WannaCRY
Analysis Report

SALIM
ABDOUNE
MALWARE ANALYST

Contents

Executive Summary

SHA256 hash	24D004A104D4D54034DBCFCC2A4B19A11F39008A575AA614EA04703480B1022C

WannaCry is a notorious ransomware that gained global attention due to its widespread impact. It targets systems by encrypting files with specific extensions, rendering them inaccessible to users. The malware operates in two stages:
Stage One : The initial dropper (wannacry.exe) serves as the entry point. Upon execution, it attempts to connect to a hardcoded domain (http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com/). If the connection fails, the malware proceeds to execute its payload.
Stage two : The secondary component (tasksche.exe) performs the encryption process using a combination of RSA and AES encryption algorithms. This ensures that encrypted files can only be decrypted with a unique private key held by the attackers.

1 High-Level Technical Summary

RansomWare consists of the following components:

- **Wannacry.exe**: The main dropper executable.
- **tasksche.exe**: An executable that generates the encryptor, decryptor.exe, and the encryption keys (e.g., 00000000.pky).
- **Decrypto.exe**: An executable that demands payment and prompts the user to enter the key to decrypt files.
- **00000000.pky**: Contains the RSA key, which is itself encrypted.

2 Malware Composition

File Name	**SHA256 Hash**
Wannacry.exe	24D004A104D4D54034DBCFCC2A4B19A11F39008A575AA614EA04703480B1022C
tasksche.exe	ED01EBFBC9EB5BBEA545AF4D01BF5F1071661840480439C6E5BABE8E080E41AA
Decrypto.exe	none
00000000.pky	28F9847EC0B75AC2D7AB00DEB071471A5B85077D0B7E3828D64B7396AE55551E

Table 1: Malware Components

3 First Denotation

The Wannacry.exe Encrypt all the data and Create Decryptor.exe and create .txt file that is created to communicate with us to tell us how to pay

