Technical Analysis Report: Artifact "AgentTesla.exe"

Data Analisi: 02 Febbraio 2026
Studente: Nicola Cassandra
Livello: TLP:WHITE (Divulgabile)

---

1. Introduzione e Obiettivi dell'Analisi

Il presente documento riporta i risultati della **Basic Static Analysis** (BSA) condotta sul sample identificato come `AgentTesla.exe`. L'obiettivo primario di questa fase è l'estrazione di Indicatori di Compromissione (IoC) e la determinazione delle capacità del malware senza la sua esecuzione, al fine di mitigare i rischi di infezione accidentale dell'ambiente di analisi.

La metodologia applicata segue lo standard industriale per il triage dei malware:

1. **Hashing:** Identificazione univoca del file.
2. **PE Analysis:** Ispezione della struttura del Portable Executable.
3. **Fingerprinting:** Identificazione di compilatori e packer.
4. **String Analysis:** Estrazione di stringhe ASCII/Unicode per inferire funzionalità.

---

2. Identificazione del Campione (Hashing)

Strumento Utilizzato: *HashMyFiles*
L'hashing è il primo step fondamentale per cristallizzare la prova digitale e verificare se il sample è già noto alle piattaforme di Threat Intelligence.

Dalle evidenze raccolte:

- **Filename:** `AgentTesla.exe`
- **Dimensione File:** 2,932,642 bytes (circa 2.80 MB)
- **MD5:** `cce284cab135d9c0a2a64a7caec09107`
- **SHA-256:** `18aab0e981eee9e4ef8e15d4b003b14b3a1b0bfb7233fade8ee4b6a22a5abbb9`

