REPORT TECNICO

ANALISI MALWARE

notepad-classico.exe

Redatto da: *Nicolò Calì Cybersecurity Student*

Data: 03/02/2026

1. Introduzione

Il presente documento riporta l'analisi tecnica del file eseguibile denominato "**notepad-classico.exe**".

L'obiettivo dell'analisi è comprendere la natura del file, identificarne le funzionalità potenzialmente malevole e documentare gli indicatori di compromissione.

L'analisi è stata condotta in un ambiente controllato (**FlareVM**) utilizzando metodologie di analisi statica e, successivamente, dinamica.

Nome file: notepad-classico.exe

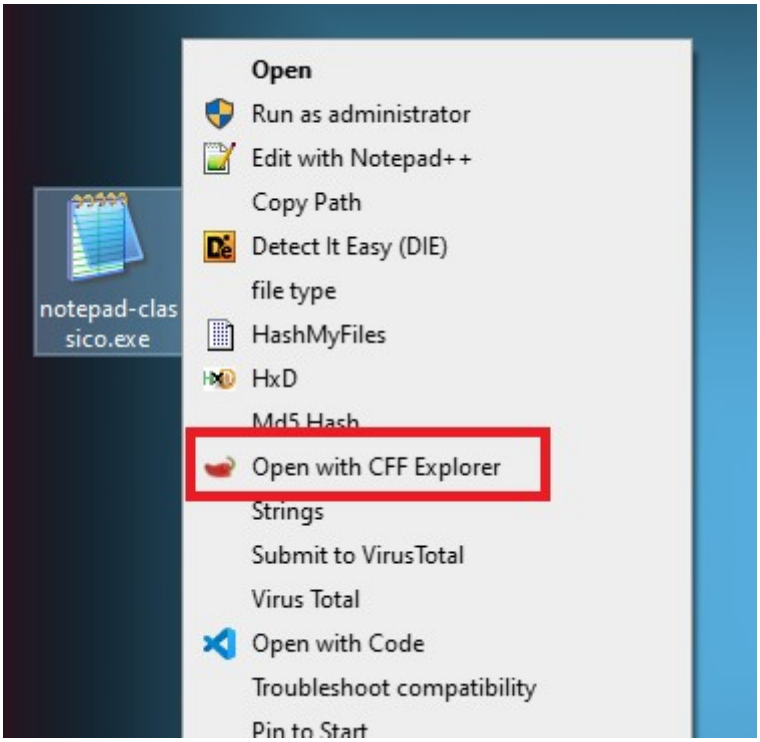