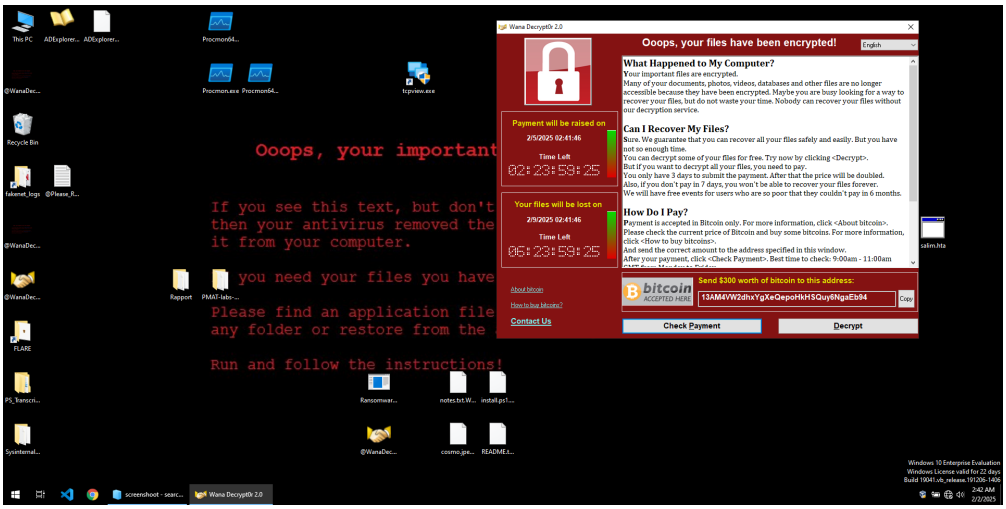


Figure 1: WannaCry Ransom Note Displayed to the User

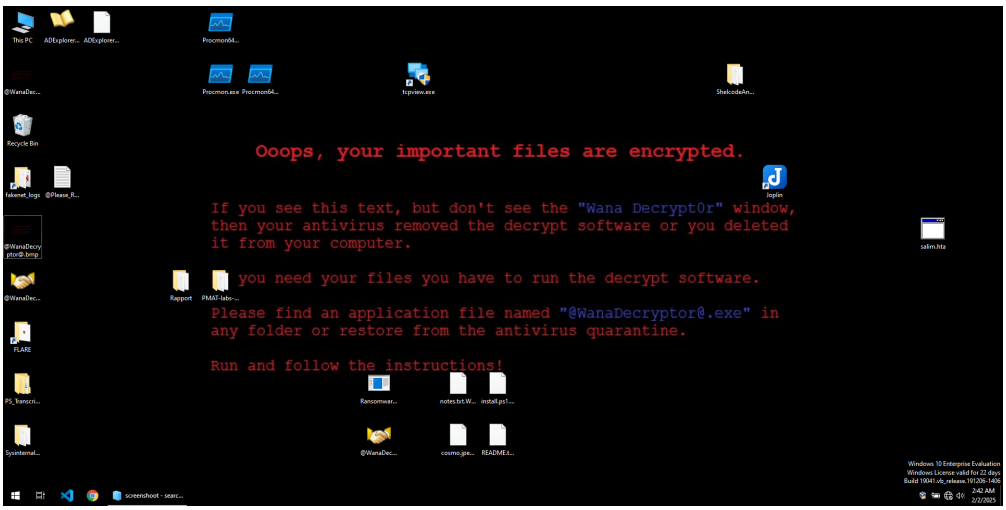


Figure 2: WannaCry File Encryption Process

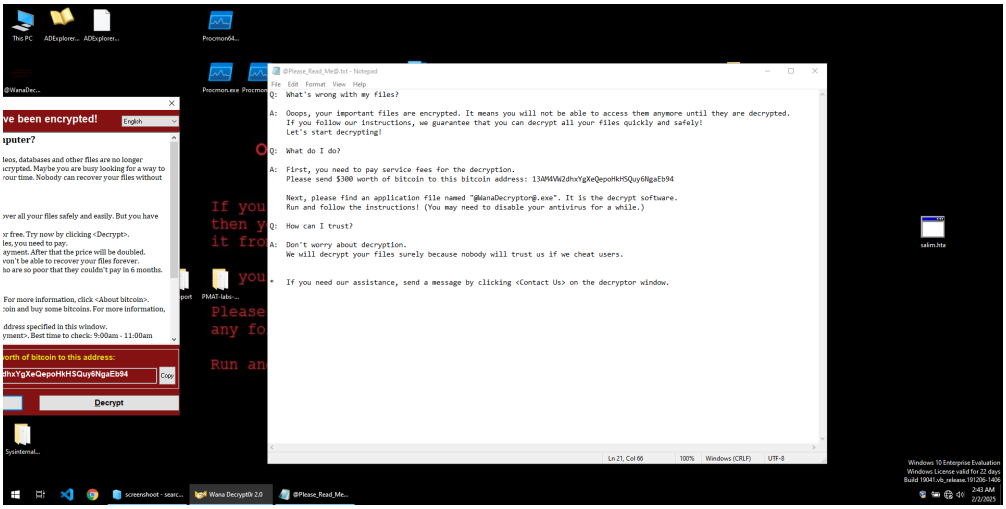


Figure 3: WannaCry Payment Instructions

4 Basic Static Analysis

- **SHA256**: 24D004A104D4D54034DBCFCC2A4B19A11F39008A575AA614EA04703480B1022C
- **MD5**: DB349B97C37D22F5EA1D1841E3C89EB4
- **Strings**: The malware contains strings related to file extensions it targets for encryption (e.g., .doc, .pdf, .jpg) and DLLs it uses (e.g., kernel32.dll, USER32.dll).
- **Domains**: http://www.iugerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
- **Mutex**: "GlobalMsWinZonesCacheCounterMutexA" is used to ensure single instance execution.
- **IAT Preview**: The Import Address Table (IAT) reveals critical APIs used by WannaCry for its functionality. Below is a breakdown of the important APIs and their purposes:
  - **ADVAPI32.dll**:
    - `StartServiceCtrlDispatcherA`: Used to register the malware as a service.
    - `RegisterServiceCtrlHandlerA`: Registers a function to handle service control requests.
    - `OpenSCManagerA`: Opens the Service Control Manager (SCM) to create or modify services.
    - `CreateServiceA`: Creates a new service for persistence.
    - `StartServiceA`: Starts the created service.
  - **KERNEL32.dll**:
    - `CreateFileA`: Opens or creates files for encryption.
    - `ReadFile`: Reads data from files.
    - `WriteFile`: Writes encrypted data back to files.
    - `GetFileSize`: Retrieves the size of files for encryption.
    - `CreateThread`: Creates threads for parallel execution of tasks.

    - `TerminateProcess`: Terminates processes to evade detection.
    - `GetTickCount`: Used for timing checks to detect debugging.
    - `QueryPerformanceCounter`: Another timing check for anti-debugging.
  - **USER32.dll**:
    - `MessageBoxA`: Displays ransom notes to the user.
  - **CRYPT32.dll**:
    `CryptGenRandom`: Generates random numbers for cryptographic operations. `CryptAcquireContext` Acquires a cryptographic context for encryption.
  - **WS2_32.dll**:
    `socket`: Creates network sockets for communication. `connect`: Connects to remote servers (e.g., for the kill switch domain). `send`: Sends data over the network. `recv`: Receives data from the network.

- **Entropy and Packing**: The malware exhibits high entropy, indicating that it is likely packed

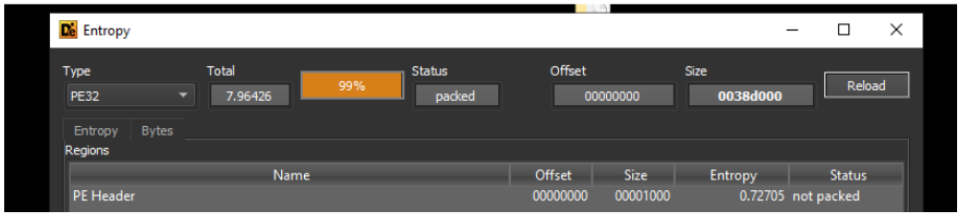


Figure 4: WannaCry CAPA Detection Results

or obfuscated to evade detection. Below are the details:

  - **Entropy**: The overall entropy of the file is **7.96**, which is very high (close to the maximum of 8). High entropy is often a sign of encryption or compression, commonly used in packed malware.
  - **Packed Sections**:
    - The `.text` section has an entropy of **7.96**, indicating it is packed.
    - The `.rsrc` section has an entropy of **7.27**, suggesting it contains encrypted or compressed resources.
  - **Packing Indicators**:
    - The PE header indicates the file is packed, with sections like `.rsrc` containing embedded resources.
    - The malware uses techniques like obfuscated stack strings and anti-debugging checks (e.g., `GetTickCount` and `QueryPerformanceCounter`) to hinder analysis.

- **CAPA Detection**: CAPA (Capability Detection) analysis reveals the following capabilities in the WannaCry malware:

  - **Anti-Analysis**:
    - `Debugger Detection`: Uses `GetTickCount` and `QueryPerformanceCounter` to detect debugging environments.
    - `Obfuscated Stack Strings`: Employs obfuscated strings to hinder static analysis.

- **Persistence**:
  - `Create or Modify System Process`: Creates a Windows service for persistence.
  - `Service Execution`: Executes malicious code as a service.
- **Execution**:
  - `Shared Modules`: Loads additional modules for functionality.
  - `Create Thread`: Creates threads for parallel execution of tasks.
- **File System**:
  - `Read File`: Reads files for encryption.
  - `Move File`: Moves files during the encryption process.
- **Network Communication**:
  - `HTTP Communication`: Connects to the kill switch domain (http://www.iugerfsodp9ifjaposdfjhgosur
  - `Socket Communication`: Creates TCP/UDP sockets for network communication.
- **Cryptography**:
  - `Generate Pseudo-random Sequence`: Uses `CryptGenRandom` for cryptographic operations.
  - `Encrypt Data`: Encrypts files using RSA and AES algorithms.

