**TRACCIA 1**

In questa analisi ho esaminato il file **AdwereCleaner.exe** per identificare possibili indicatori di compromissione senza eseguirlo. L'obiettivo principale è stato quello di raccogliere informazioni utili per determinare se il file potesse rappresentare una minaccia e, in tal caso, quali misure adottare per mitigarne il rischio.

Per condurre l'analisi ho utilizzato **un approccio statico**, ovvero senza l'esecuzione diretta del malware, riducendo così il rischio di infezione. Questo tipo di analisi permette di ottenere dettagli sulla struttura del file, sulle **API utilizzate**, sulle **stringhe contenute nel codice**, e di verificare se il file fosse già noto ai database di sicurezza.

Per garantire la sicurezza durante l’analisi, ho utilizzato una **macchina virtuale Kali Linux**, ambiente ideale per la cybersecurity, dove ho impiegato strumenti di scansione e reverse engineering.

## 

## **1. Configurazione dell'Ambiente di Analisi**

Per eseguire l’analisi in sicurezza, ho configurato una **macchina virtuale Kali Linux** con connessione **NAT**, garantendo così l’accesso ai servizi online per la verifica del file senza esporre il sistema reale a rischi.

Per un’analisi efficace, ho utilizzato diversi strumenti:

* **VirusTotal** → Per verificare se il file fosse già noto come malware.
* **sha256sum, md5sum, sha1sum** → Per calcolare gli hash crittografici e confrontarli con database di sicurezza.
* **strings** → Per estrarre stringhe leggibili dal file ed evidenziare possibili indicatori di minaccia.
* **objdump** → Per analizzare le API di Windows utilizzate dal file.
* **UPX** → Per verificare la presenza di offuscamento o compressione del file.

Questa configurazione mi ha permesso di raccogliere **informazioni dettagliate** senza necessità di eseguire il file, riducendo i rischi e garantendo un’analisi controllata.