Dimensione: 626.95 KB

Hash MD5: EEAA667929C0415EA0B74B7B9E9F9C23

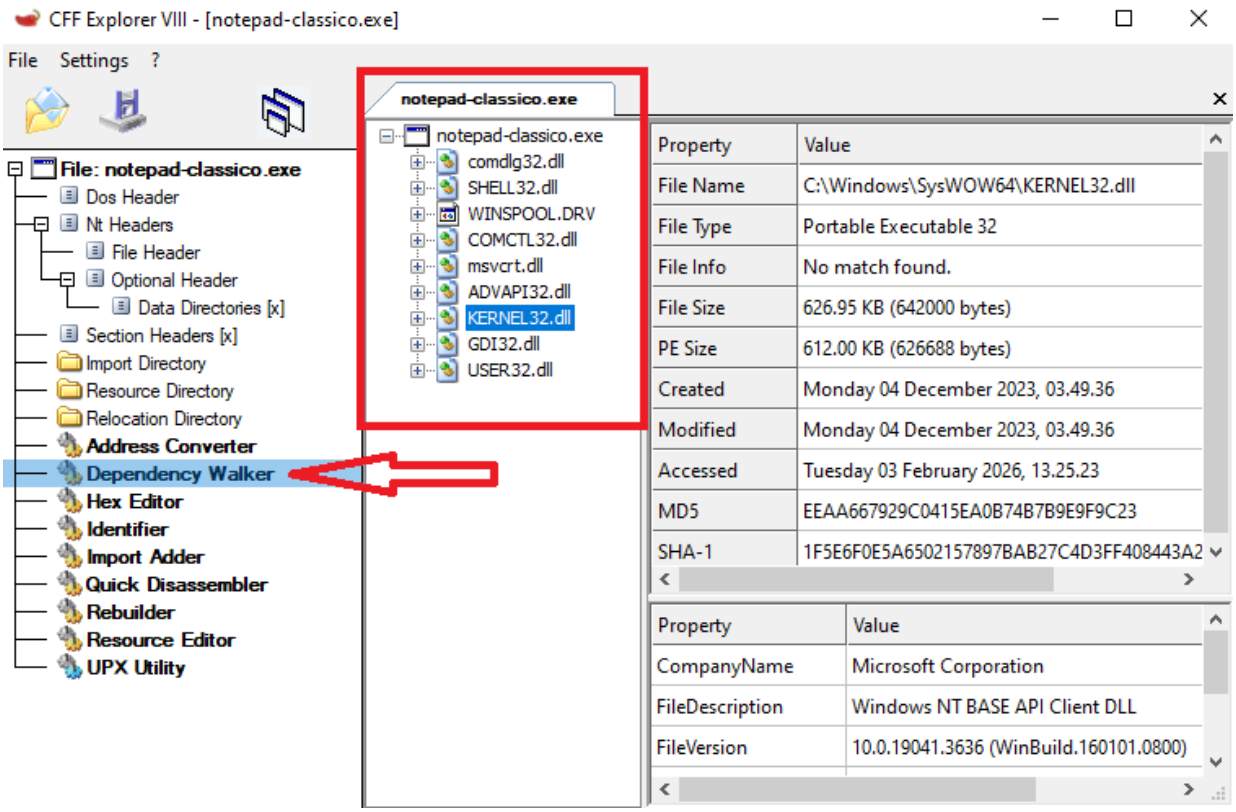
2. Analisi Statica - Librerie Importate

In questa sezione andremo ad analizzare le librerie importate dal malware, forniremo una descrizione di ognuna di esse e, per finire, andremo ad analizzare delle funzioni sospette racchiuse all'interno di queste librerie.

Per l'analisi utilizzerò il tool **CFF Explorer.**



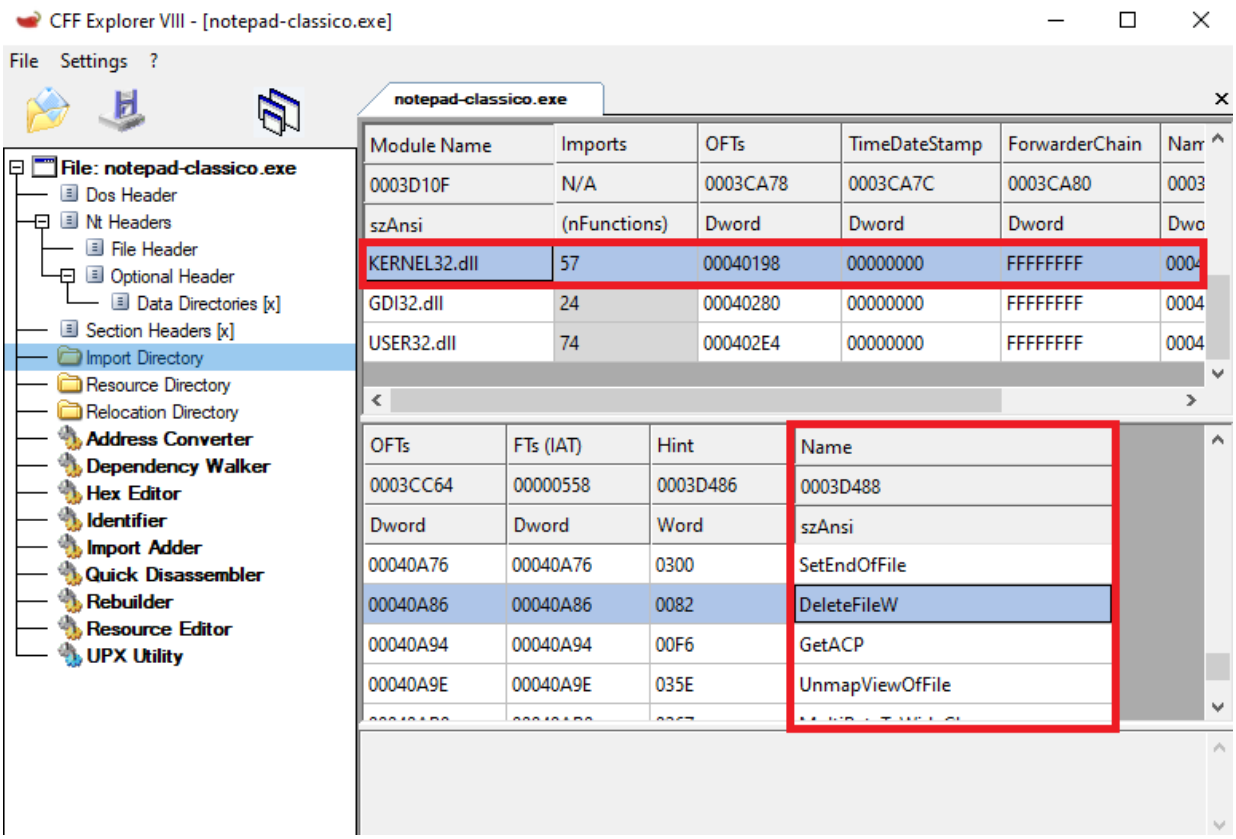
Una volta che CFF Explorer si apre andiamo nella sezione "**Dipendency Walker**" per vedere con facilità l'elenco delle Librerie utilizzate dal malware.