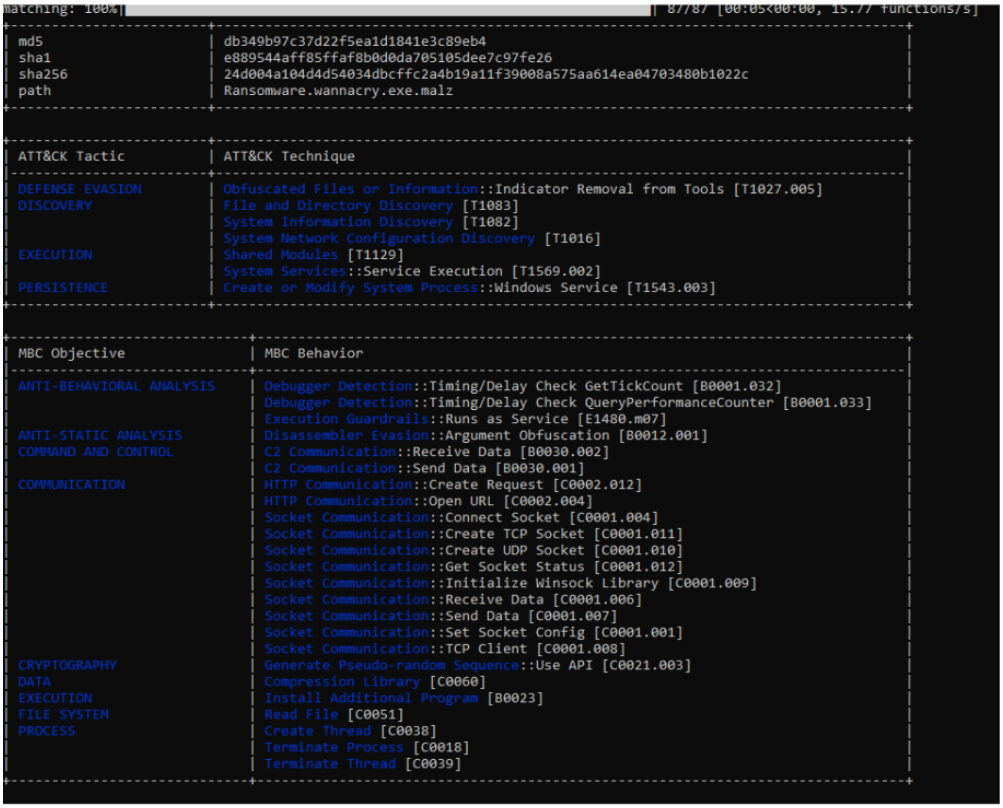


Figure 5: WannaCry CAPA Detection Results

5 Basic Dynamic Analysis

- **Network Activity**: The malware attempts to connect to the domain http://www.iugerfsodp9ifjaposdfjhgosu
  If the connection fails, it proceeds with the encryption process.

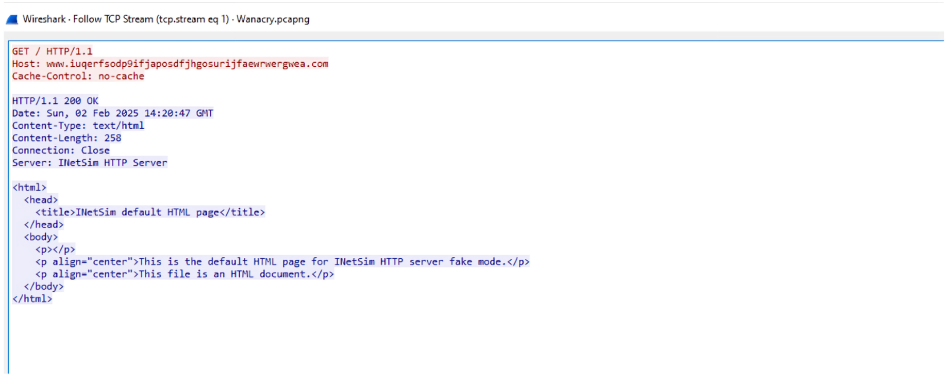


Figure 6: WannaCry Network Activity (DNS and HTTP Requests)

- **File System Changes**: The malware creates files like tasksche.exe and modifies registry keys to bypass proxy settings and treat UNC paths as part of the Local Intranet zone.

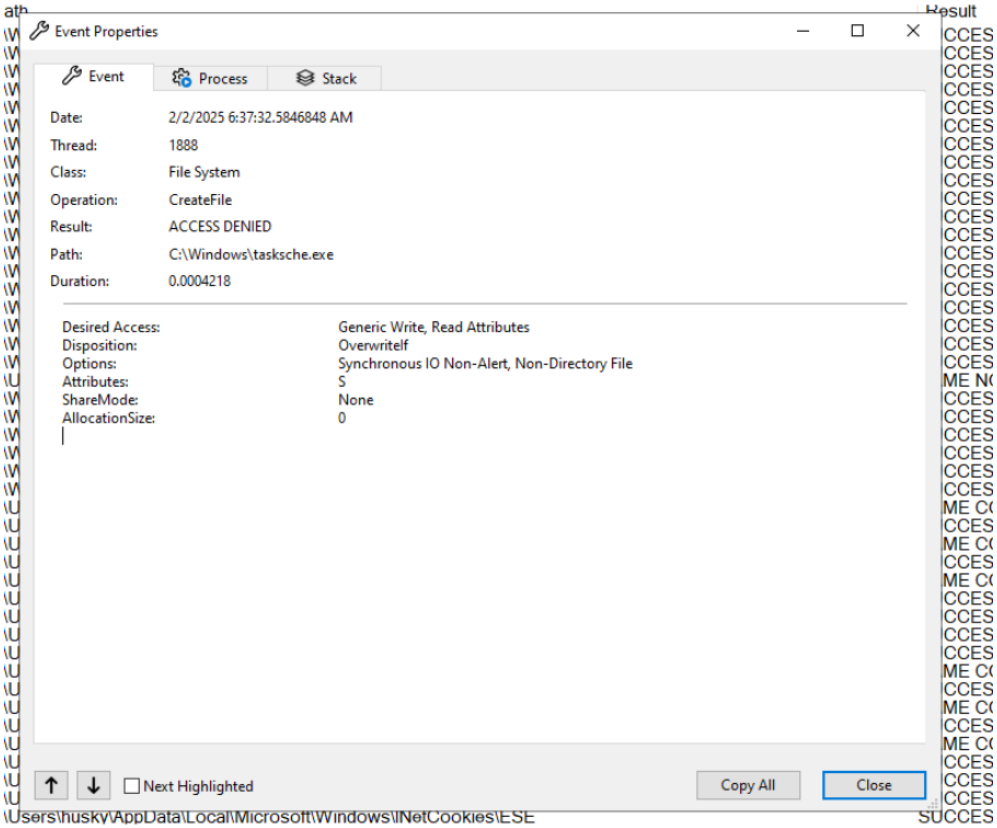
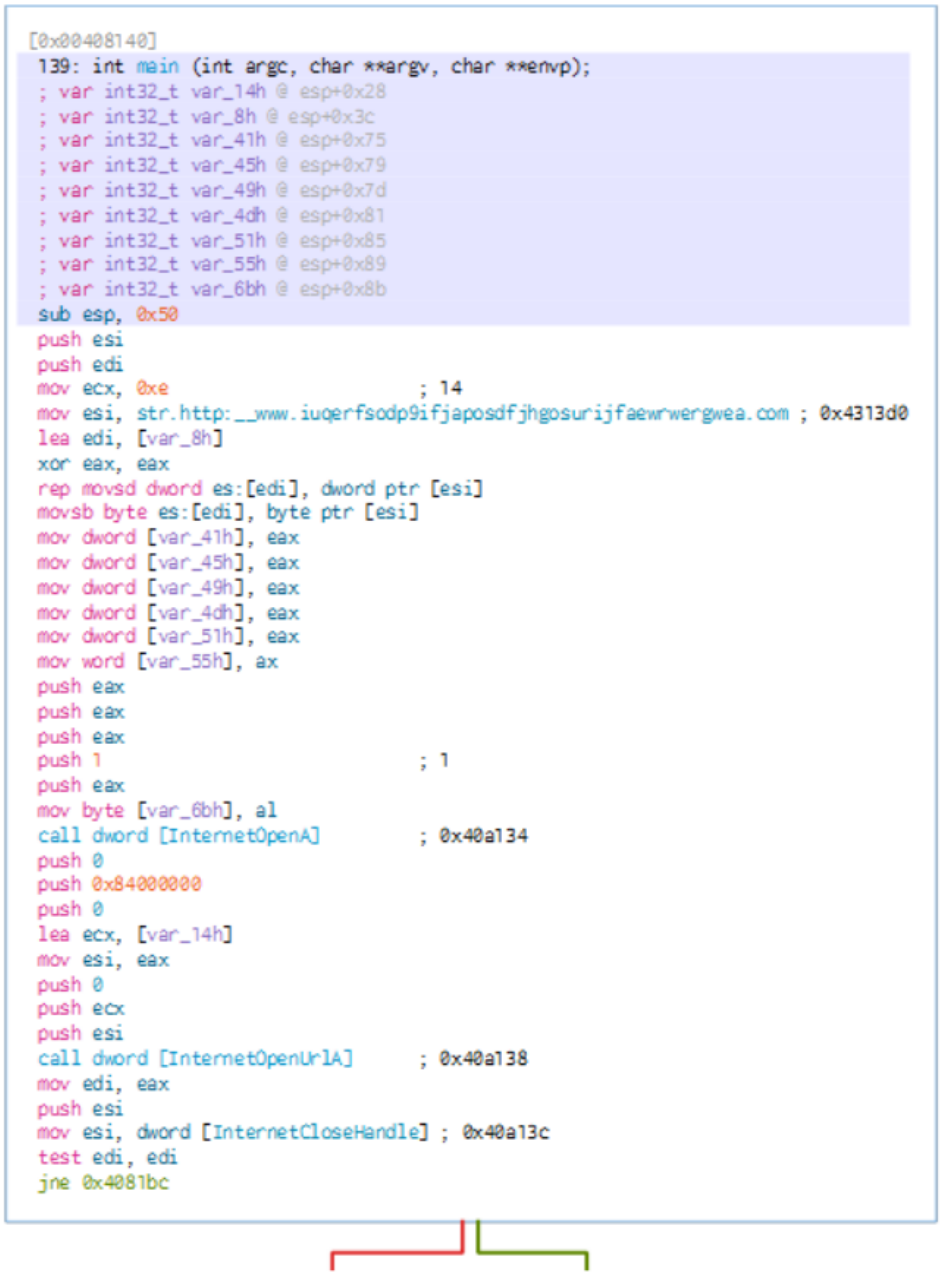