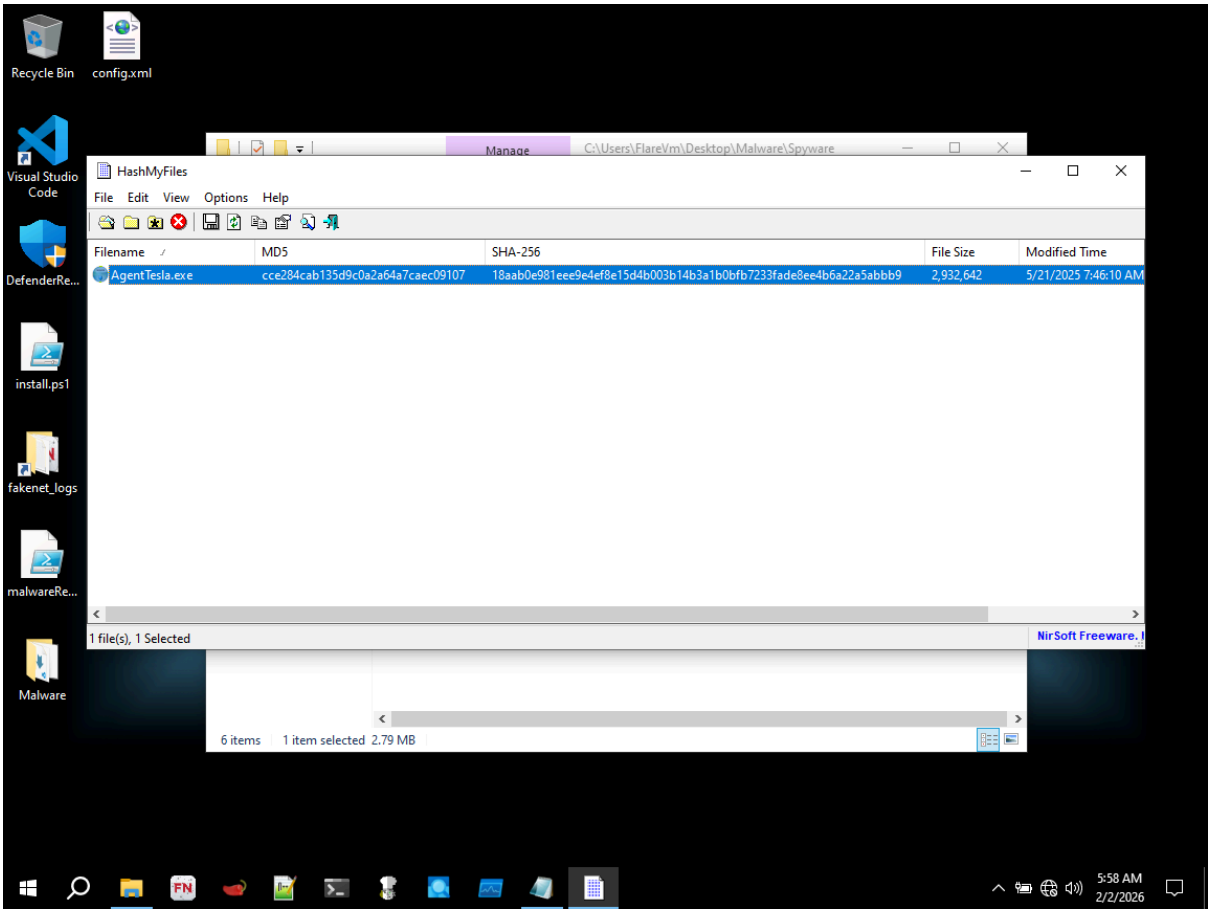


Figura 1: Calcolo degli hash crittografici tramite HashMyFiles.

---

3. Analisi della Struttura PE (Portable Executable)

Strumento Utilizzato: *CFF Explorer* CFF
Explorer permette la dissezione degli header del file binario. L'analisi degli header *File* e *Optional* fornisce dettagli critici sull'architettura target e sulla modalità di esecuzione.

3.1 Caratteristiche Architetturali

Dall'analisi del **File Header**, il campo `Machine` riporta il valore `014C (Intel 386)`.

- **Deduzione Tecnica:** Il malware è compilato per architettura **x86 (32-bit)**. Questo garantisce la retrocompatibilità (WoW64) permettendo l'esecuzione sia su sistemi target a 32 che a 64 bit.

3.2 Timestamp di Compilazione

Il campo `TimeDateStamp` riporta il valore esadecimale `5DF6D4E7`.

- **Analisi:** Questo timestamp indica la data in cui il linker ha generato il file. Sebbene i malware author utilizzino spesso tecniche di *Timestomping* (falsificazione della data),

questo valore deve essere correlato con le campagne di infezione note.

