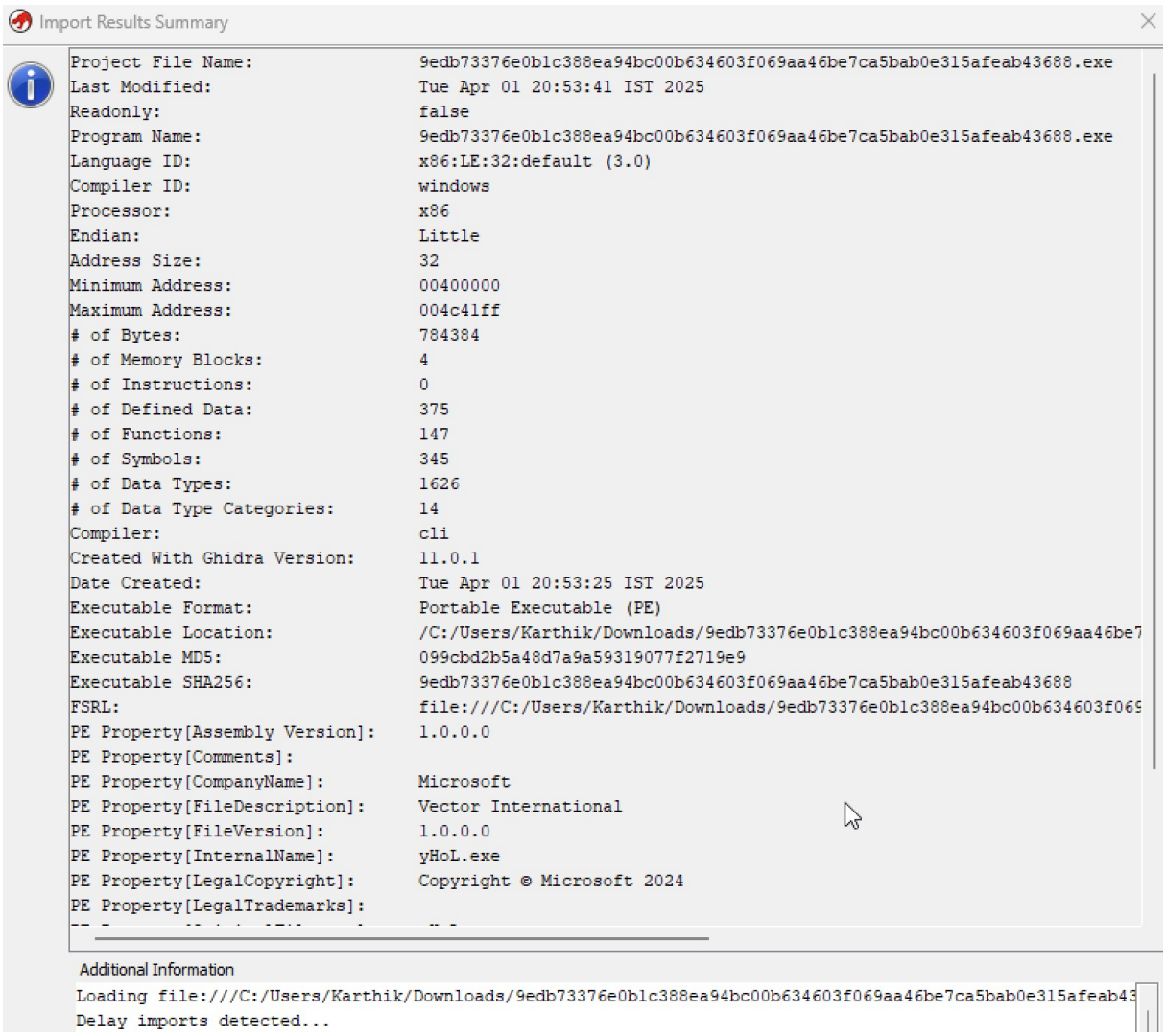


4. **BinText:** Extracts suspicious strings such as `keylog`, `sendtoC2`, and `stealpassword`, revealing its malicious intent.



5. GHIDRA

GHIDRA, an open-source reverse engineering tool by the NSA, is widely used for malware analysis. To load a sample, create a new project, import the file (**File > Import File**), and start the analysis. Once complete, open **CodeBrowser** to inspect disassembled code, functions, and imports. Use the decompiler to reconstruct logic, helping to understand the malware's behavior efficiently.