Capiamo il funzionamento di ognuna di esse:

- **comdlg32.dll** → Gestisce le finestre di dialogo comuni di Windows, come "Apri file", "Salva", o "Stampa". (Tipica di applicazioni con interfaccia grafica).
- **SHELL32.dll** → Fornisce funzioni per interagire con la shell di Windows, come l'apertura di file, l'esecuzione di programmi e la gestione delle icone.
- **WINSPOOL.DRV** → Contiene i driver e le funzioni per inviare documenti allo spooler di stampa (necessaria se il programma vuole stampare).
- **COMCTL32.dll** → Gestisce i controlli comuni dell'interfaccia utente moderna, come barre di stato, barre di avanzamento e tooltips.
- **msvcrt.dll** → È la libreria standard del C (Microsoft C Runtime); contiene funzioni base di programmazione come la gestione della memoria, stringhe e input/output.
- **ADVAPI32.dll** → (**Critica**) Gestisce funzionalità avanzate come il **Registro di Sistema**, i Servizi di Windows e gli account utente/sicurezza.
- **KERNEL32.dll** → (**Critica**) È il cuore del sistema operativo; gestisce la memoria, i file, i processi e i thread. Quasi ogni eseguibile dipende da questa.
- **GDI32.dll** → (Graphics Device Interface) Si occupa del disegno grafico di base: linee, curve, font e gestione dei colori sullo schermo.
- **USER32.dll** → Gestisce l'interfaccia utente di base: creazione e gestione delle finestre, mouse, tastiera e messaggi di sistema.

Ci concentreremo principalmente su **KERNEL32.dll**, per andare a visualizzare le funzioni contenute all'interno di questa specifica libreria ci sposteremo sulla sezione "**Import Directory**".



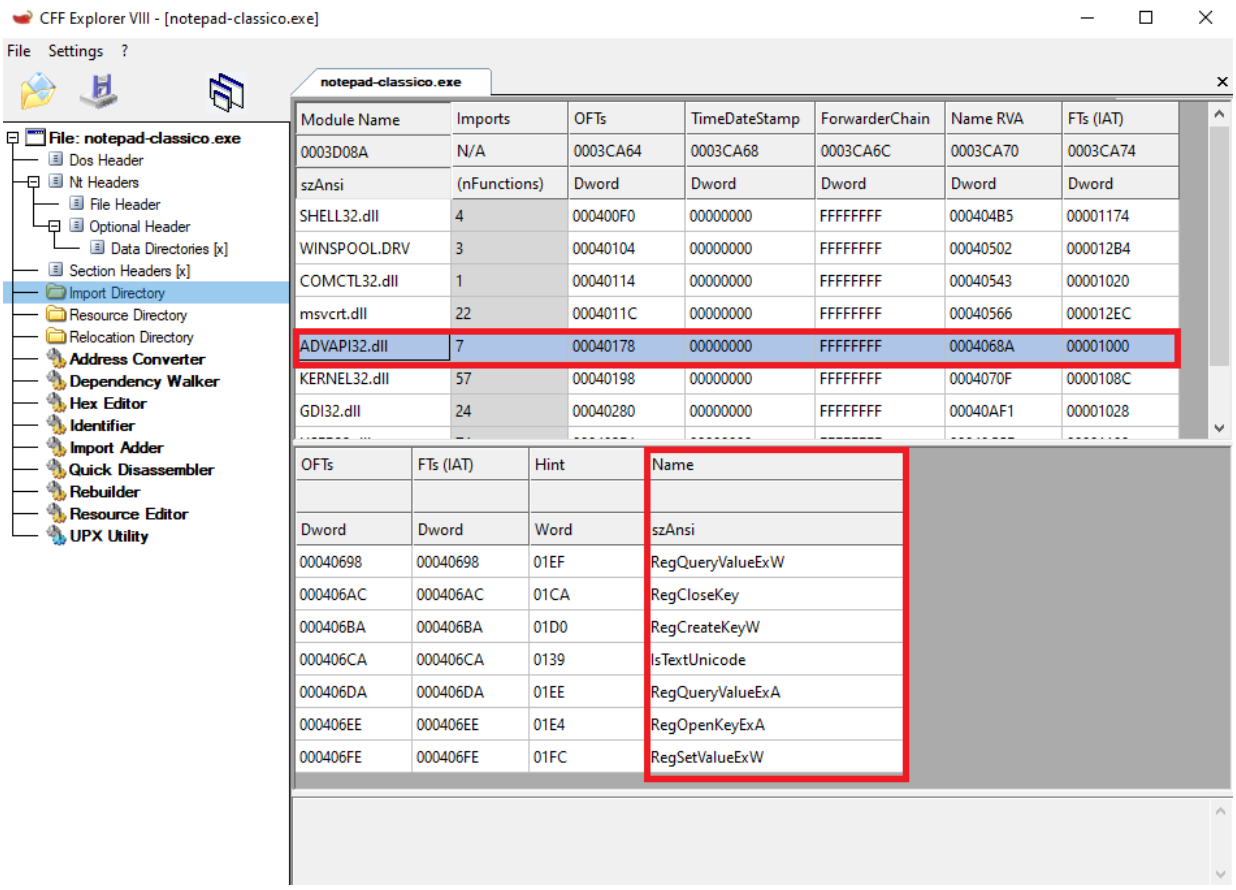
Le funzioni presenti nella libreria dimostrano alcuni comportamenti potenzialmente dannosi che il **malware** sarà in grado di avere.

Tra quelle visionate precedentemente notiamo:

- **CreateFileW/WriteFile**: Il malware può creare nuovi file o scrivere codice all'interno di file esistenti (magari per infettarli o per salvare dati rubati).
- **DeleteFileW**: Ha la capacità di cancellare file (comportamento tipico di un wiper o di un malware che vuole eliminare le proprie tracce).
- **FindFirstFileW**: Cerca file nel sistema (spesso usato per mappare il computer della vittima).

