S9 – L1
Analisi Statica Malware Agent Tesla (FlareVM)

Introduzione

In questo esercizio ho eseguito un'attività di **analisi statica di un malware di tipo Agent Tesla** utilizzando l'ambiente di laboratorio **FlareVM**.
L'obiettivo è stato **identificare le principali caratteristiche del campione senza eseguirlo, applicando tecniche di fingerprinting, analisi della struttura PE, rilevamento del linguaggio/packer e analisi delle stringhe.**

0) Estraggo Agent Tesla

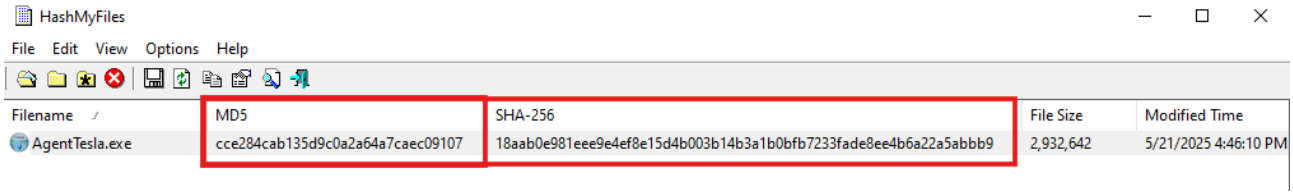
1. Copio l'archivio di **Agent Tesla** sul **Desktop** di FlareVM.
2. Tasto destro sull'archivio → **7-Zip** → **Extract to…**
3. Quando chiede la password scrivo: **infected** (**non è stata richiesta la password**)
4. Controllo che nella cartella estratta ci sia il file del sample (es. `.exe`) **senza eseguirlo**.

Output: **archivio sul Desktop + cartella estratta** + richiesta password (non richiesta)

---

1) Calcolo gli hash (Fingerprint)

5. Tasto destro sul file estratto → **HashMyFiles** (oppure lo apro e trascino il file).
6. Copio nel report:
    - **MD5**
    - **SHA256**

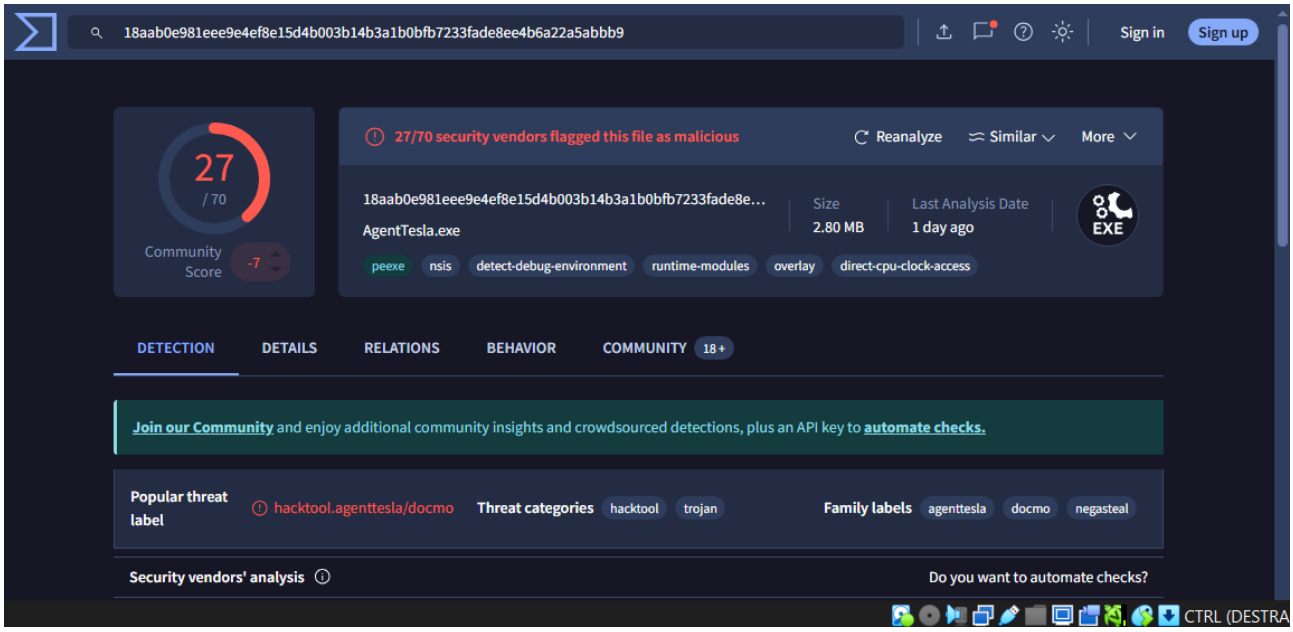


Output: HashMyFiles con **MD5 e SHA256** visibili.