Entry Point & Main Function :

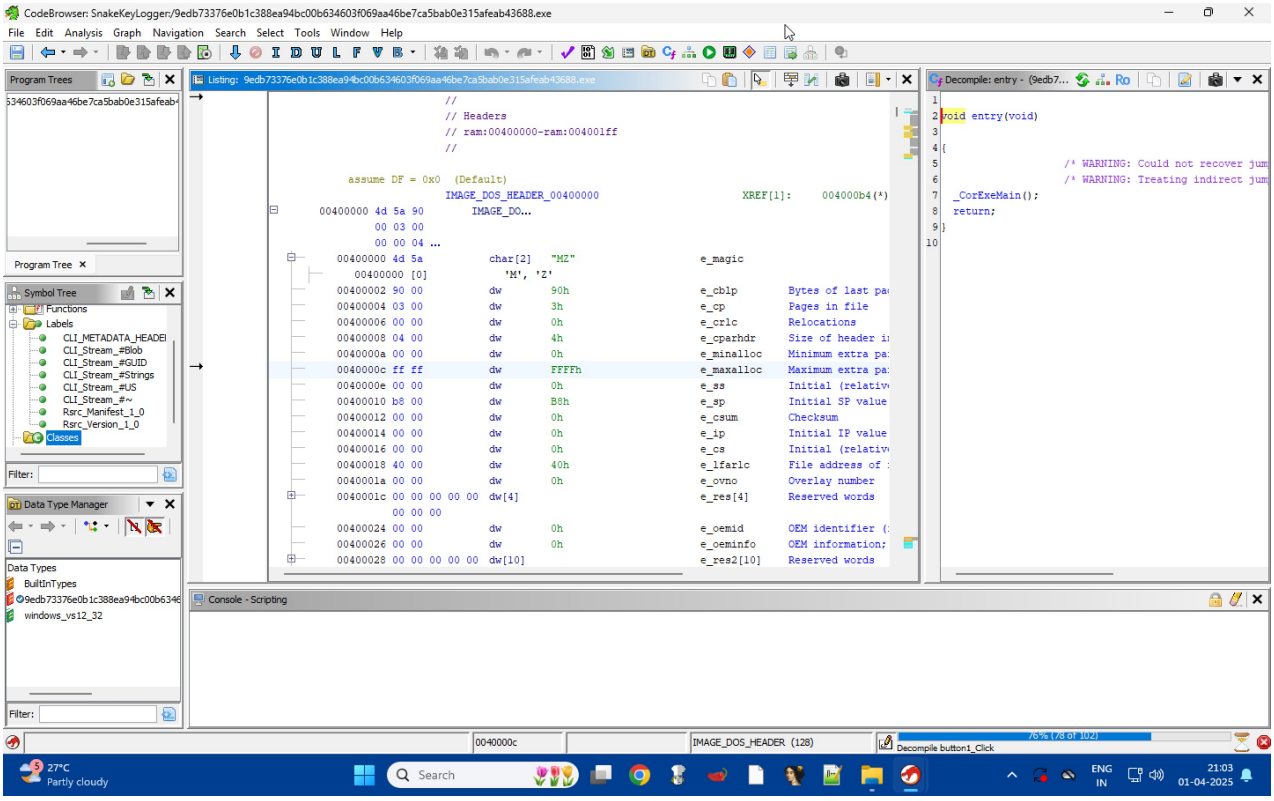
_CorExeMain() confirms it's a **.NET-based executable**.

The main logic is embedded in **CIL (Common Intermediate Language) bytecode**.

Possible use of **obfuscation or packing** to evade analysis.

Import Descriptor Table :

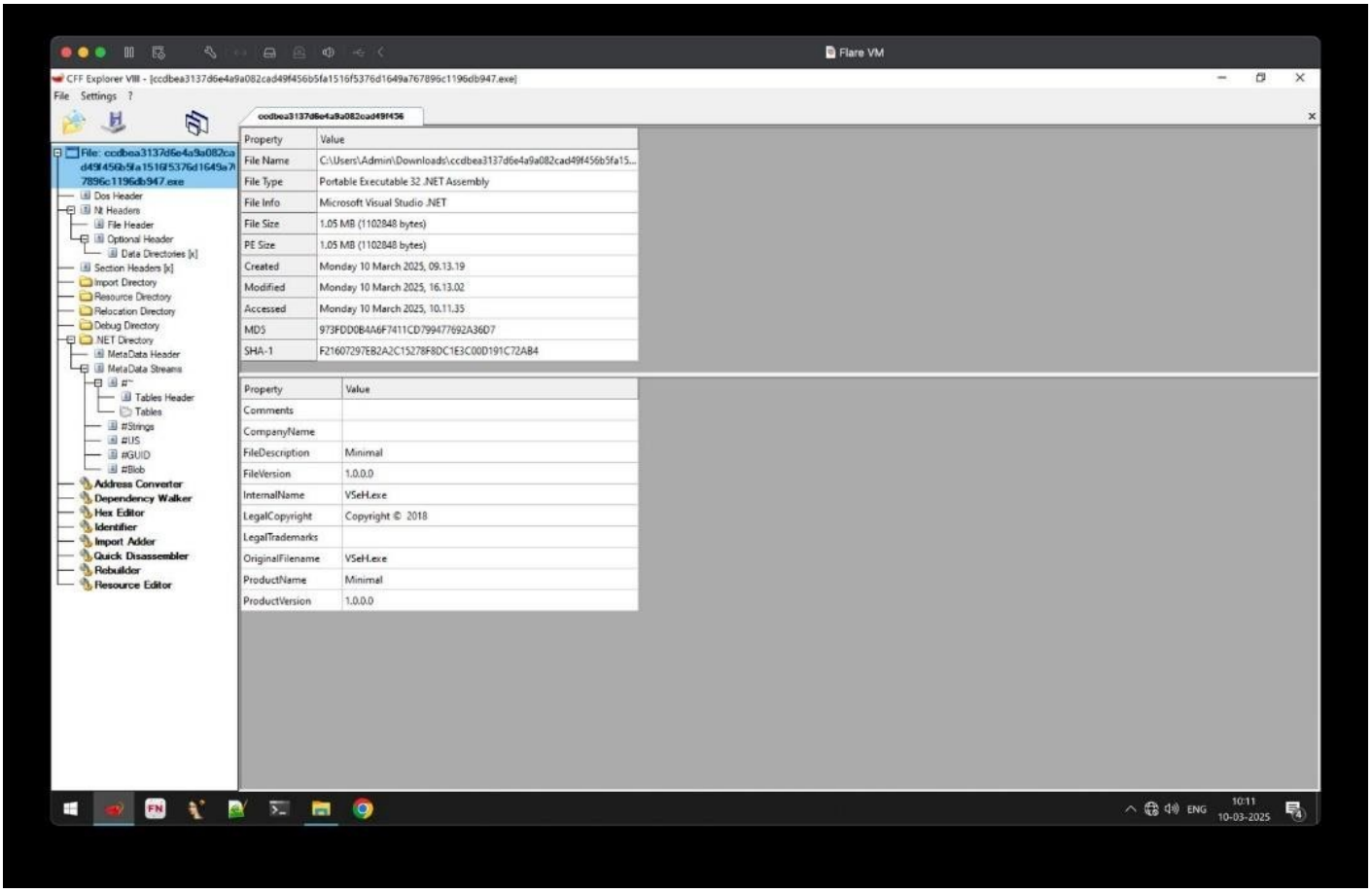
The **IMAGE_IMPORT_DESCRIPTOR** section contains references to **suspicious DLLs**.
Some fields hold **unusual values (0x2000)**, indicating **import table manipulation**.
Malware may be using **runtime API resolution** to conceal critical functions.