Figure 7: WannaCry Registry and File System Modifications

- **Commands Executed**:
  - `icacls . /grant Everyone:F /T /C /Q`: Grants full control to "Everyone" on the current directory and subdirectories.
  - `attrib +h .`: Hides files or directories by setting the hidden attribute.

6 Advanced Static Analysis

- **Disassembled Code Analysis**: The disassembled code of the ‘main‘ function reveals the malware's initialization and network communication logic. Below is a snippet of the disassembled code:



```
[0x00408140]
139: int main (int argc, char **argv, char **envp);
; var int32_t var_14h @ esp+0x28
; var int32_t var_8h @ esp+0x3c
; var int32_t var_41h @ esp+0x75
; var int32_t var_45h @ esp+0x79
; var int32_t var_49h @ esp+0x7d
; var int32_t var_4dh @ esp+0x81
; var int32_t var_51h @ esp+0x85
; var int32_t var_55h @ esp+0x89
; var int32_t var_6bh @ esp+0x8b
sub esp, 0x50
push esi
push edi
mov ecx, 0xe                          ; 14
mov esi, str.http:__www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com ; 0x4313d0
lea edi, [var_8h]
xor eax, eax
rep movsd dword es:[edi], dword ptr [esi]
movsb byte es:[edi], byte ptr [esi]
mov dword [var_41h], eax
mov dword [var_45h], eax
mov dword [var_49h], eax
mov dword [var_4dh], eax
mov dword [var_51h], eax
mov word [var_55h], ax
push eax
push eax
push eax
push 1                                ; 1
push eax
mov byte [var_6bh], al
call dword [InternetOpenA]            ; 0x40a134
push 0
push 0x84000000
push 0
lea ecx, [var_14h]
mov esi, eax
push 0
push ecx
push esi
call dword [InternetOpenUrlA]         ; 0x40a138
mov edi, eax
push esi
mov esi, dword [InternetCloseHandle] ; 0x40a13c
test edi, edi
jne 0x4081bc
```

Figure 8: Disassembled Code of WannaCry's `main` Function

- **Key Observations**:
  - The function starts by initializing variables and setting up the stack.

- It loads the hardcoded domain string: `http://www.iugerfsodp9ifjaposdfjhgpsurijfaewrwergwea.c`
- The malware uses the `InternetOpenA` and `InternetOpenUrlA` functions from the `WinINet` library to establish a connection to the domain.
- If the connection fails, the malware proceeds to execute its payload (encryption routine).
- The code includes anti-debugging checks, such as timing delays using `GetTickCount` and `QueryPerformanceCounter`.

- **Code Snippet Explanation**:
  - `sub esp, 0x50`: Allocates space on the stack for local variables.
  - `mov esi, str.http:...`: Loads the hardcoded domain string into the `esi` register.
  - `call dword [InternetOpenA]`: Calls the `InternetOpenA` function to initialize the use of the WinINet library.
  - `call dword [InternetOpenUrlA]`: Calls the `InternetOpenUrlA` function to open a connection to the specified URL.
  - `test edi, edi`: Checks if the connection was successful. If not, the malware proceeds to the encryption routine.

7 Advanced Dynamic Analysis

7.1 Analysis of WannaCry.exe

- **Process Injection**: The malware injects code into processes to evade detection.
- **File Creation**: The malware creates a file named `tasksche.exe`.

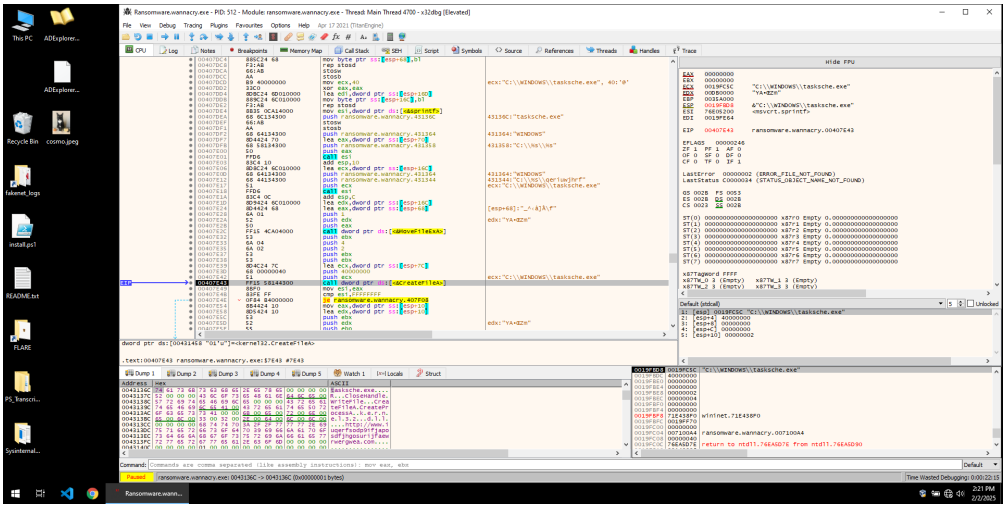


Figure 9: WannaCry Creating `tasksche.exe`

- **Service Creation**: It creates and modifies Windows services for persistence.