---

2) Controllo su VirusTotal (da hash)

7. Copio lo **SHA256** e lo cerco su **VirusTotal** (search by hash).
8. Tramite la ricerca dello SHA256 su VirusTotal, il campione è risultato segnalato come malevolo da **27 su 70** motori antivirus.
   Il file è classificato come **hacktool / trojan** ed è associato alla famiglia **AgentTesla**, con ulteriori etichette quali *docmo* e *negasteal*.



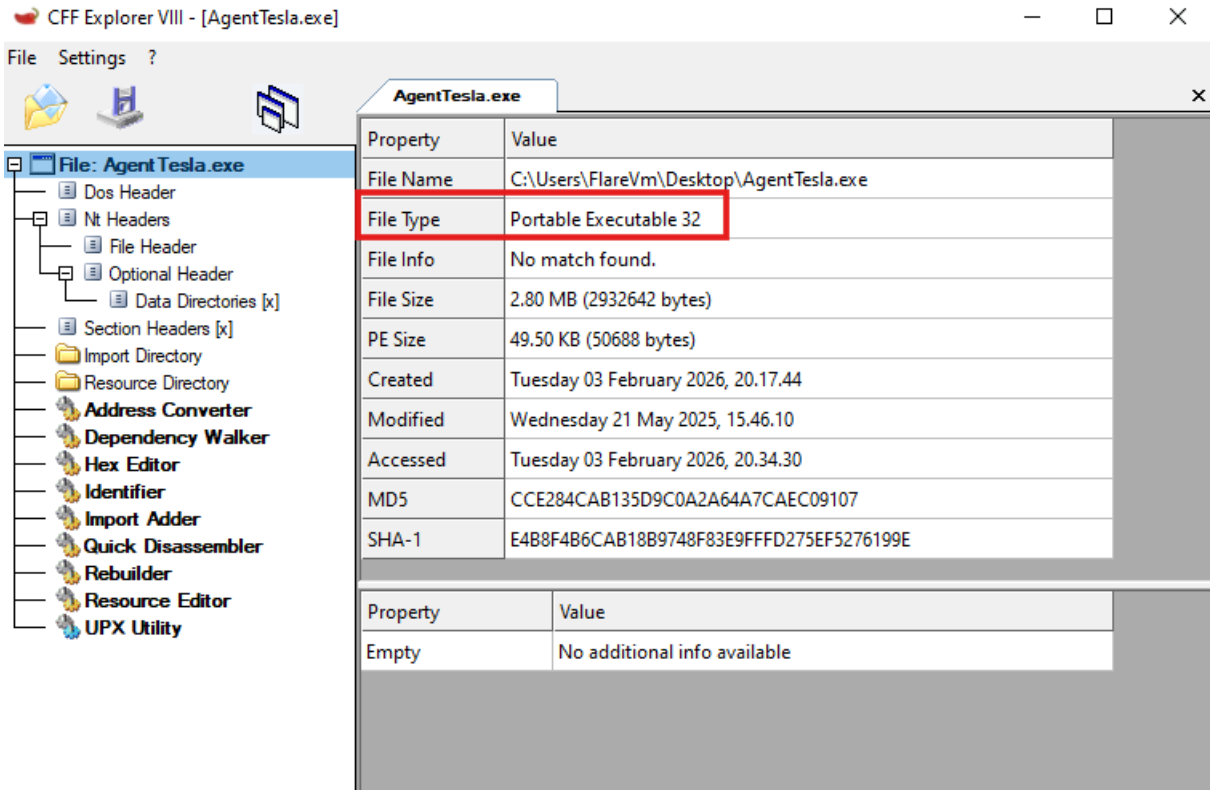
Output: **pagina VirusTotal con detection**