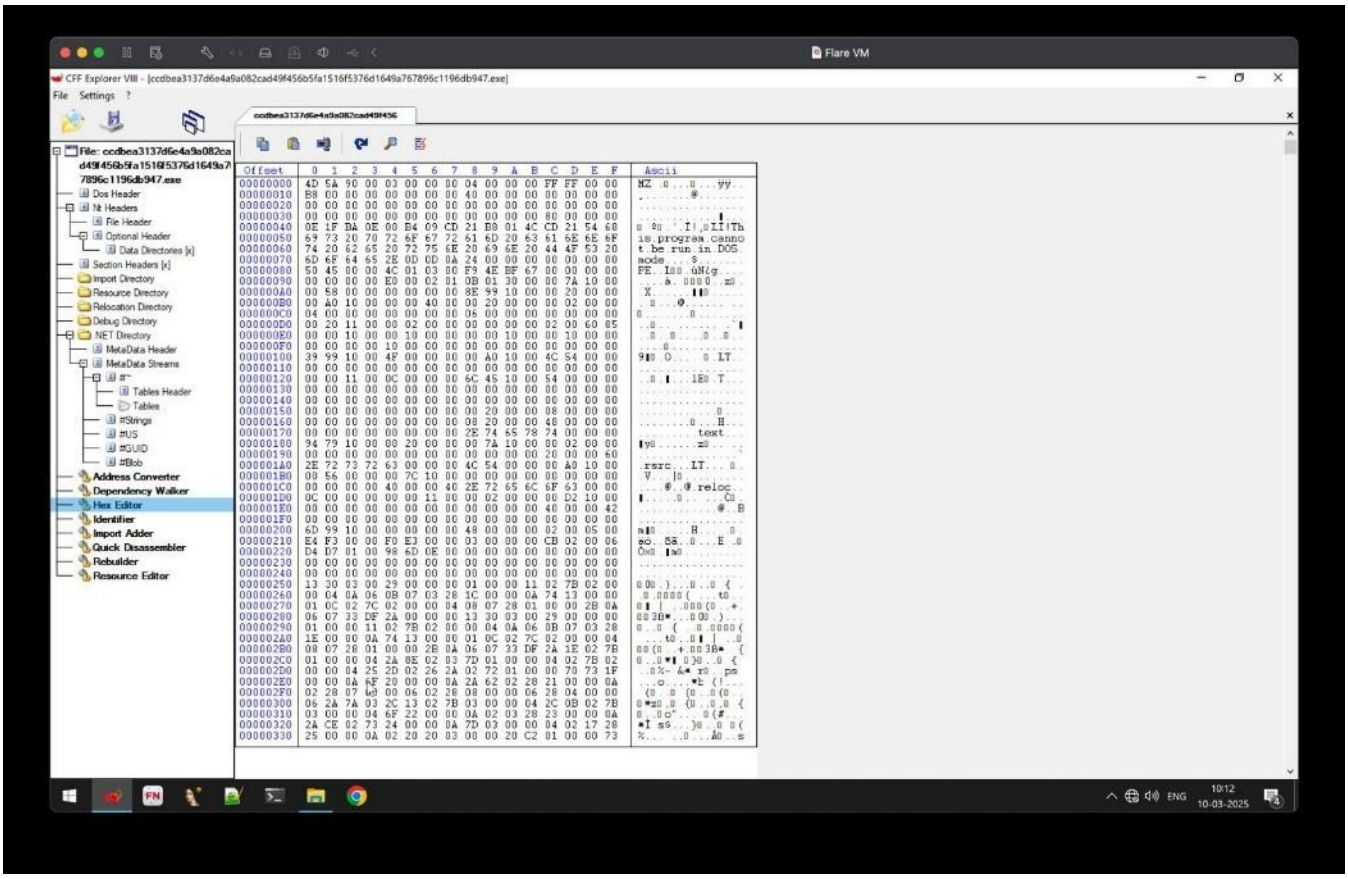


6. CFF Explorer

CFF Explorer is a powerful tool for analyzing Portable Executable (PE) files. To examine a malware sample, open CFF Explorer and load the executable. Navigate through headers, imports, and sections to inspect its structure.

Check the **Optional Header** for subsystem info and security flags, and review **Import Directory** to identify external dependencies. This helps in detecting suspicious API calls and potential evasion techniques.





This image displays an analysis of a **.NET Portable Executable** using CFF Explorer. It reveals key attributes like **file version, internal name, and hash values (MD5 & SHA-1)**. The presence of a **MetaData Header** suggests it's a .NET-based file

.

Shows imports like `GetAsyncKeyState`, `WriteFile`, and `IsDebuggerPresent`, which confirm its keylogging and anti- analysis functions.