***Nota tecnica**:* La "W" finale sta per "**Wide**", indicando che la funzione supporta i caratteri **Unicode**.

Adesso procediamo ad esaminare la libreria “**ADVAPI32.dll** ” e le funzioni in esso incluse per vedere anche qui se esistono delle funzioni potenzialmente pericolose che il malware potrebbe utilizzare.



Le funzioni che iniziano con “**Reg**” fanno riferimento alle **Registry**, ovvero il Registro di sistema.

Il **Registro di Sistema** è un enorme database dove Windows salva le impostazioni.

Le funzioni particolarmente interessanti sono:

- **`RegCreateKeyW`**: Crea una nuova "cartella" (chiave) nel registro.
- **`RegSetValueExW`**: Scrive un valore dentro una chiave.

Perche risultano “Sospette”?

Spesso i malware usano RegSetValueExW per scrivere il proprio percorso in chiavi di avvio automatico. In questo modo, **ottengono la Persistenza**: ogni volta che accendi il computer, Windows legge quella chiave e avvia il virus automaticamente.

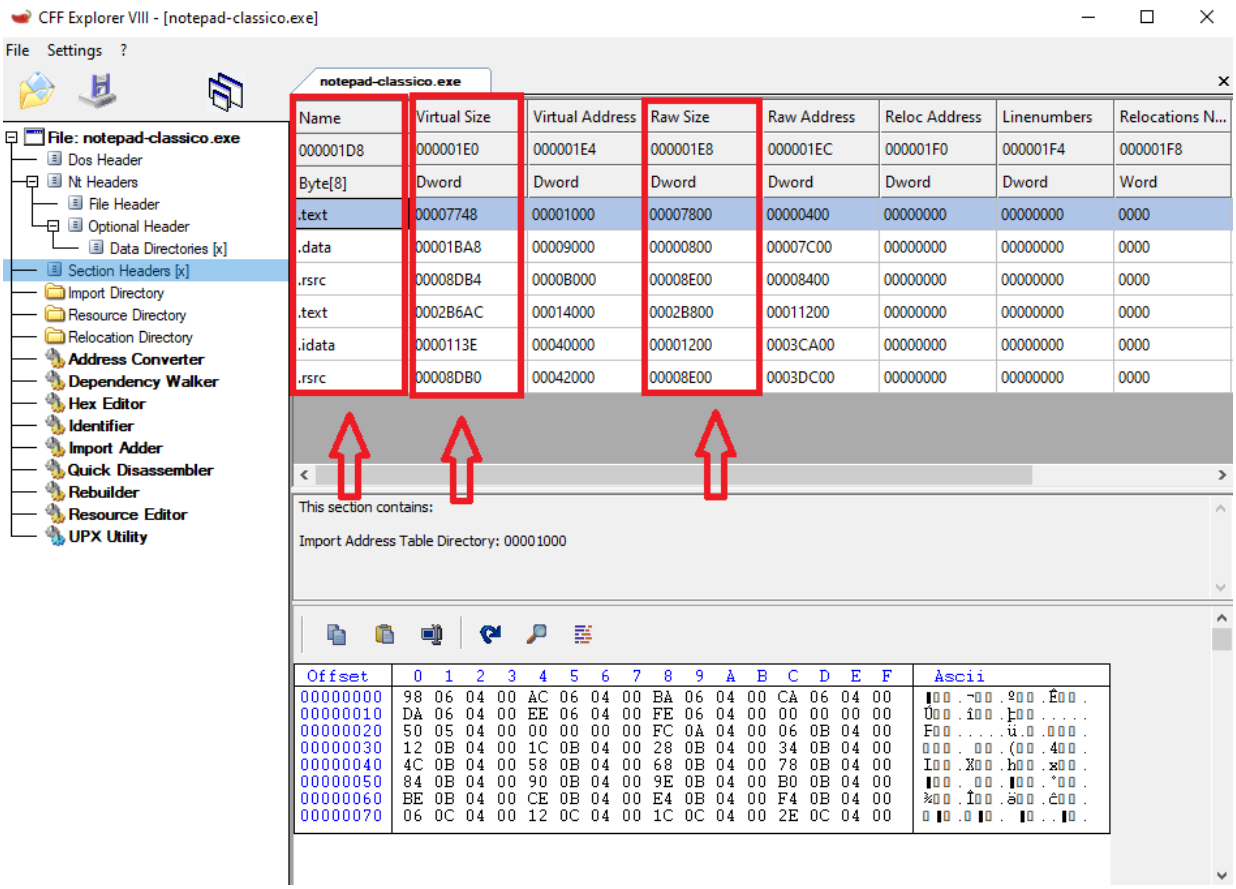
3. Analisi Statica - Sezioni del File

In questa fase continueremo ad effettuare un'analisi statica del malware al fine di esaminare nel dettaglio le sezioni di cui si compone, fornendo una descrizione per ognuna di esse.

Per questa ulteriore analisi utilizzeremo sempre CSS Explorer spostandoci però nel menu "**Section Headers**".