---

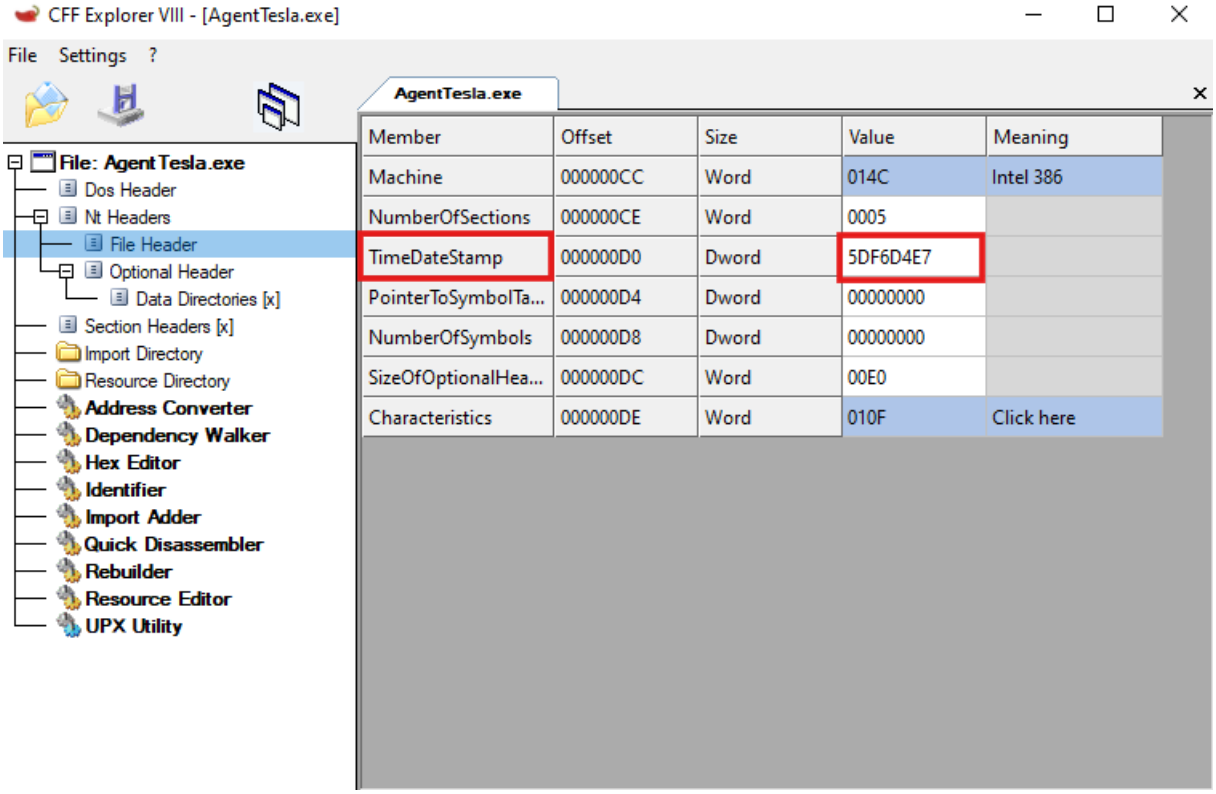
3) Analisi PE (struttura del file)