The .text section (likely containing executable code) and .rdata section (likely containing strings and imports) are visible in the dump..reloc suggests the presence of relocation data, indicating it might be dynamically loaded into memory.

Dynamic Analysis

Dynamic analysis is about running a program in a controlled setup to study its real behavior. This helps in detecting malware actions like process injection, registry changes, and network activity.

Using Hybrid Online Malware Sandboxing

VMRay

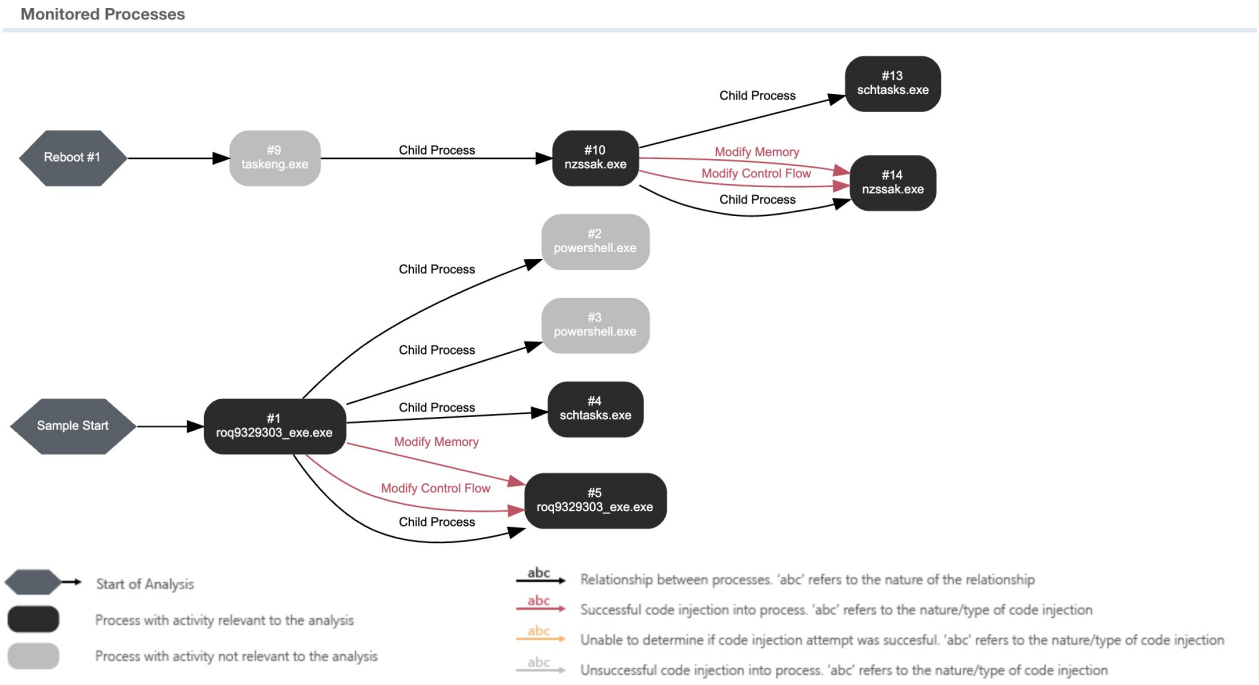
VMRay was used to analyze the behavior of a suspicious executable in a controlled sandbox environment. It helped trace the process tree, detect code injection activities, and identify keylogging capabilities by extracting critical artifacts like encryption keys, SMTP credentials, and C2 server URLs.