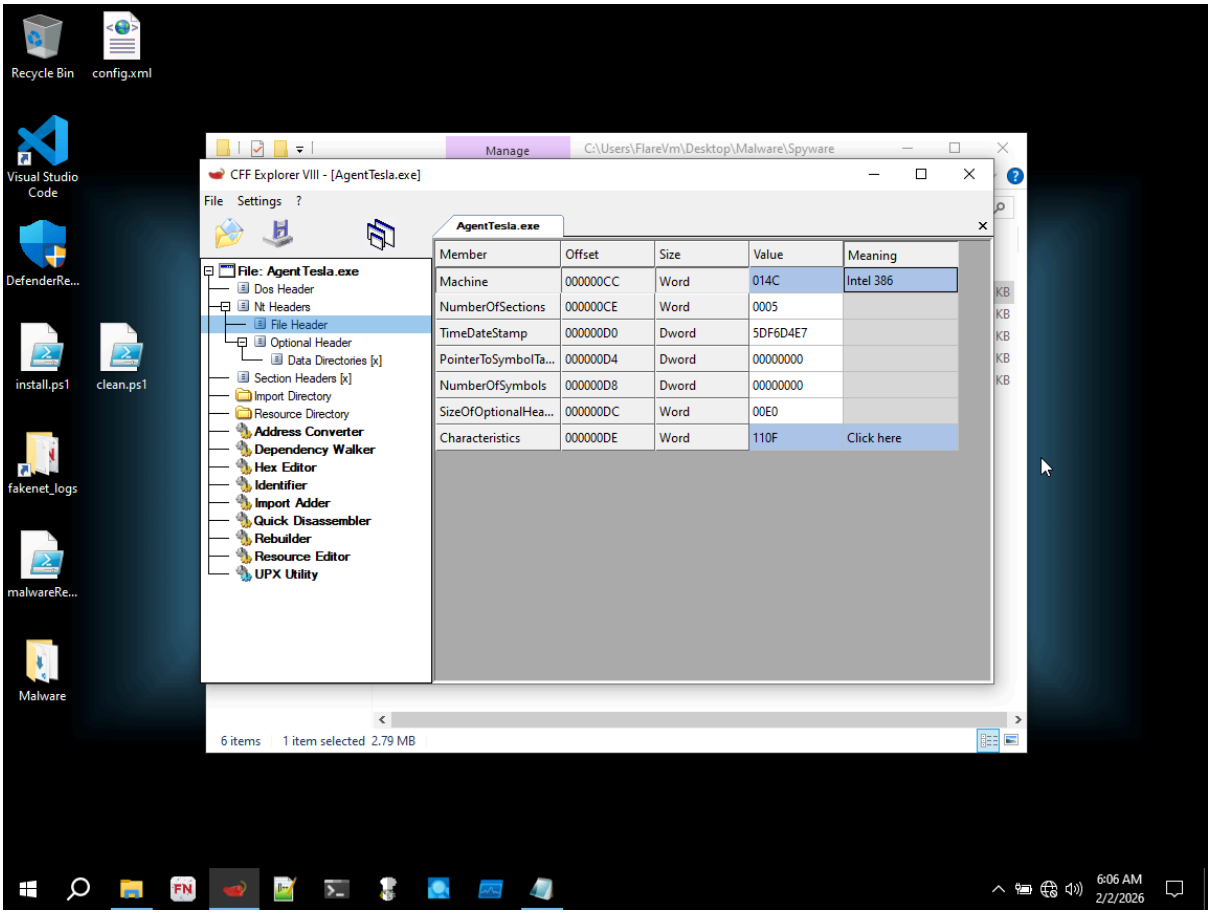
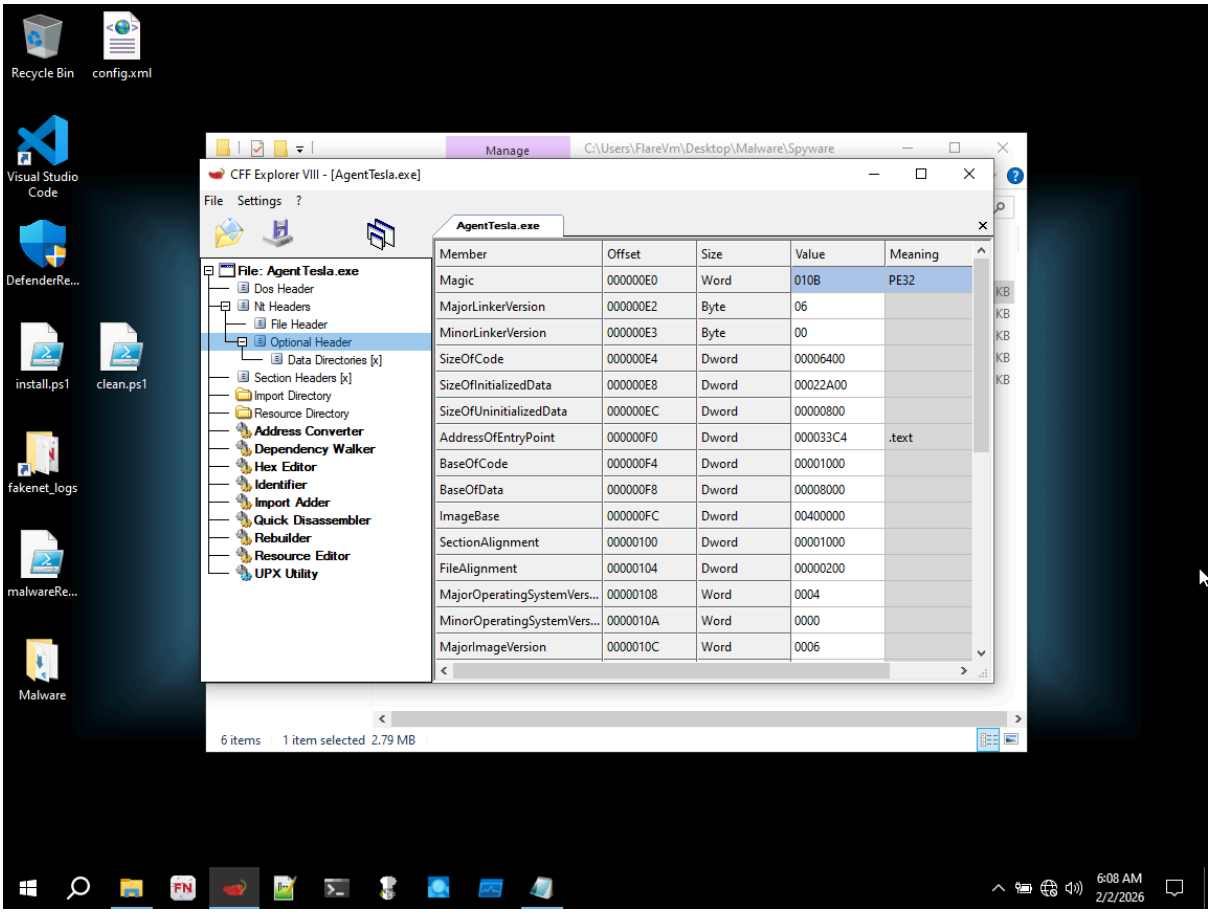