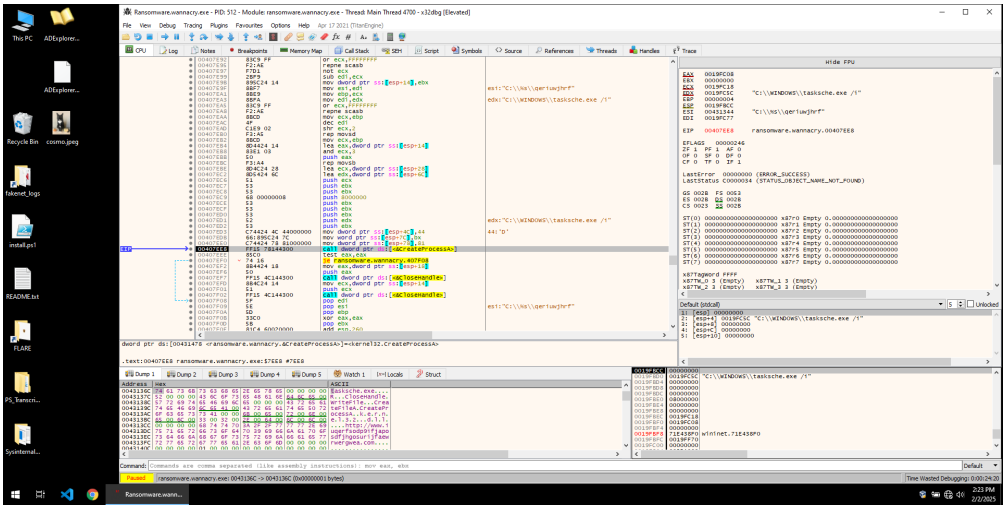


Figure 10: tasksche.exe Service Creation

- **Worm Capabilities**: The malware exhibits worm-like behavior, enabling it to spread across networks and infect other systems.

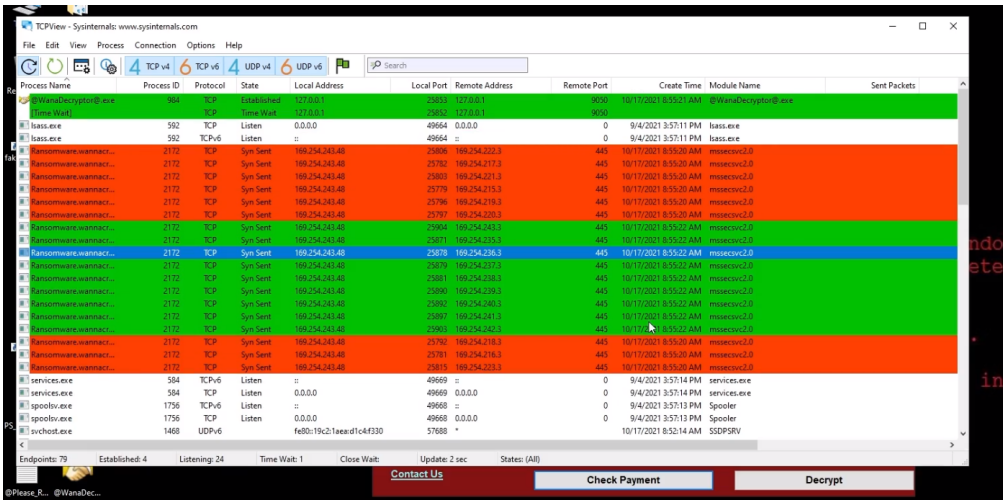


Figure 11: WannaCry.exe Worm Capabilities from tcpView

7.2 Analysis of tasksche.exe

Generates Two Other Executables: `tasksche.exe` generates two additional executables as part of its payload.

Generates `.wnry` Files: It creates multiple files with the `.wnry` extension, which are used in the subsequent stages of the payload.

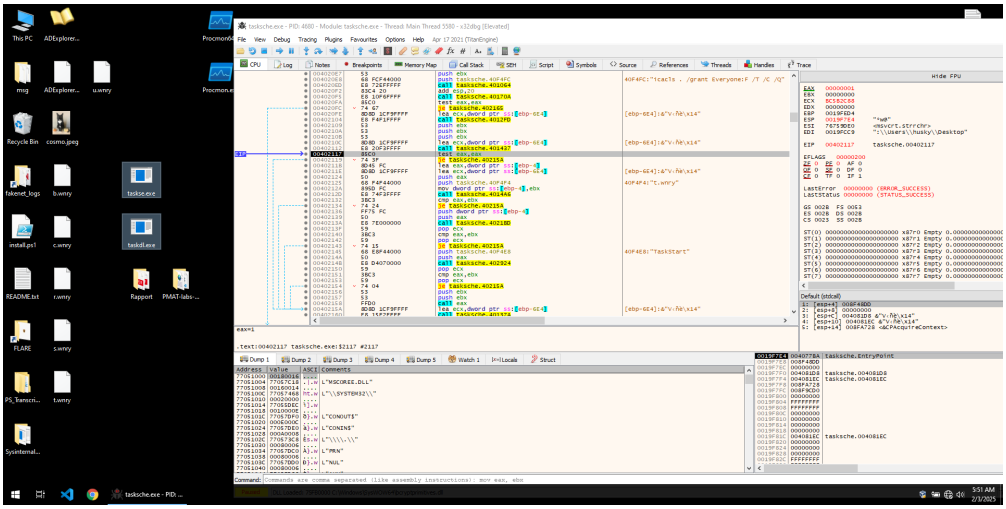


Figure 12: tasksche.exe Generating `.exe and .wnry` Files

Encrypts Data and Generates `Decryptor.exe`: It encrypts the victim's data, generates `Decryptor.exe`, and changes the desktop wallpaper to display the ransom note.