VMRay Threat Identifiers (22 rules, 72 matches)

Score	Category	Operation	Count	Classification
5/5	YARA	Malicious content matched by YARA rules	2	Spyware
5/5	Extracted Configuration	VIPKeylogger configuration was extracted	1	Spyware
4/5	YARA	Malicious content matched by YARA rules	2	Spyware
4/5	Reputation	Malicious file detected via reputation	1	-
4/5	Reputation	Malicious host or URL detected via reputation	3	-
4/5	Defense Evasion	Modifies Windows Defender configuration	2	-
3/5	Network Connection	Performs DNS request for known DDNS domain	1	-
2/5	Discovery	Searches for sensitive mail data	3	-
2/5	Data Collection	Reads sensitive mail data	2	-
2/5	Discovery	Searches for sensitive browser data	22	-

YARA Matches (6)

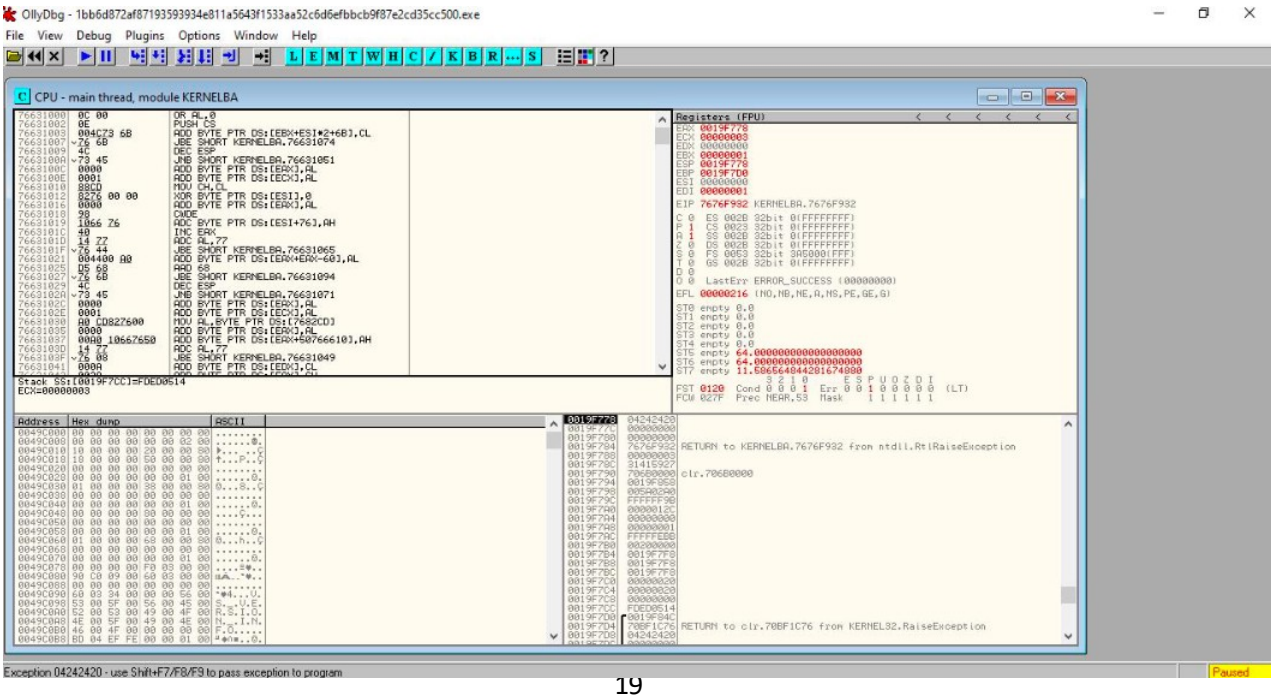
Ruleset Name	Rule Name	Rule Description	File Type	File Name	Classification	Score	Actions
Malware	VIPKeylogger	VIPKeylogger strings	Memory Dump	-	Spyware	5/5	•••
Malware	VIPKeylogger	VIPKeylogger strings	Memory Dump	-	Spyware	5/5	•••
Malware	VIPKeylogger	VIPKeylogger strings	Memory Dump	-	Spyware	5/5	•••
Generic	InfoStealer_Generic	Generic InfoStealer detection	Memory Dump	-	Spyware	4/5	•••
Generic	InfoStealer_Generic	Generic InfoStealer detection	Memory Dump	-	Spyware	4/5	•••
Generic	InfoStealer_Generic	Generic InfoStealer detection	Memory Dump	-	Spyware	4/5	•••