Figura 2: File Header in CFF Explorer che mostra architettura x86 e timestamp.

3.3 Sottosistema ed Entry Point

L'**Optional Header** rivela:

- **Subsystem:** `0002 (Windows GUI)`.
  - *Significato:* Il processo non alloca una console al momento dell'avvio. Gira in background o presenta un'interfaccia grafica, comportamento tipico per evitare il rilevamento visivo da parte dell'utente.
- **AddressOfEntryPoint:** `000033C4`.
  - *Significato:* Indirizzo di memoria relativo (RVA) della prima istruzione eseguita.


Recycle Bin
config.xml
Visual Studio Code
DefenderRe...
install.ps1
clean.ps1
fakenet_logs
malwareRe...
Malware
Manage
C:\Users\FlareVm\Desktop\Malware\Spyware
CFF Explorer VIII - [AgentTesla.exe]
File Settings ?
AgentTesla.exe
File: AgentTesla.exe
Dos Header
Nt Headers
File Header
Optional Header
Data Directories [x]
Section Headers [x]
Import Directory
Resource Directory
Address Converter
Dependency Walker
Hex Editor
Identifier
Import Adder
Quick Disassembler
Rebuilder
Resource Editor
UPX Utility
Member | Offset | Size | Value | Meaning
Magic | 000000E0 | Word | 010B | PE32
MajorLinkerVersion | 000000E2 | Byte | 06
MinorLinkerVersion | 000000E3 | Byte | 00
SizeOfCode | 000000E4 | Dword | 00006400
SizeOfInitializedData | 000000E8 | Dword | 00022A00
SizeOfUninitializedData | 000000EC | Dword | 00000800
AddressOfEntryPoint | 000000F0 | Dword | 000033C4 | .text
BaseOfCode | 000000F4 | Dword | 00001000
BaseOfData | 000000F8 | Dword | 00008000
ImageBase | 000000FC | Dword | 00400000
SectionAlignment | 00000100 | Dword | 00001000
FileAlignment | 00000104 | Dword | 00000200
MajorOperatingSystemVers... | 00000108 | Word | 0004
MinorOperatingSystemVers... | 0000010A | Word | 0000
MajorImageVersion | 0000010C | Word | 0006
6 items
1 item selected 2.79 MB
6:08 AM
2/2/2026