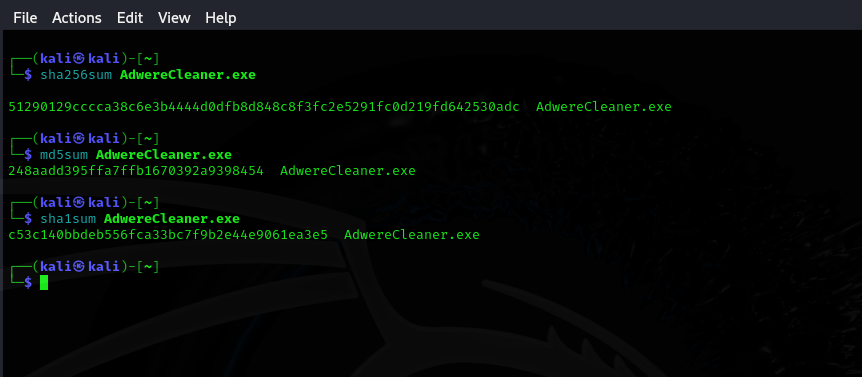
## 

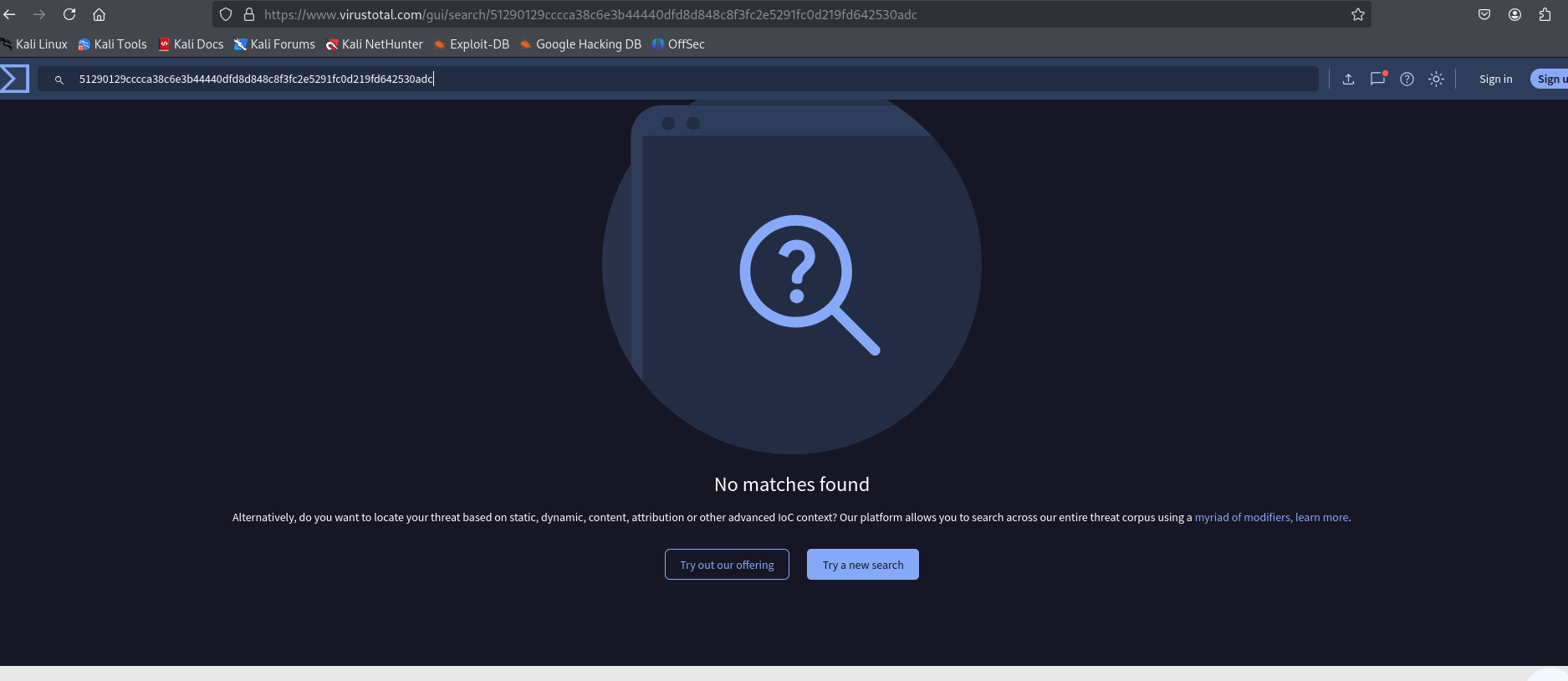
## **2. Download e Verifica del File**

Ho scaricato AdwereCleaner.exe da **GitHub** e ,dopo il download, ho effettuato una **verifica dell'integrità e dell'unicità del file** utilizzando il calcolo degli **hash crittografici** che servono per identificare in modo univoco il file e verificarne eventuali alterazioni:

* **SHA-256** → Uno degli algoritmi più sicuri per verificare l'integrità di un file.
* **MD5** → Algoritmo più veloce ma meno sicuro, usato per verifiche rapide.
* **SHA-1** → Simile a MD5, ancora utilizzato in alcuni database antivirus per confronti.