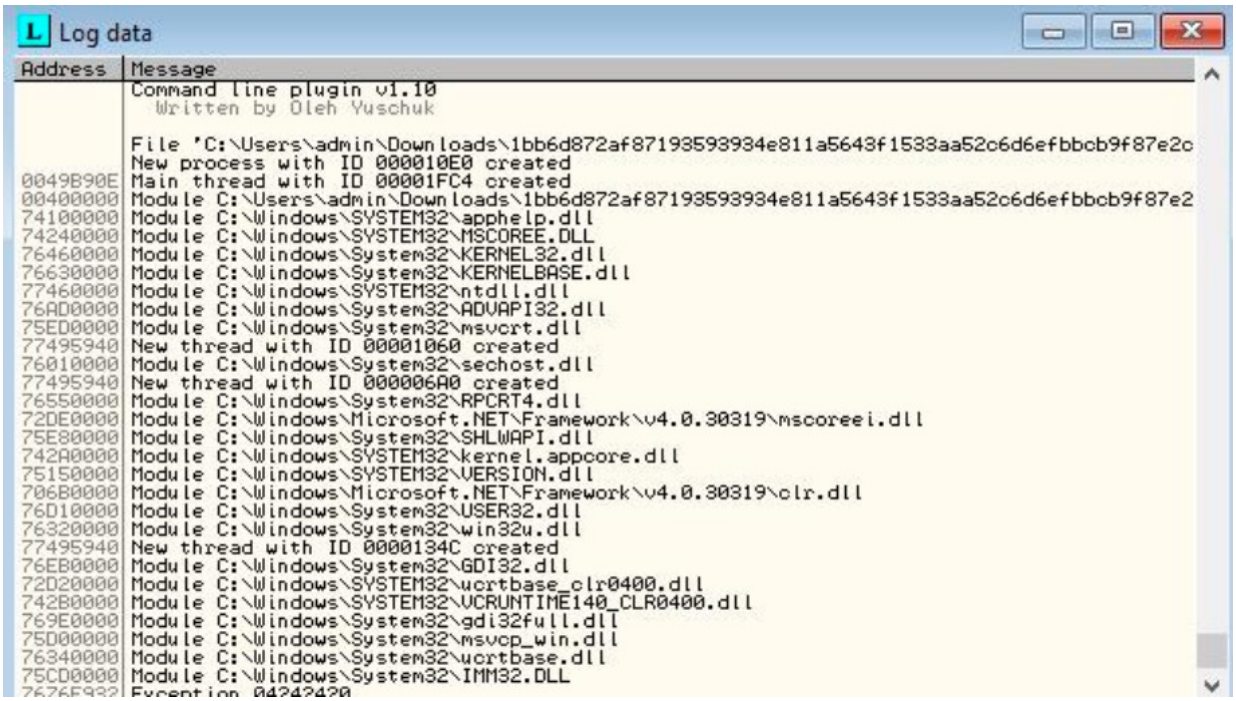


OLLYDBG

OllyDbg was utilized to inspect the dynamic behavior of the suspected malware. It revealed the loading of several standard DLLs, including `mscoree.dll` and `mscoreei.dll`, suggesting the application is built using .NET.

The debugger logs also highlighted thread creation and module loading activities within system directories like `C:\Windows\System32`, which are indicative of potential malware behavior—especially linked to threats like Snake Keylogger.





Joe Sandbox

Joe Sandbox was employed to perform a static and dynamic analysis of the suspicious executable `rROQ9329303.exe`. It provided crucial metadata such as the file hashes (MD5, SHA1, SHA256) and tagged the sample as associated with **Snake Keylogger**, indicating its malicious nature. The use of tags like `exe`, `SnakeKeylogger`, and `user-Porcupine` further supports its classification as a known threat, enabling threat intelligence correlation across systems.