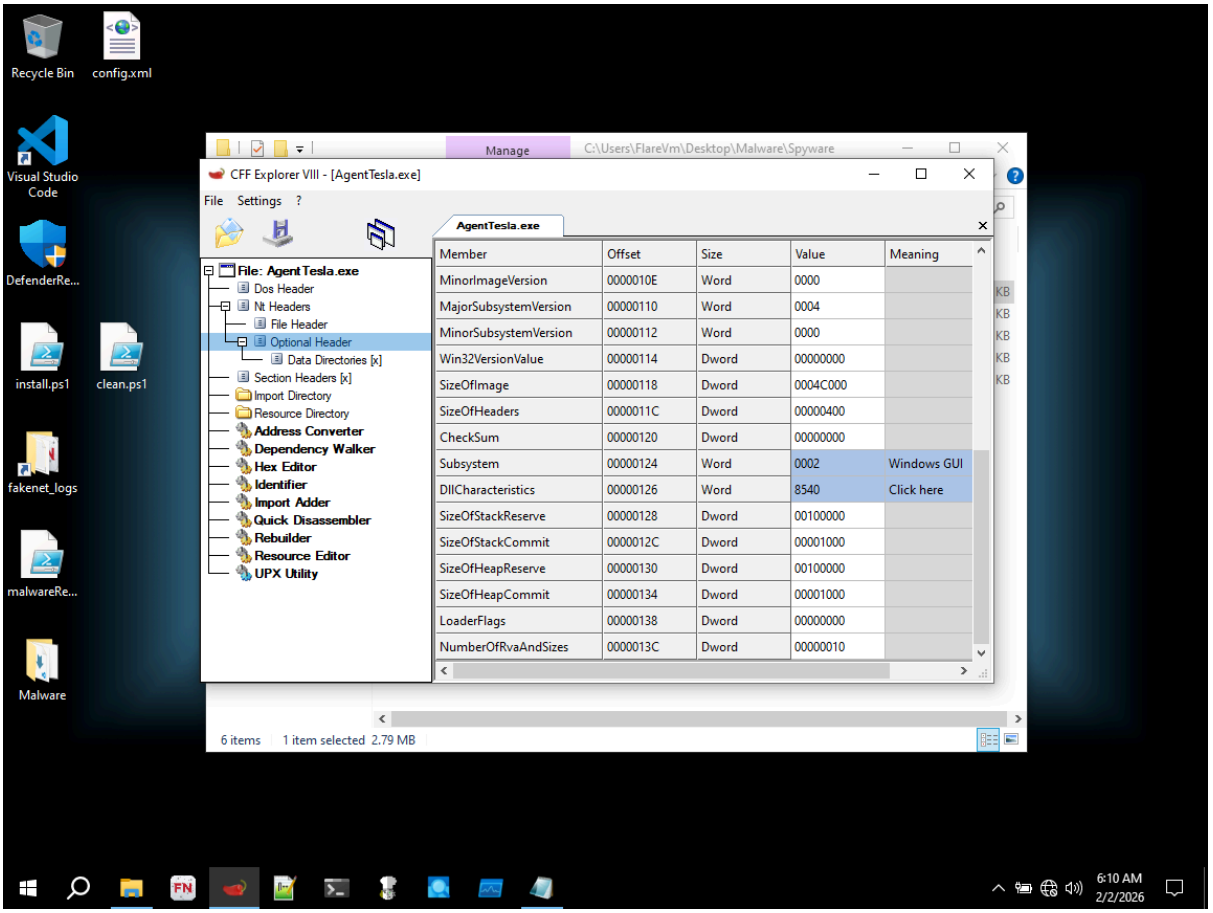


Figura 3: Dettagli dell'Optional Header: Entry Point e Subsystem GUI.

---

4. Fingerprinting e Rilevamento Packer

Strumento Utilizzato: *Detect It Easy (DiE)*
Questa fase è cruciale per comprendere se il codice malevolo è offuscato. DiE analizza l'entropia e le firme binarie per identificare compilatori e packer.

Risultati dell'Analisi

L'analisi evidenzia una discrepanza significativa rispetto al nome del file (AgentTesla è noto come malware .NET), rivelando la vera natura del vettore d'attacco:

- **Packer/Installer:** `Nullsoft Scriptable Install System (3.05) [lzma]`
- **Linguaggio Rilevato:** `C`
- **Overlay:** Rilevata presenza di dati NSIS

Valutazione Tecnica: Il file analizzato **non è il payload finale**, ma un **Dropper** o **Loader** creato con **NSIS** (un sistema di installazione scriptabile legittimo, spesso abusato dai criminali informatici). Il malware vero e proprio (probabilmente un payload .NET/C#) è compresso e crittografato all'interno della sezione "Overlay" dell'installer. Al momento dell'esecuzione, lo script NSIS estrarrà ed eseguirà il payload malevolo. Questo spiega perché DiE rileva "C" invece di ".NET".