Qui prenderemo in esame le tre colonne riportate nell'immagine sottostante:

- **Name** → Identifica l'etichetta della sezione (come .text per il codice o .rsrc per le risorse), permettendo di riconoscere il tipo di contenuto ospitato.
- **Virtual Size** → Indica la dimensione effettiva che la sezione occuperà una volta caricata nella memoria volatile (RAM) del sistema.
- **Raw Size** → Indica la dimensione fisica che la sezione occupa all'interno del file salvato sul disco rigido.



Per prima cosa sono andato a confrontare le **Virtual Size** con le **Raw Size**.

Se il Virtual Size è molto più grande del Raw Size, significa che c'è del codice "**compresso**" che esplode quando il programma parte.

La sezione **`.data`** ha una Virtual Size (1BA8) che è un po' più grande della Raw Size (800), ma non è una differenza così enorme.

Esaminando però la colonna **Name** notiamo qualcosa di anomalo: **le sezioni tendono a ripetersi** (sono presenti due .txt o due .rsrc).

Questa ridondanza non è conforme agli standard dei compilatori legittimi. È un forte indicatore che l'eseguibile è stato manipolato post-compilazione. È altamente probabile che il file sia stato **"packato"** (compresso/offuscato) o infettato, e che le sezioni duplicate contengano il **payload malevolo** che verrà estratto o eseguito in memoria, separatamente dal codice originale del "Notepad".

4. Analisi Dinamica

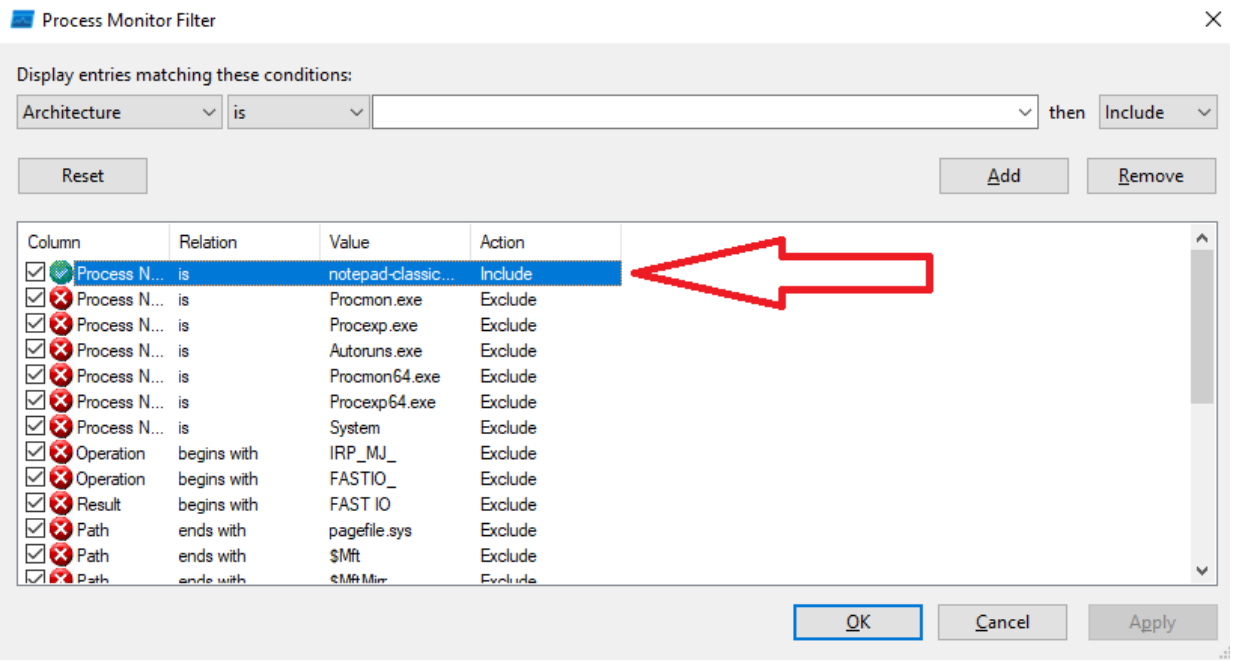
In questa fase passeremo da un analisi Statica ad una **Dinamica** ed osserveremo in tempo reale come il programma interagisce con il sistema operativo: quali file crea, quali chiavi di registro tocca e se prova a comunicare con l'esterno.

Dal momento che dovremo avviare il malware è importante assicurarci che la macchina **FlareVM** non sia connessa ad internet.