Mitre Att&ck Matrix

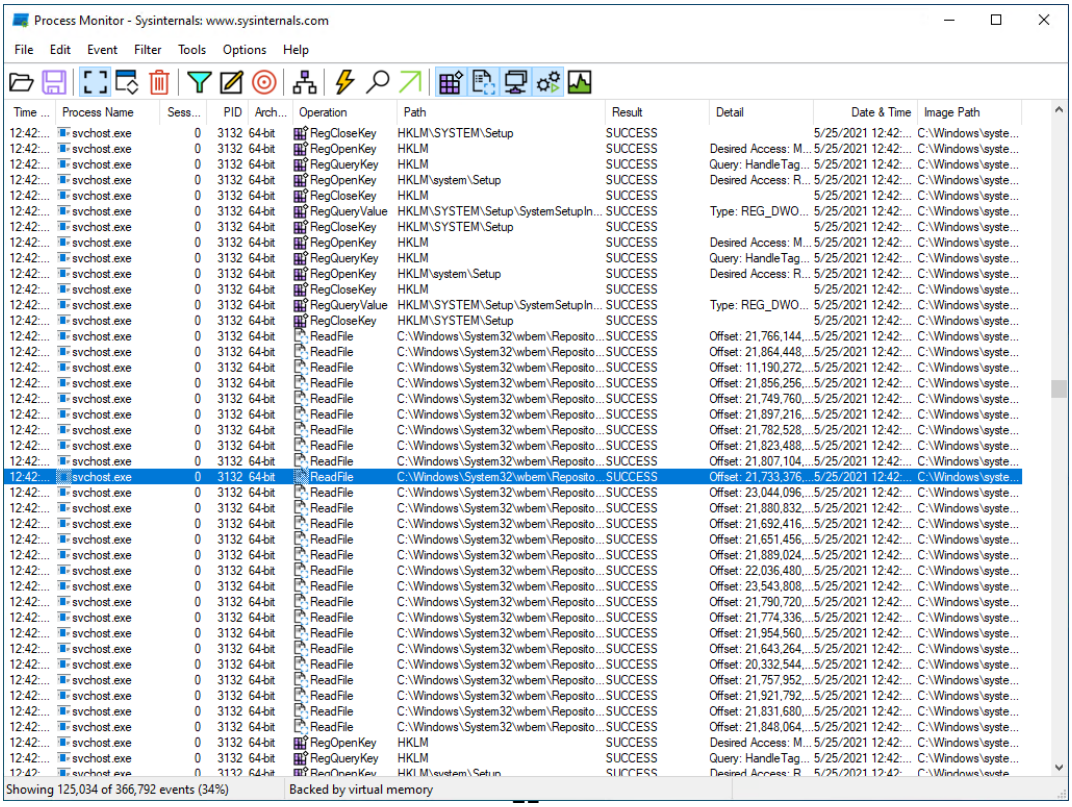
Reconnaiss...	Resource Development	Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Gather Victim Identity Information	Acquire Infrastructure	Valid Accounts	1 Scheduled Task/Job	1 DLL Side-Loading	1 DLL Side-Loading	1 1 Disable or Modify Tools	1 OS Credential Dumping	1 File and Directory Discovery	Remote Services	1 1 Archive Collected Data	1 Web Service	Exfiltration Over Other Network Medium	Abuse Accessibility Features
Credentials	Domains	Default Accounts	Scheduled Task/Job	1 Scheduled Task/Job	1 1 1 Process Injection	1 Deobfuscate/Decode Files or Information	LSASS Memory	1 3 System Information Discovery	Remote Desktop Protocol	1 Data from Local System	3 Ingress Tool Transfer	Exfiltration Over Bluetooth	Network Denial of Service
Email Addresses	DNS Server	Domain Accounts	At	Logon Script (Windows)	1 Scheduled Task/Job	3 Obfuscated Files or Information	Security Account Manager	1 1 Security Software Discovery	SMB/Windows Admin Shares	1 Email Collection	1 1 Encrypted Channel	Automated Exfiltration	Data Encrypted for Impact
Employee Names	Virtual Private Server	Local Accounts	Cron	Login Hook	Login Hook	1 2 Software Packing	NTDS	1 Process Discovery	Distributed Component Object Model	Input Capture	1 Non-Standard Port	Traffic Duplication	Data Destruction
Gather Victim Network Information	Server	Cloud Accounts	Launchd	Network Logon Script	Network Logon Script	1 Timestomp	LSA Secrets	3 1 Virtualization/Sandbox Evasion	SSH	Keylogging	3 Non-Application Layer Protocol	Scheduled Transfer	Data Encrypted for Impact
Domain Properties	Botnet	Replication Through Removable Media	Scheduled Task	RC Scripts	RC Scripts	1 DLL Side-Loading	Cached Domain Credentials	1 Application Window Discovery	VNC	GUI Input Capture	2 4 Application Layer Protocol	Data Transfer Size Limits	Service Stop
DNS	Web Services	External Remote Services	Systemd Timers	Startup Items	Startup Items	1 Masquerading	DCSync	1 System Network Configuration Discovery	Windows Remote Management	Web Portal Capture	Commonly Used Port	Exfiltration Over C2 Channel	Inhibit System Recovery

```
{
  "Exfil Mode": "SMTP",
  "Email ID": "587",
  "Password": "matias@repsider.com.ar",
  "Host": "Lanus090909$",
  "Port": "587"
}
```


The configuration data of **VIP Keylogger** reveals that it uses **SMTP** for exfiltrating stolen information. Credentials such as `matias@repsider.com.ar` with the password `Lanus090909$` are hardcoded, along with the SMTP port `587`. This shows the malware is configured to automatically send logged keystrokes to an attacker-controlled email server, making it a clear example of credential theft and data exfiltration via email.