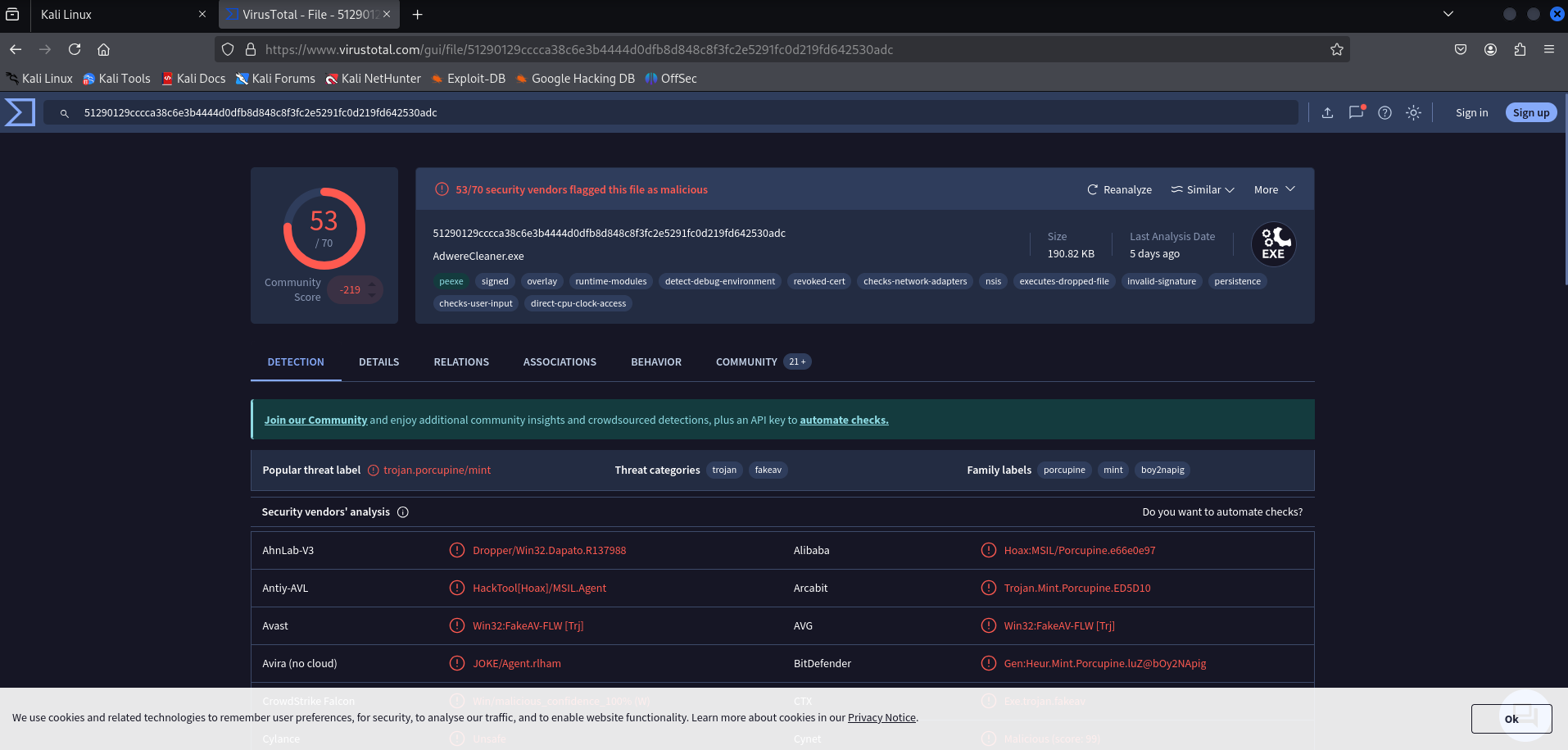
Dopo aver ottenuto gli hash, ho eseguito una ricerca su **VirusTotal** per vedere se il file fosse già stato analizzato.