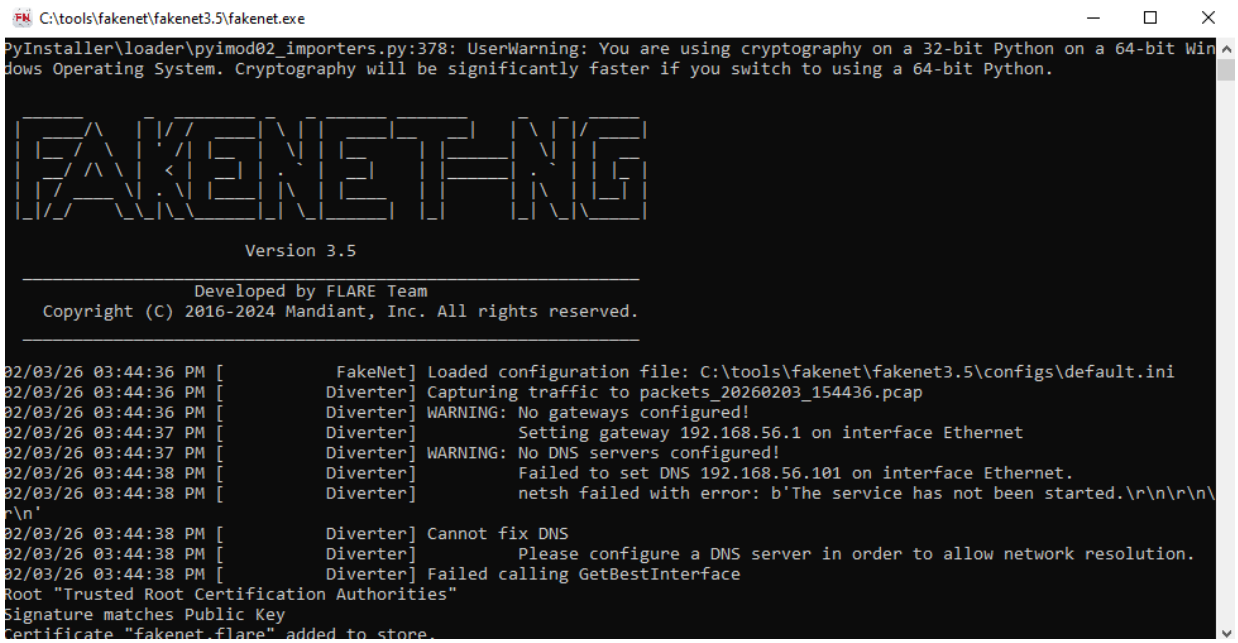
Per effettuare l'analisi utilizziamo i seguenti tool:

- **Process Monitor** che si occupa di registrare ogni singolo accesso ai file ed al registro di sistema.
- **FakeNet-NG** che si occupa di simulare internet.

Su Process Monitor assicuriamoci di applicare correttamente i filtri in modo da vedere solamente i Log inerenti al malware **notepad-classico.exe**.

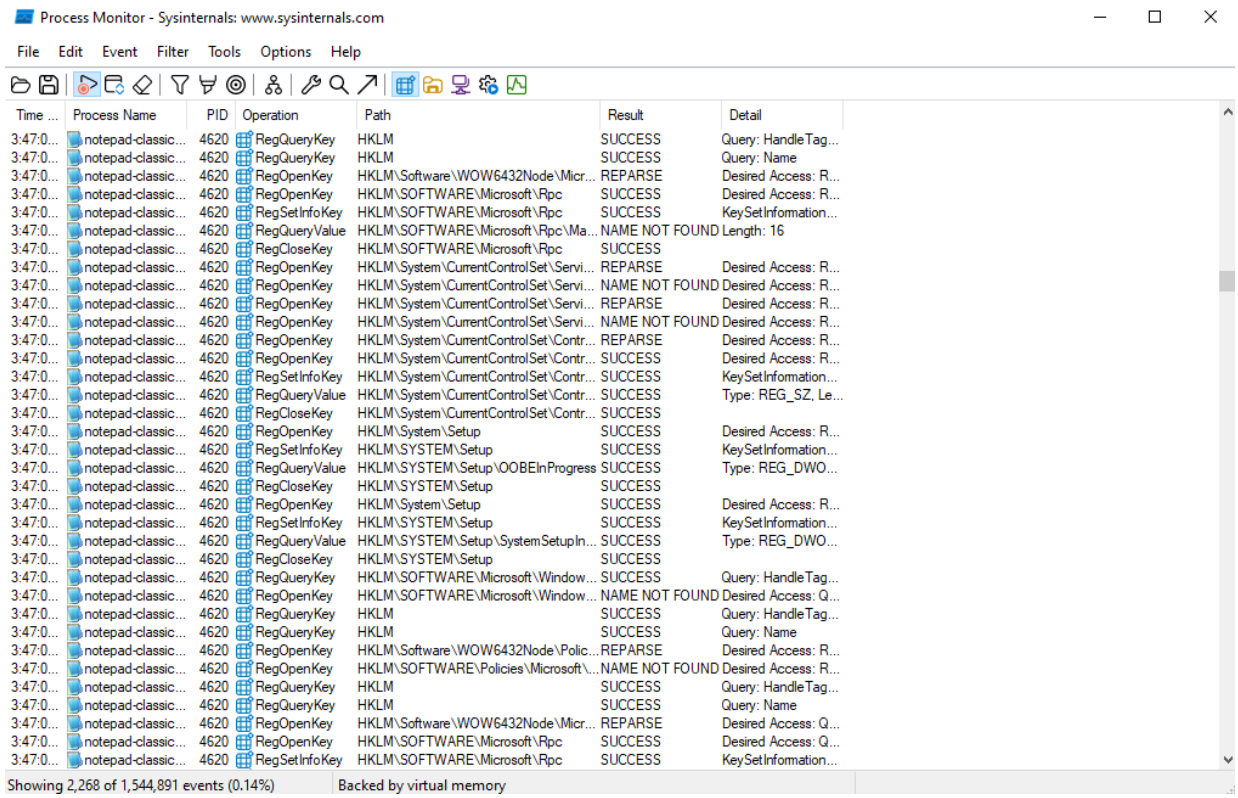


Avviamo **Fakenet** dal terminale tramite il comando **fakenet**.



