Report S10-L1

Malware Analysis – Analisi Statica basica

Traccia:

Nella lezione teorica del mattino, abbiamo visto come recuperare informazioni su un malware tramite l'analisi statica basica.

Con riferimento al file eseguibile contenuto nella cartella «**Esercizio_Pratico_U3_W2_L1**» presente sul desktop della vostra macchina virtuale dedicata all'analisi dei malware, rispondere ai seguenti quesiti:

- Indicare le librerie importate dal malware, fornendo una descrizione per ognuna di esse.
- Indicare le sezioni di cui si compone il malware, fornendo una descrizione per ognuna di essa.
- Aggiungere una considerazione finale sul malware in analisi in base alle informazioni raccolte.

Sommario

Introduzione teorica

Nella prima settimana della Unit 3 si sono affrontati i concetti legati alle security operations, tra le quali rientrano le azioni preventive che le compagnie possono adottare per impedire il verificarsi di incidenti di sicurezza.

Malgrado l'implementazione delle azioni preventive, c'è la probabilità che un incidente di sicurezza possa verificarsi. Per questo motivo, le aziende mettono in pratica piani di Business Continuity e Disaster Recovery per assicurare la continuità dei servizi critici ai propri utenti.

L'incidente di sicurezza in sé viene gestito tramite le procedure di «incident response», dove il CSIRT, team dedicato, mette in campo le azioni per contrastare la propagazione dell'incidente.

Ad oggi, la maggior parte degli incidenti di sicurezza è dovuta ad attacchi dall'esterno e, in particolar modo, ad attacchi malware.

Malware e Malware Analysis

I **Malware** (malicious software) includono una vasta gamma di **programmi scritti per arrecare danno a sistemi informatici**, spesso a scopo di lucro.

L'analisi del malware o «**malware analysis**» è l'insieme di competenze e tecniche che permettono ad un analista della sicurezza informatica di indagare accuratamente un malware per studiare e capire esattamente il suo comportamento al fine di rimuoverlo dal sistema.
Queste competenze sono fondamentali per i membri tecnici del CSIRT durante la risposta agli incidenti di sicurezza.
Sono due le tecniche principali di analisi:

- **Analisi statica:** comporta l'esame del codice sorgente o del file eseguibile del malware senza eseguirlo.

- **Analisi dinamica**: presuppone l'esecuzione del malware in ambiente controllato, come una sandbox, per monitorare il suo comportamento in tempo reale.

Mentre l'analisi dinamica presuppone l'esecuzione del malware in ambiente controllato, l'analisi statica fornisce tecniche e strumenti per analizzare il comportamento di un software malevolo senza la necessità di eseguirlo.
Le due **tecniche** sono tra di loro **complementari**, per un'analisi efficace i risultati delle analisi statiche devono essere poi confermate dai risultati delle analisi dinamiche.

Entrambe le tecniche si dividono in «basica» e «avanzata», pertanto si hanno:

- **Analisi statica basica**
  L'analisi statica basica consiste nell'***esaminare un eseguibile senza vedere le istruzioni che lo compongono.***
  Lo **scopo** dell'analisi basica statica è di confermare se un dato file è malevolo e fornire informazioni generiche circa le sue funzionalità.

L'analisi statica basica è sicuramente la **più intuitiva** e **semplice** da mettere in pratica, **ma** risulta anche essere la **più inefficiente** soprattutto contro malware sofisticati.

- **Analisi dinamica basica**
  L'analisi dinamica basica presuppone l'***esecuzione del malware in modo tale da osservare il suo comportamento sul sistema infetto al fine di rimuovere l'infezione***.
  I malware devono essere eseguiti in ambiente sicuro e controllato in modo tale da eliminare ogni rischio di arrecare danno a sistemi o all'intera rete.
  Così come per l'analisi statica basica, l'analisi dinamica basica è piuttosto semplice da mettere in pratica, ma **non è molto efficace** quando ci si trova ad analizzare **malware sofisticati**.

- **Analisi statica avanzata**
  L'analisi statica avanzata presuppone la ***conoscenza dei fondamenti di «reverse-engineering»*** al fine di identificare il comportamento di un malware a partire dall'analisi delle istruzioni che lo compongono.
  In questa fase vengono utilizzati dei **tool** chiamati «**disassambler**» che ricevono in input un file eseguibile e restituiscono in output il linguaggio «Assembly».

- **Analisi dinamica avanzata**
  L'analisi dinamica avanzata presuppone la **conoscenza dei debugger** per esaminare lo stato di un programma durante l'esecuzione.

Lo studio e la comprensione del comportamento esatto di un malware è un compito piuttosto complicato.
Tuttavia, **può essere semplificato identificando il tipo di malware** che si sta analizzando.
Un esempio, è quello di un malware che si mette in ascolto su una porta TCP e garantisce una shell a chi si connette, il che fa presumere che si tratti di una backdoor.
Allo stesso modo, un malware che contatta un dominio per scaricare un altro file eseguibile, potrebbe essere un «downloader», ovvero un malware che scarica altri malware.
Capire preventivamente il tipo di malware da alcuni caratteri generali può, dunque, aiutare nella comprensione del comportamento generale.