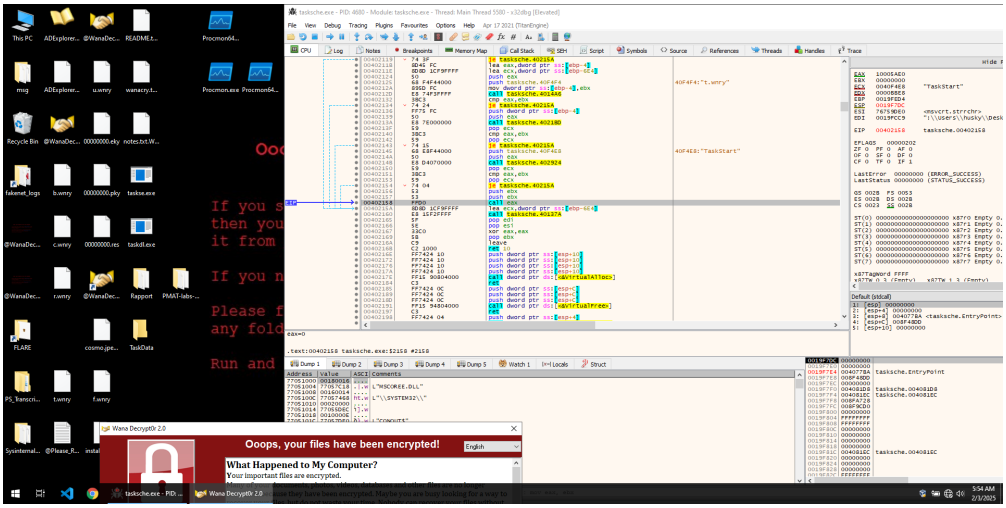


Figure 13: tasksche.exe Generating `Decryptor.exe` and Changing Wallpaper

Four Essential Functions of `tasksche.exe`:

- **Function 1**: Generates the cryptor and modifies registry values.

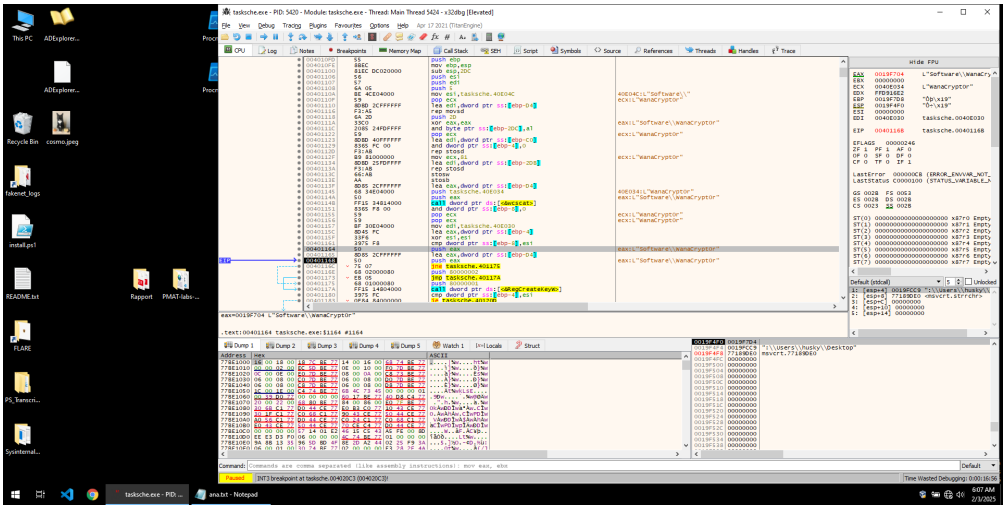


Figure 14: tasksche.exe Generating Cryptor and Modifying Registry

- **Function 2**: Creates the .exe files `/msg` folder and `.wnry` files, which are used in the payload's execution.

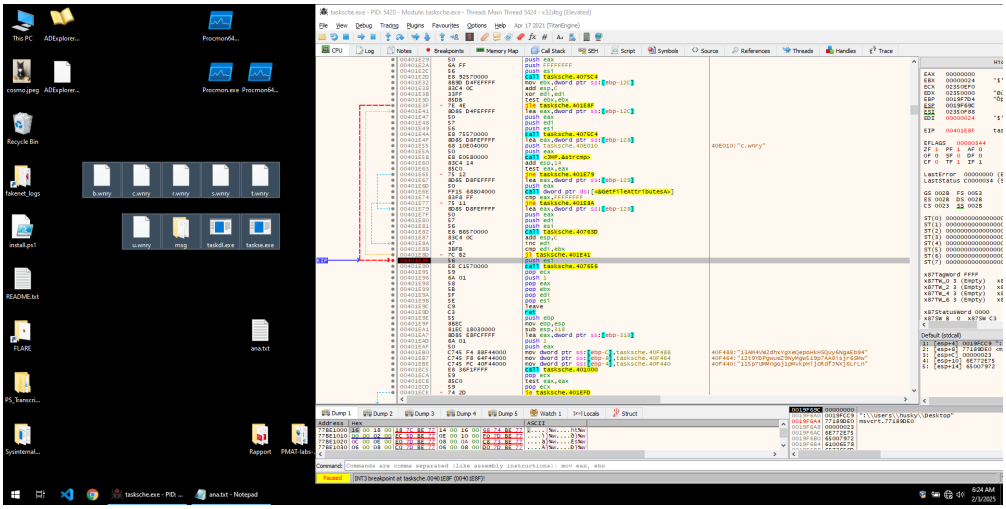


Figure 15: tasksche.exe Creating `/msg` Folder and `.wnry` Files

- **Function 3**: Contains Bitcoin wallet addresses and is responsible for handling ransom payments.

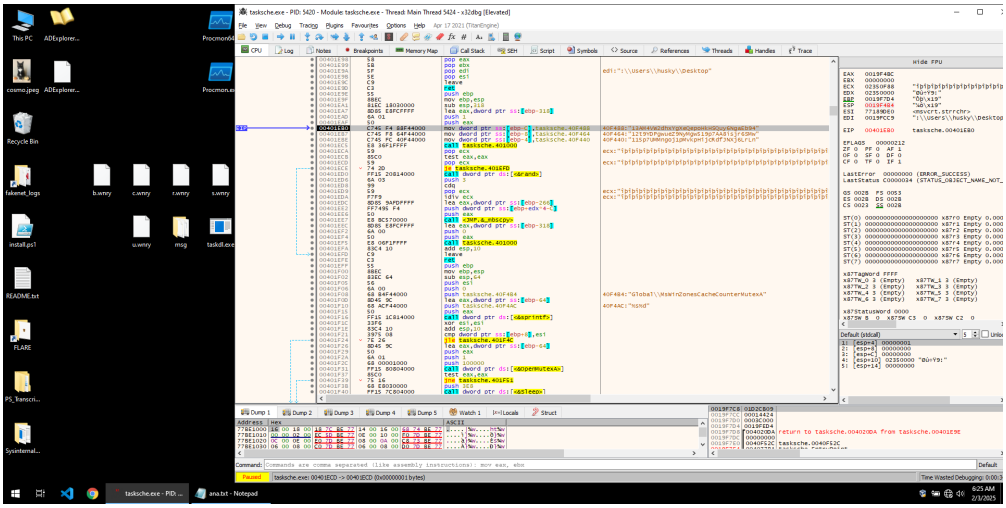


Figure 16: Bitcoin Wallet Addresses and Payment Handling

- **Function 4**: Generates the `000000.pky` file, which contains the RSA key encrypted with a specialized method, and creates the decryptor **And it's the responsible of the Encryption See the second Images**