4.1 Registry Activity

L'analisi con Process Monitor ha evidenziato tentativi di interazione con chiavi sensibili del Registro di Sistema.

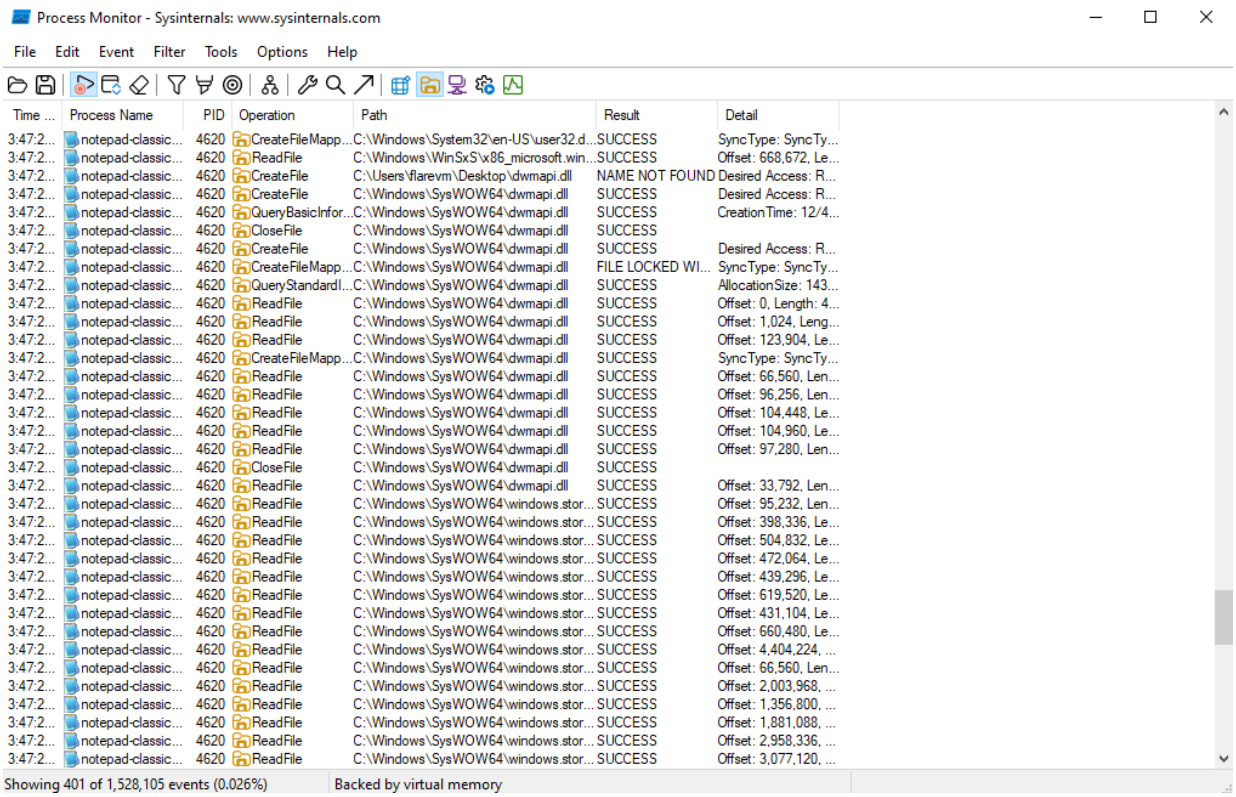


Comportamento osservato: Sono state rilevate operazioni di apertura e interrogazione (**`RegOpenKey`**, **`RegQueryKey`**) verso il ramo **HKLM** (HKEY_LOCAL_MACHINE), specificamente in percorsi come `HKLM\SYSTEM\Setup` e `HKLM\SOFTWARE\Microsoft\Rpc`.

Analisi: L'accesso al ramo HKLM suggerisce che il malware tenta di interagire con le configurazioni globali del sistema, piuttosto che limitarsi a quelle dell'utente corrente. Questo comportamento richiede solitamente privilegi elevati e può indicare tentativi di evasione, persistenza o modifica delle impostazioni di sicurezza.

4.2 File System Activity

L'analisi dei log di Process Monitor ha permesso di osservare come il malware interagisce con il file system durante la sua esecuzione.