9. Apro il file con **PEStudio** (o **CFF Explorer**).
10. Nel report segno:

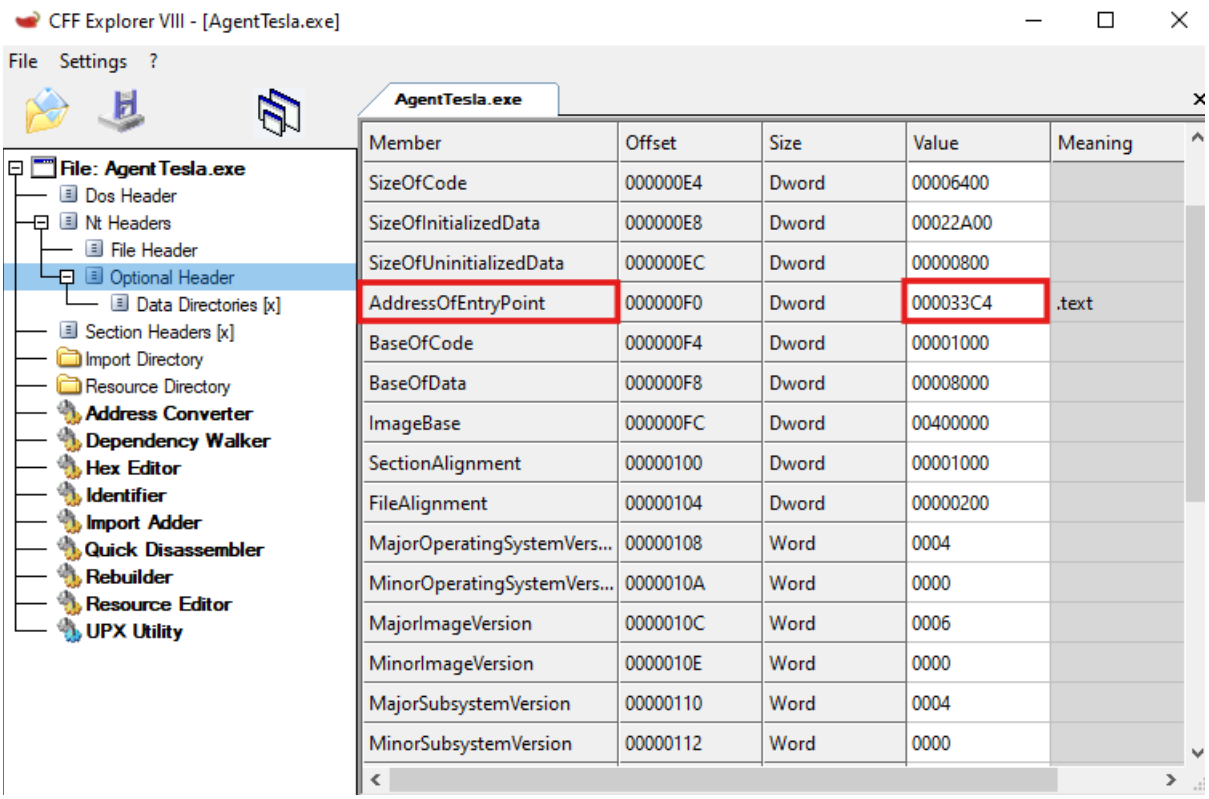
- **Architettura** (x86 / 32 Bit) --- (**File Type: Portable Executable 32)**