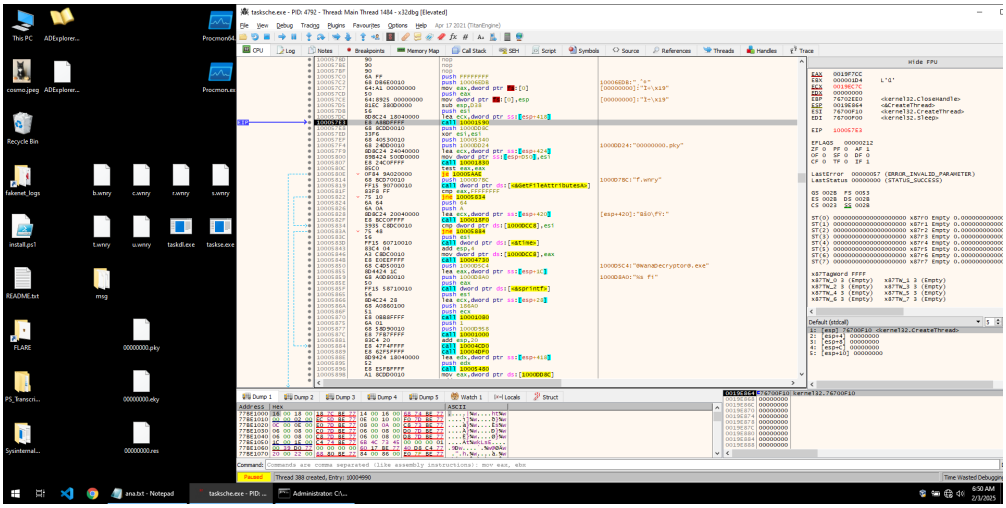


Figure 17: tasksche.exe Generating `000000.pky` and Decryptor

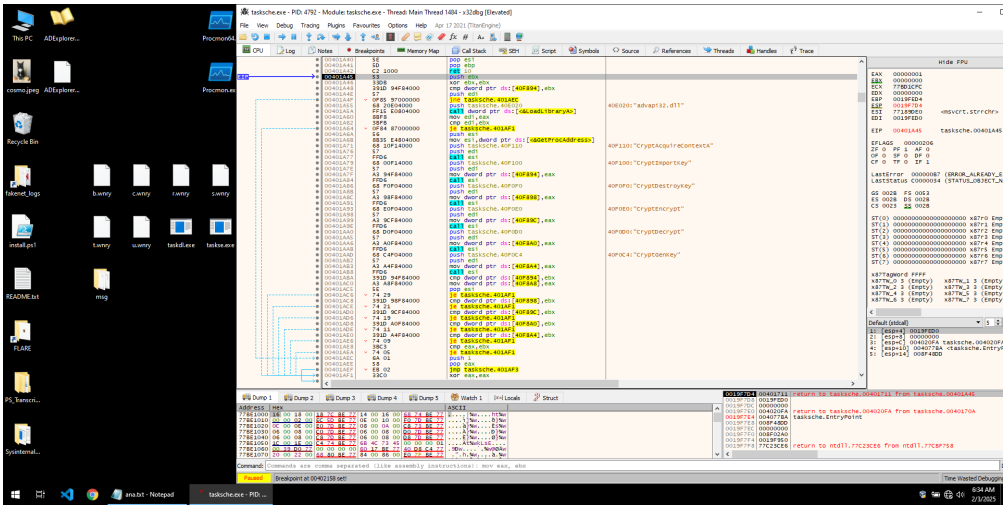


Figure 18: tasksche.exe do the encryption

8 Indicators of Compromise

8.1 Network Indicators

- **Domains**: http://www.iugerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
- **Fig 6:** WireShark Packet Capture of initial beacon check-in.

8.2 Host-based Indicators

- **Registry Changes**:
  - Bypass proxy settings: `RegSetValue` modifies proxy settings to bypass restrictions.
  - Intranet settings: `Intranet` and `UNCasIntranet` are set to 1, treating UNC paths as part of the Local Intranet zone.

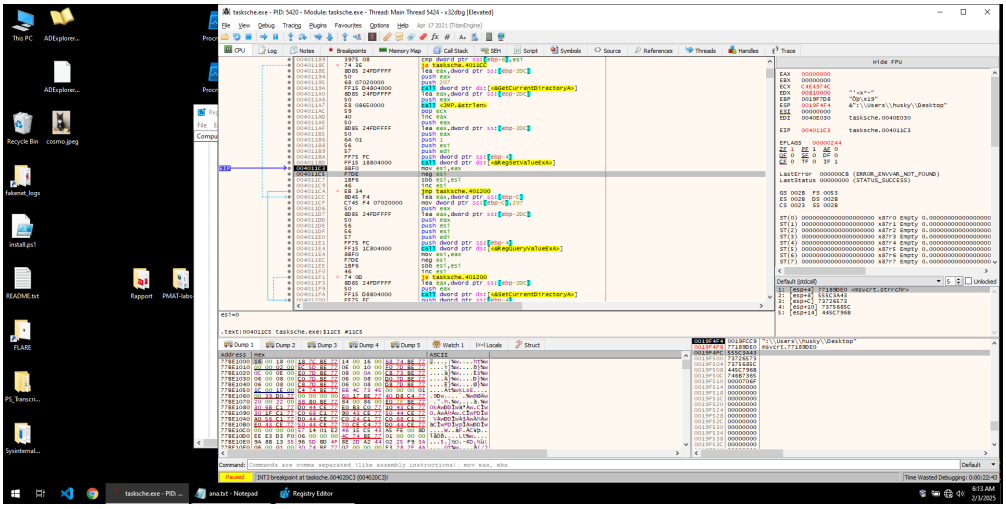


Figure 19: tasksche.exe do the encryption

- **File Creation**:
  - tasksche.exe: Responsible for encryption and generating others .exe and files.
  - 00000000.pky: Contains the encrypted RSA key.
  - decryptor.exe : Responsible for display the bitcoin wallet address and decrypt whenenter the right key after paying .