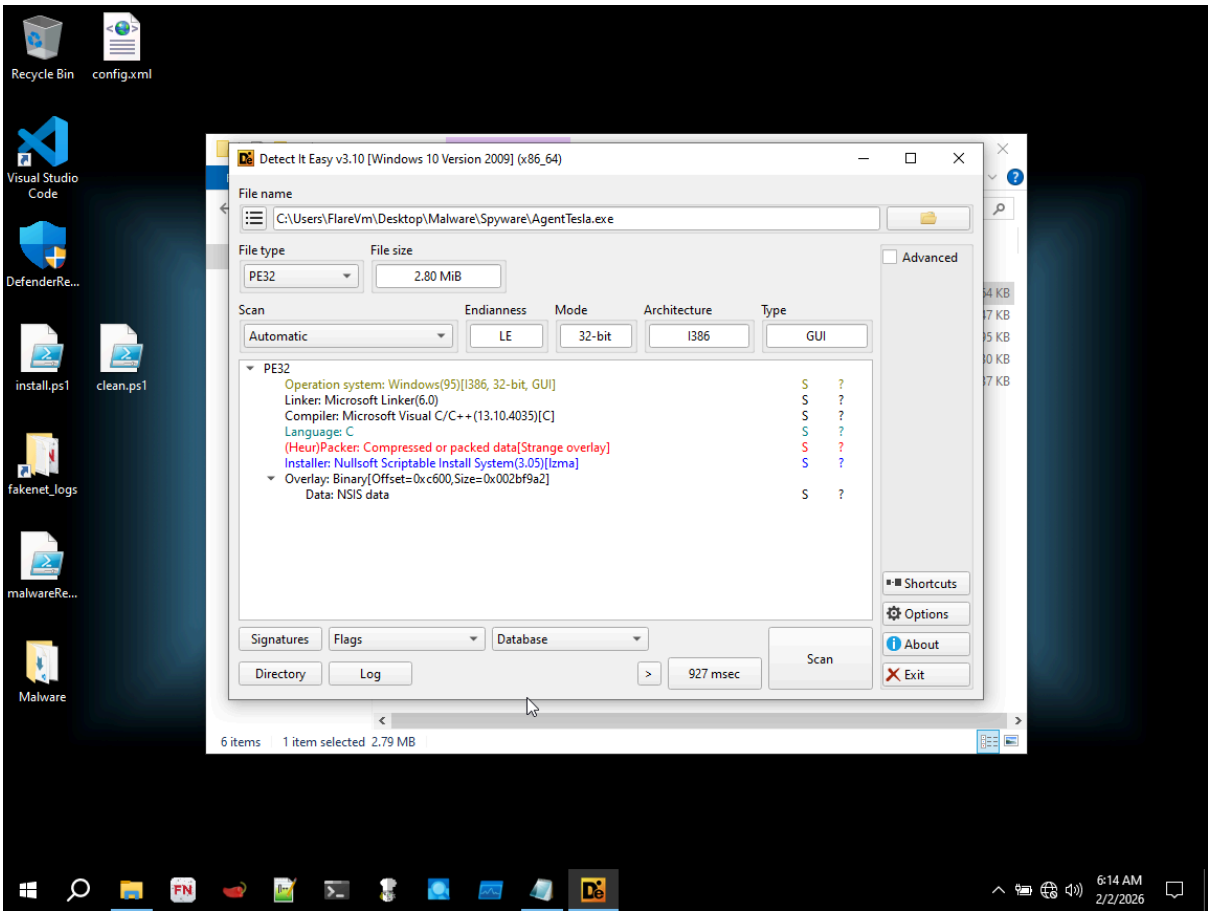


Figura 4: Analisi entropica e rilevamento firme con Detect It Easy (DiE). Rilevato installer NSIS.

---

5. Analisi delle Stringhe e Import API

Strumento Utilizzato: *Strings / BinText* L'estrazione delle stringhe ASCII/Unicode permette di identificare le chiamate alle API di Windows (Import Table), svelando le capacità operative del software.

Dallo screenshot, sono state isolate le seguenti API critiche:

API Call	Categoria	Funzionalità Sospetta
`CreateDirectoryW`	File System	Creazione di directory (spesso in `%AppData%` o `%Temp%` per nascondersi).
`WriteFile`	I/O	Scrittura del payload estratto su disco.
`CreateProcessW`	Process Execution	Esecuzione del payload appena droppato (o *Child Process*).
`MoveFileExW`	File System	Spesso usato per rinominare file o pianificare cancellazioni al riavvio.
`GetTempFileNameW`	File System	Generazione di nomi file casuali per eludere rilevamenti statici basati su nomi fissi.