- **Timestamp --- TimeDateStamp (Value):** `5DF6D4E7`

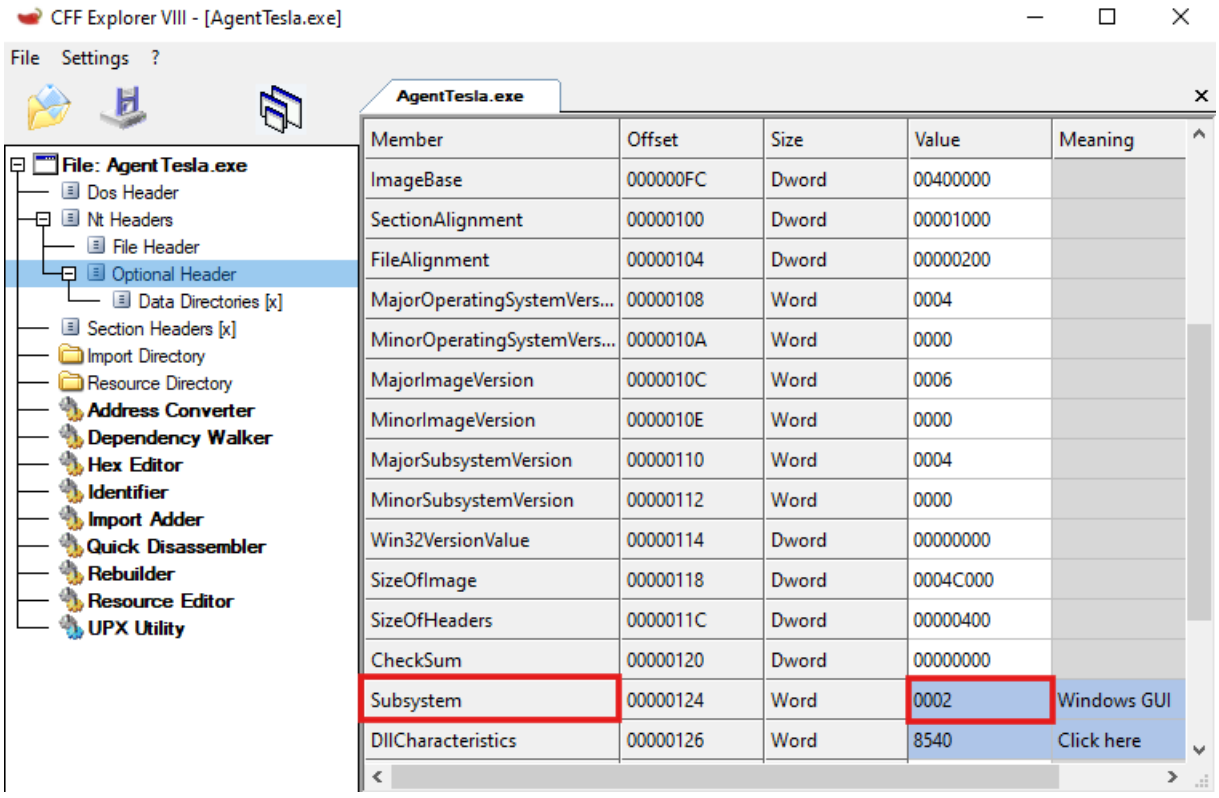


- **Entry Point (AddressOfEntryPoint = 000033C4)**



- **Subsystem** (GUI/Console)

  Il **Subsystem** del file è **Windows GUI**, come indicato dal campo *Subsystem* dell'Optional Header.



Output: schermata CFF Explorer con i campi visibili.

---

4) Rilevo packer/linguaggio (.NET o altro)

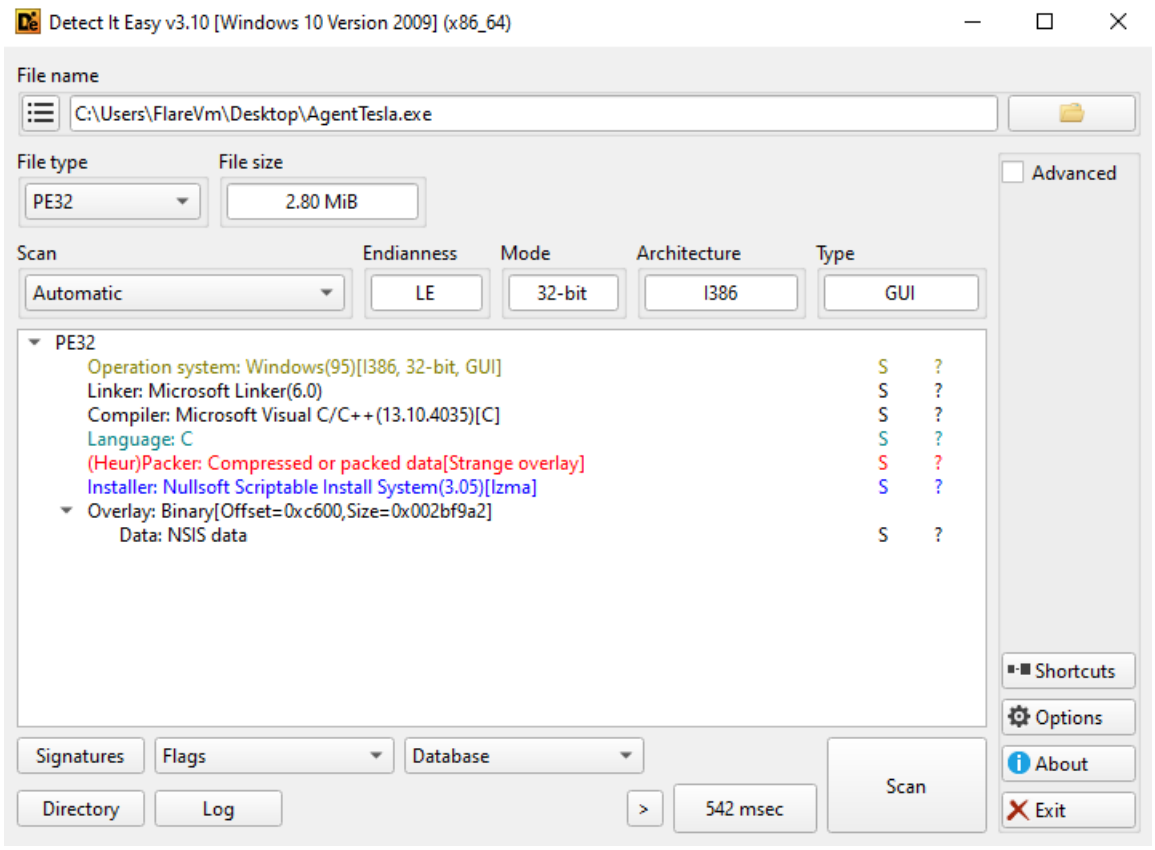
11. Apro il file con **Detect It Easy (DiE)**