\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

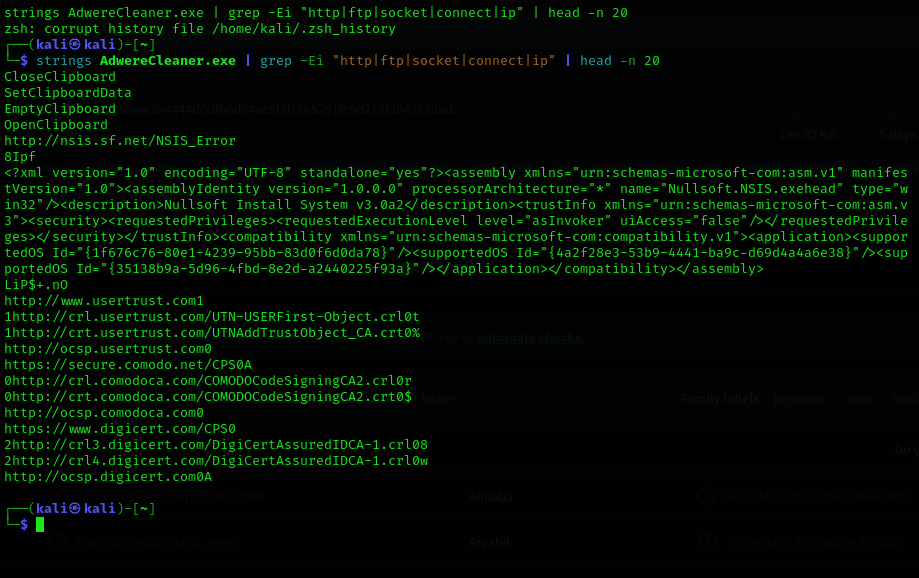
**3. Analisi tramite VirusTotal**

VirusTotal ha confermato che il file è altamente sospetto, con **53 su 70 motori antivirus** che lo hanno identificato come **Trojan/FakeAV**. Questo tipo di malware è comunemente usato per **ingannare gli utenti**, facendosi passare per software legittimo mentre esegue attività dannose come il furto di credenziali o il download di altri malware.

## **4. Estrazione e Analisi delle Stringhe**

Le stringhe contenute nel file possono rivelare **indirizzi sospetti, comandi nascosti o chiavi di registro modificate**. Analizzando il file con strings, ho trovato:

* **Interazione con la Clipboard** (OpenClipboard, SetClipboardData), il che potrebbe indicare un possibile furto di dati copiati dagli utenti.
* **Riferimenti a NSIS Installer**, suggerendo che il file potrebbe essere un **installer malevolo**.
* **Connessioni a certificati digitali noti** (DigiCert, Comodo, UserTrust), che potrebbero essere utilizzati per bypassare i meccanismi di sicurezza.



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**5. Analisi delle API di Windows**

Per comprendere il comportamento del file, ho analizzato le **API di Windows** richiamate dal malware. Queste funzioni di sistema possono indicare **azioni sospette** che il malware potrebbe tentare di eseguire.

Le librerie principali individuate includono:

* **KERNEL32.dll** → Gestione della memoria, processi e file system.
* **USER32.dll** → Interazione con tastiera, mouse e interfaccia grafica.
* **SHELL32.dll** → Accesso al file system ed esecuzione di comandi.
* **ADVAPI32.dll** → Manipolazione del Registro di Windows.



Le API sospette individuate includono:

* **CreateProcessA** → Possibile avvio di processi dannosi.
* **RegCreateKeyExA / RegSetValueExA** → Potenziale modifica del Registro di Windows per garantire persistenza.

## **6. Verifica di Offuscamento o Packer**

Per verificare se il file fosse stato **offuscato o packato** per nascondere codice malevolo, ho utilizzato UPX, uno dei packer più diffusi:

Il test ha confermato che il file **non è packato**, il che significa che il codice malevolo è direttamente accessibile per ulteriori analisi senza necessità di decompressione.

## **Conclusioni e Raccomandazioni**

L’analisi statica ha evidenziato che AdwereCleaner.exe presenta **indicatori di compromissione significativi**:

* **Alto rischio di esecuzione di codice malevolo.**
* **Possibile furto di dati dalla clipboard.**
* **Potenziale persistenza nel Registro di Windows.**

🔴 **Raccomandazioni:**

* **NON eseguire il file in ambienti non isolati.**
* **Effettuare un’analisi dinamica in sandbox** per verificare il comportamento effettivo.
* **Monitorare il traffico di rete e il Registro di Windows** per individuare attività sospette.

📌 **Status: ALTAMENTE SOSPETTO – Ulteriori analisi necessarie.**