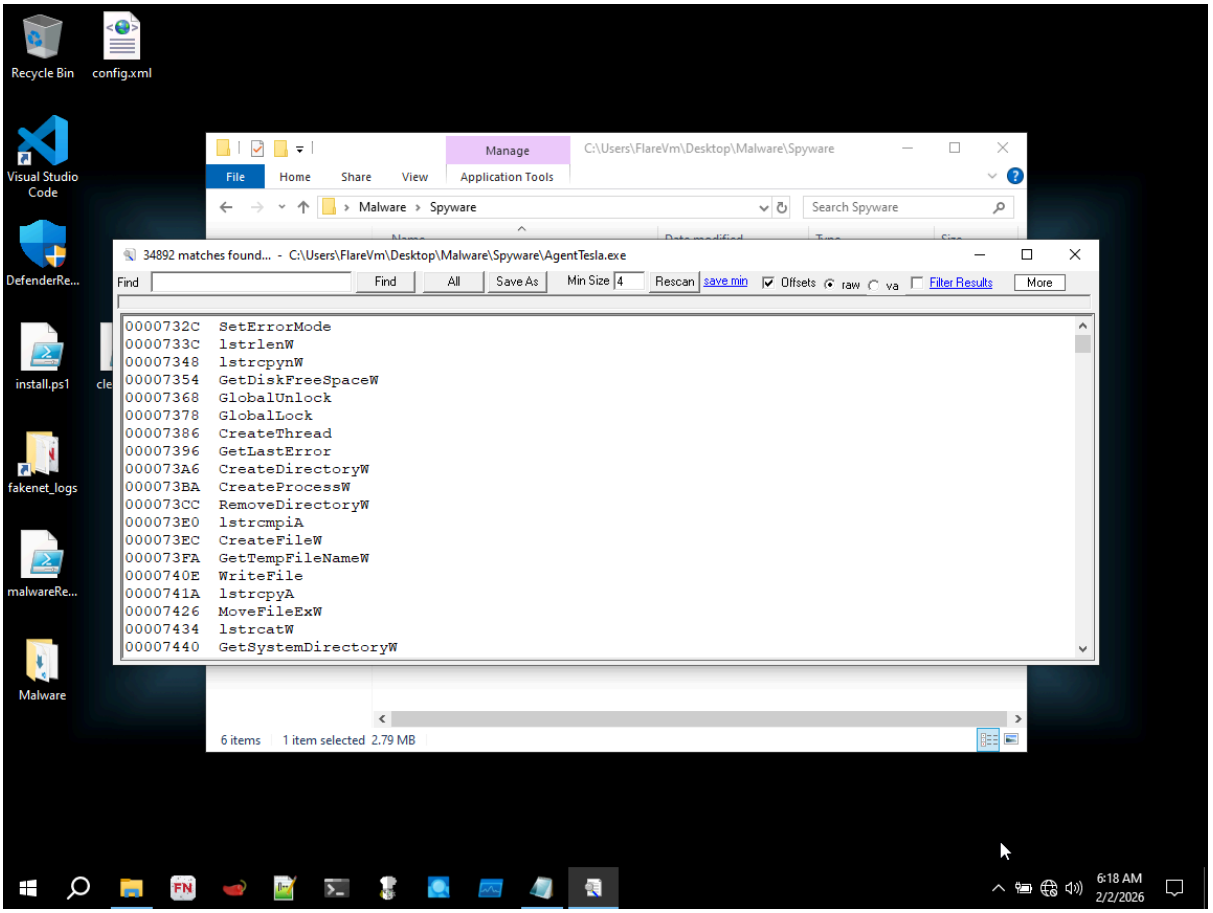


Figura 5: Estrazione stringhe. Evidenziate le chiamate API per la manipolazione di file e processi.

6. Conclusioni e Raccomandazioni

L'analisi statica conferma che il file `AgentTesla.exe` è un **NSIS Dropper**. Non contiene direttamente il codice malevolo in chiaro, ma funge da vettore di consegna. Le evidenze (API di scrittura file ed esecuzione processi) indicano che, una volta avviato, il sample "dropperà" un secondo eseguibile (il vero Agent Tesla) ed eseguirà la sua routine di spionaggio.