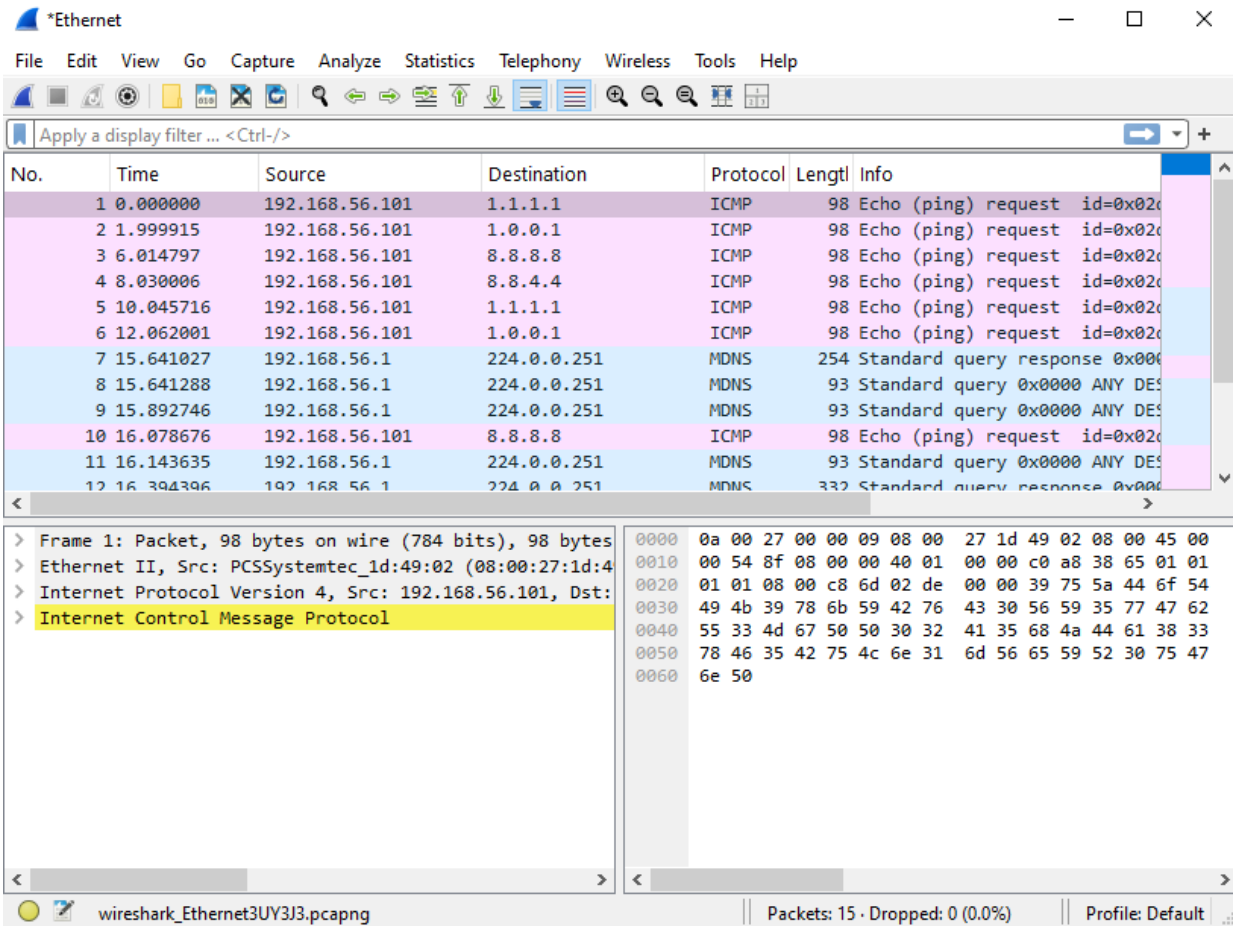
Comportamento osservato: Sono state rilevate numerose operazioni di lettura (`ReadFile`) e apertura (`CreateFile`) relative a librerie di sistema, come `dwmapi.dll` e `windows.storage.dll`.

Analisi: È stato notato un tentativo di caricamento della DLL `dwmapi.dll` direttamente dalla directory di esecuzione (`C:\Users\flarevm\Desktop\`),

terminato con esito `NAME NOT FOUND`, seguito dal caricamento corretto dalla cartella di sistema `SysWOW64`.

4.3 Network Activity

Durante l'esecuzione, il malware ha generato traffico di rete immediato, rilevato tramite Wireshark.



Comportamento osservato: Il processo `notepad-classico.exe` ha inviato richieste **ICMP (Ping)** verso indirizzi IP pubblici specifici:

- `8.8.8.8` (Google DNS)
- `1.1.1.1` (Cloudflare DNS)

Analisi: Questo comportamento non è coerente con un normale editor di testo. Si tratta verosimilmente di un **Connectivity Check**: il malware verifica se la macchina

infetta ha accesso a Internet attivo, probabilmente come pre-requisito per scaricare ulteriori payload o contattare un server di Comando e Controllo.

6. Conclusione e Considerazioni

Sulla base dell'analisi effettuata, il campione `notepad-classico.exe` è classificato come **malevolo**.

Nonostante il camuffamento da applicazione legittima (Notepad), il file presenta indicatori di compromissione (**IOC**) critici:

- **Offuscamento:** La presenza di sezioni duplicate (`.text`, `.rsrc`) indica l'uso di software di packing per nascondere il codice originale.
- **Persistenza:** I tentativi di scrittura nel Registro di Sistema (`HKLM`) dimostrano l'intenzione di radicarsi nel sistema operativo.
- **Comportamento di Rete:** Il connectivity check verso DNS pubblici (8.8.8.8, 1.1.1.1) all'avvio è tipico dei malware che necessitano di comunicare con l'esterno.

Le caratteristiche osservate suggeriscono che il malware agisca come un **Dropper** o un **Downloader**. Il suo obiettivo primario non sembra essere il danneggiamento diretto immediato, bensì garantire la persistenza e verificare la connessione internet, presumibilmente per scaricare ed eseguire ulteriori payload malevoli in una fase successiva.