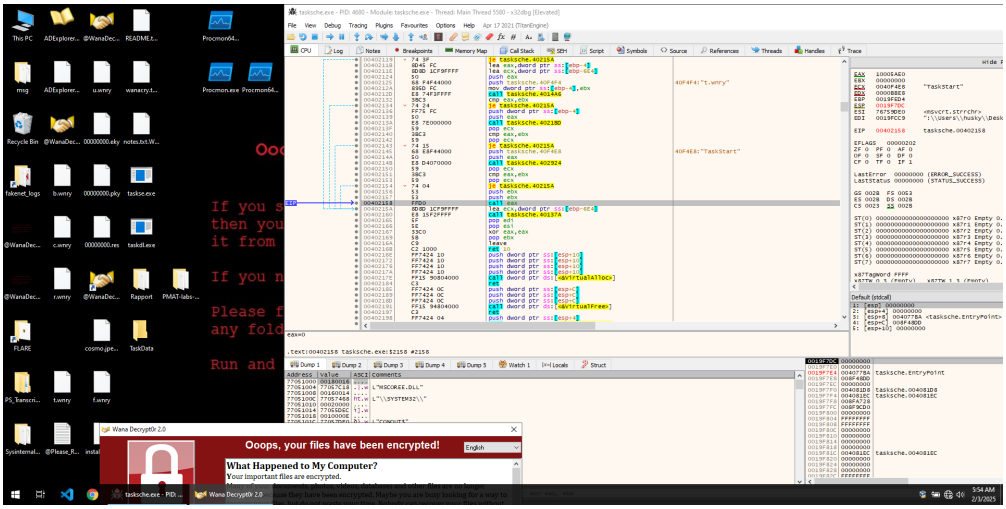


Figure 20: tasksche.exe create files

9 Rules & Signatures

A full set of YARA rules is included in Appendix A.

10 Appendices

10.1 A. Yara Rules

You can Check all yara rules in my github : S4L1Mabd

10.1.1 First Rule

```
rule Milicious_Domain{
    meta:
        description = "This Rule For detecting the Malicious Domain and IPs In wannacary
            Rans"
        author = "Salim ABDOUNE"
        date = "2025-02-03"

    strings:
        $domain = "http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com" ascii


    condition:
        $domain
}
```

Listing 1: Rule N'01

10.1.2 Second Rule

```
rule DetectAsInvoker {
    meta:
        description = "Detects requestedExecutionLevel set to asInvoker"
        author = "Salim ABDOUNE"
        date = "2025-02-03"

    strings:
        $asInvoker = "<requestedExecutionLevel level=\"asInvoker\" />"

    condition:
        $asInvoker
}
```

Listing 2: Rule N'02

10.1.3 Third Rule

```
rule ProofOfWannacry {
    meta:
        description = "Detects Wannacry-related file extensions"
        author = "Salim ABDOUNE"
        date = "2025-02-03"

    strings:
        $prof1 = ".wnry0"
        $prof2 = ".wnry"
        $prof3 = ".pky"
        $prof4 = ".eky"

    condition:
        any of ($prof1, $prof2, $prof3, $prof4)
}
```

Listing 3: Rule N'0

10.2 B. Callback URLs

Domain	Port
http://www.iugerfsodp9ifjaposdfjhgosurijfaewrwergwea.com	80

Table 2: Callback URLs

10.3 C. Decompiled Code Snippets

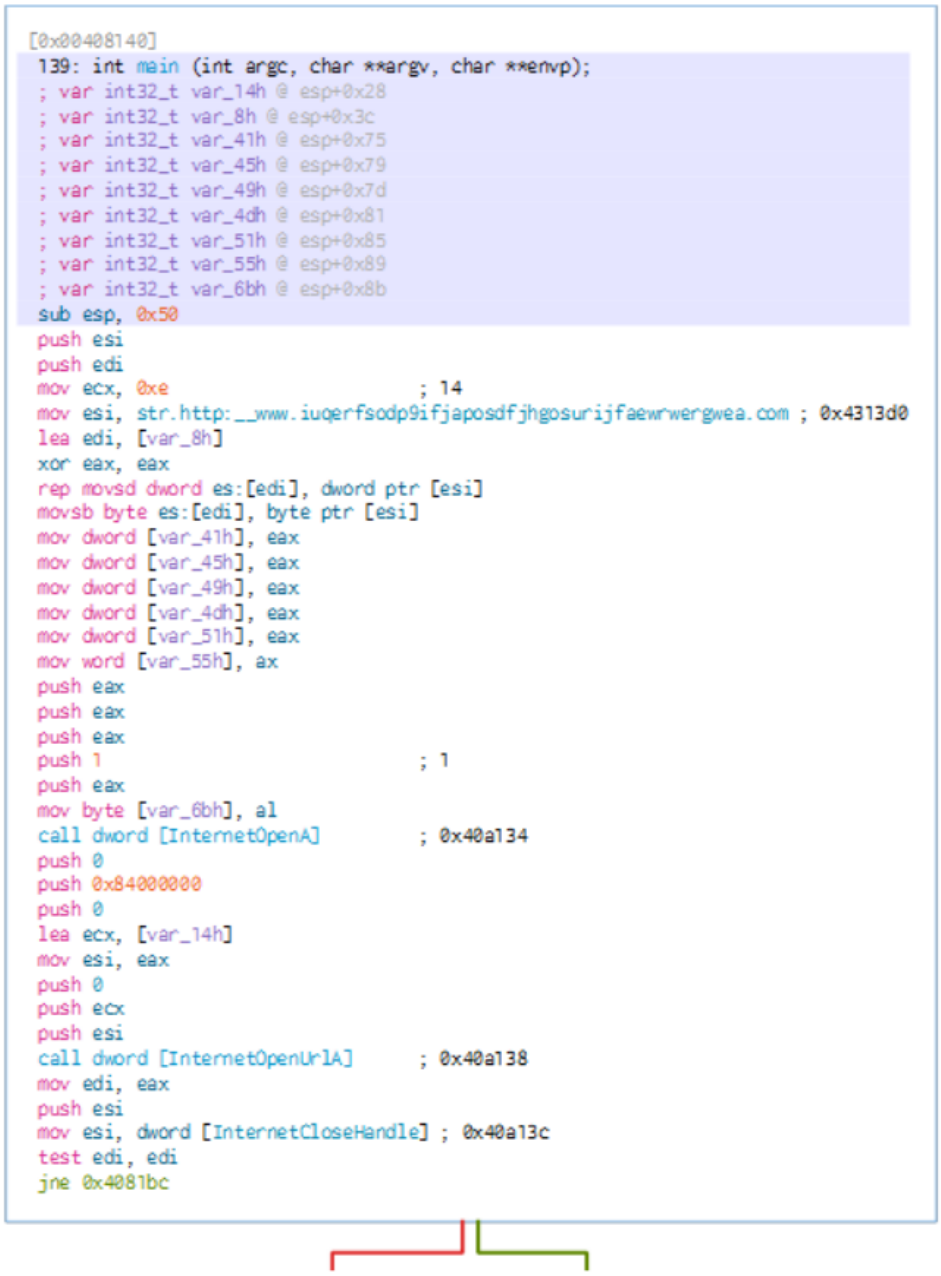


Figure 21: killswitch Routine in Cutter