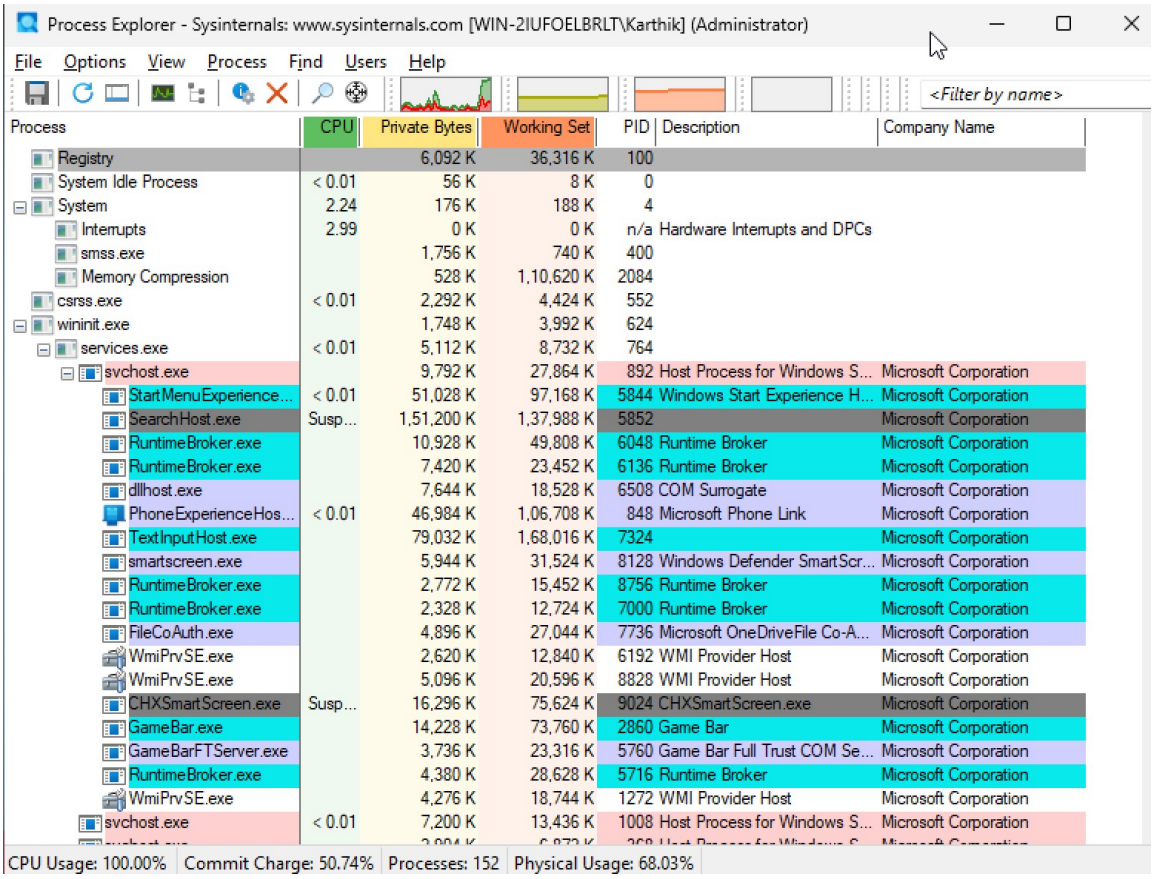
Process Monitor

Process Monitor was utilized to observe real-time system-level behavior of the suspicious executable `roq9329303_exe.exe`. It revealed the creation of multiple child processes such as `schtasks.exe` and `powershell.exe`, often associated with persistence and script execution. The tool also captured frequent file and registry operations in critical directories like `System32` and `AppData`, along with network activity over port 587, reinforcing its role in detecting keylogger behavior and data exfiltration attempts via SMTP.



Process Explorer

Process Explorer displayed the creation of multiple child processes under the main executable but only for a second before the process was hid, and this makes the mitigation tough.



Mitigation and Removal Steps:

- Open Process Explorer or Task Manager.
- Identify and terminate all instances of the keylogger executable.
- Use Regshot to confirm that no permanent registry changes were made.
- Manually delete the executable file from the desktop or its original location.
- If pop-ups or symptoms persist, revert the VM to a pre-execution snapshot.

Additional Solutions and Preventive Measures:

1. **Reboot in Safe Mode:** If pop-ups prevent user interaction, reboot the system in Safe Mode to disable non-essential services and processes. From there, navigate to the file's location and delete it manually.

2. **Use an Antivirus Tool:** Many antivirus programs detect this file as prankware. Running a scan with a reputable antivirus tool can assist in automatic detection and removal.

3. **Create a System Restore Point:** Before executing suspicious files, always create a restore point. In cases where the malware alters settings or causes UI denial, system restore can revert the machine to a prior, clean state.

4. **Use AppLocker or Software Restriction Policies:** Prevent execution of unauthorized applications in enterprise or lab environments by enforcing these group policies.

5. **Educate on Safe Handling:** Ensure malware samples are only run in isolated virtual environments with snapshots and restricted access. Disable file sharing between host and guest VMs to avoid accidental host infection.

6. **Monitor Network Traffic:** Although this sample does not exhibit network behavior, using tools like Wireshark can ensure there is no covert data exfiltration in other malware cases.

Conclusion:

The keylogger in question is a malicious software designed to record keystrokes and exfiltrate sensitive data. While its primary function is covert data theft, it also demonstrates key malware behaviors such as code injection, process spawning, and file manipulation. The keylogger might not cause immediate visible damage, but its potential for data exfiltration and compromising user privacy is significant. This analysis highlights the importance of using static tools like **PEiD** and **OllyDbg** to analyze executable files and dynamic tools like **ProcMon** and **Process Explorer** to observe real-time behaviors. The sample also showcases how seemingly simple malware can exhibit complex tactics, techniques, and procedures (TTPs) often associated with advanced threats.

This project serves as a practical case study in identifying, analyzing, and documenting

malware in a controlled lab environment, further emphasizing the importance of proactive cybersecurity measures.