12. **Dati tecnici rilevati (dal tool)**

- **Tipo file**: PE32
- **Architettura**: 32-bit (i386)
- **Sistema operativo**: Windows (95–11), GUI
- **Compiler**: Microsoft Visual C/C++
- **Linguaggio**: C/C++
- **Installer / Packer**: **NSIS (Nullsoft Scriptable Install System)**
- **Overlay presente** (dato compresso/packed)

13. **L'analisi del campione tramite Detect It Easy ha evidenziato che il file è un PE32 a 32 bit per sistemi Windows.**

14. **Il malware risulta compilato in C/C++ mediante Microsoft Visual C/C++ ed è confezionato tramite NSIS (Nullsoft Scriptable Install System), con presenza di overlay, indicativo di dati compressi o impacchettati all'interno dell'eseguibile.**

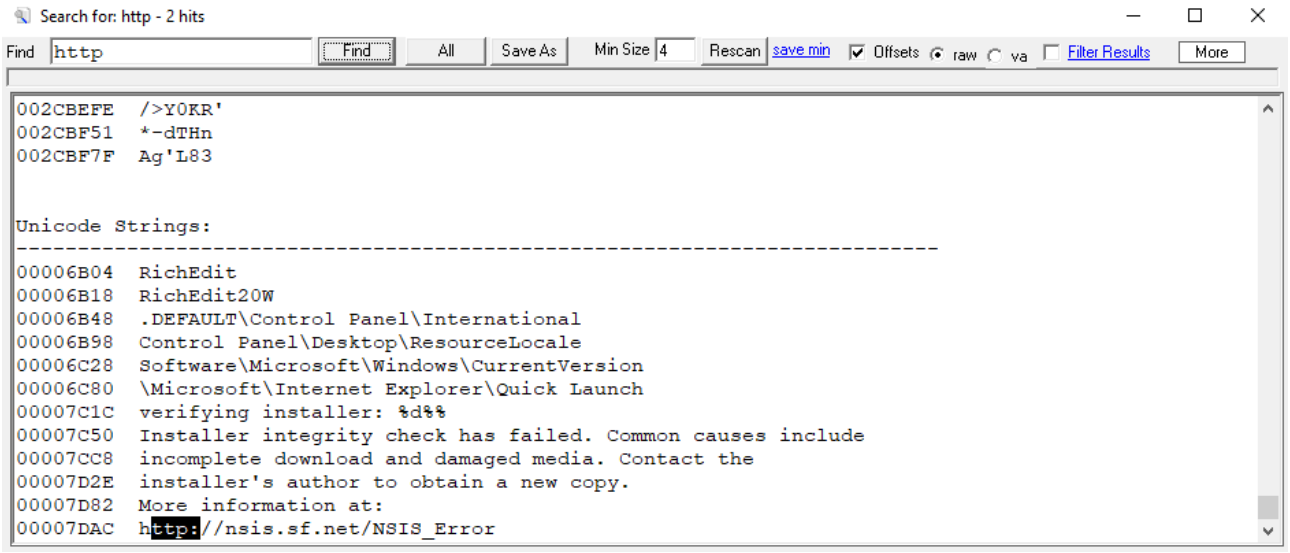


📸 Output: DiE con risultato visibile.

---

5) Estraggo stringhe e IoC

13. Estraggo le stringhe (tool “**Strings**”)
14. Dall’analisi delle stringhe statiche del file sono stati individuati Indicatori di Compromissione (IoC).
    In particolare **è stato rilevato un riferimento a un URL (`http://nsis.sf.net/NSIS_Error`), coerente con l’utilizzo del framework NSIS per il confezionamento dell’eseguibile.**



Output: presenza di un **URL**, confermando l’utilizzo di **NSIS**.

Conclusioni

L’analisi statica ha permesso di **raccogliere informazioni rilevanti sul malware Agent Tesla** senza comprometterne l’esecuzione, **individuando hash, struttura dell’eseguibile, linguaggio di sviluppo e principali indicatori di compromissione.**
Questa fase rappresenta un passaggio fondamentale per comprendere il comportamento di un malware e **costituisce la base per eventuali analisi dinamiche o attività di difesa e threat intelligence.**