Nelle esercitazione della Unit 3 del corso di Cyber Security, si utilizzerà la **macchina virtuale** «**Malware Analysis**», la quale, una volta scaricata, va semplicemente importata in Oracle VirtualBox (è sufficiente cliccare due volte sul file .ova della VM per determinarne l'importazione).

NB: La macchina **deve essere configurata in modalità di rete interna** e non deve avere connessioni con il mondo esterno, in quanto, essendo utilizzata come **macchina di test per lo studio dei malware**, contiene al suo interno eseguibili dannosi che, se non gestiti correttamente, creano danni importanti.

Analisi statica Basica

L'analisi statica basica consiste nell'***esaminare un eseguibile senza vedere le istruzioni che lo compongono.***
Lo **scopo** dell'analisi basica statica è di **confermare se un dato file è malevolo** e fornire informazioni generiche circa le sue funzionalità.

Quando si analizza un potenziale malware il prima passo da fare è **assicurarsi che sia di fatto un malware**, attraverso l'analisi delle firme dei malware.
Nello studio della Unit 2 si è visto che ogni file ha una propria firma (**file signature**), di conseguenza anche i malware hanno una propria firma.
Per capire se quello che si analizza sia effettivamente un malware, si effettua la comparazione della firma del file sospetto con quelle contenute nei database degli antivirus, per vedere se risulta corrispondenza con la firma nota di un particolare malware.

Siti per identificazione dei malware

Siti come **VirusTotal** (https://www.virustotal.com/gui/home/upload), permettono di caricare un file eseguibile e controllare se si tratti di un malware, in base ad un numero variabile ma consistente di software antivirus.

Per utilizzare VirusTotal, una volta connessi al link cliccare su «choose file» e caricare il file che si vuole controllare.
VirusTotal calcolerà per prima cosa la firma del software e poi controllerà se la stessa è già presente nei database dei software antivirus, e la sua eventuale categorizzazione come «malware».

Calcolo hash di un malware

Alternativamente si può calcolare «l'hash» di un malware, ovvero una ***stringa alfanumerica unica per identificare un file.***

Gli hash sono **valori crittografici univoci** generati da una funzione matematica di hash che prende in input dati, o nel caso dei malware, file di qualsiasi dimensione e restituisce una **stringa alfanumerica di lunghezza fissata**.

L’hash, detto anche Checksum, è un “l’**impronta digitale**” univoca che identifica in modo univoco ogni file.
Vista l’univocità dell’hash di ogni file, è quasi impossibile trovare due file diversi che abbiano lo stesso HASH, così come è anche impossibile modificare un file senza che si modifichi il suo HASH.

Negli ambienti di sicurezza informatica, l'hash viene spesso utilizzato per identificare e verificare l'integrità dei file.
Quando si tratta di malware, l'hash di un file malware è essenzialmente una **firma univoca** basata sulla sua struttura e sul contenuto.
Gli analisti di sicurezza generano spesso l'hash di un file sospetto o noto malware e lo condividono in database di firma o listati di hash con altri operatori di sicurezza.

L’hash consente una rapida identificazione del file associato, la verifica dell’integrità dei dati in esso contenuti, garantendo che non siano stati modificati e compromessi, ma soprattutto consente il rilevamento tempestivo dei malware noti.

Md5deep

L’utility «md5deep», che nella macchina virtuale è nella cartella «md5deep-4.3» del desktop**, può essere utilizzato per calcolare l’hash di un file.**

Trattandosi di un tool a riga di comando, per il suo utilizzo si deve seguire una serie di step:
- Aprire il «command prompt» (presente sul desktop)
- Spostarsi con il comando «cd» nella cartella contenente l’eseguibile da utilizzare (dir=ls)
- Eseguire il comando seguito dai suoi parametri: una volta che si è nella stessa cartella del file eseguibile, si può lanciare il comando, la cui sintassi è ‘md5deep «percorso del file del quale si vuole calcolare l’hash»’.

Il risultato del comando è una stringa alfanumerica seguita dal path del file stesso, per il quale è stato calcolato l’hash.

Una volta recuperato l'hash di un file, è possibile **utilizzarlo come un'etichetta** per l'identificazione univoca del file stesso.
La **condivisione** di questo identificativo **con altri analisti di sicurezza** agevola il riconoscimento di eventuali malware.
Inoltre, la ricerca online, sfruttando l’hash, consente la conferma o meno dell’identità di un file come malware e di ottenere informazioni sul comportamento dello stesso.

Utility strings

Gli eseguibili contengono molto spesso delle **stringhe** al loro interno, ad esempio utilizzate per scrivere a schermo un messaggio di benvenuto, o dettagliare l’utilizzo di un software o per connettersi ad un dato URL online.
Nel caso di software dannosi, si potrebbero recuperare importanti informazioni dalle stringhe contenute all’interno degli eseguibili.
A tal proposito, l’utility da riga di comando «strings*» **permette di trovare tutte le stringhe utilizzate all’interno di un file eseguibile.***

È da notare che nella macchina di laboratorio fornita, Malware Analysis, l'utility «strings» è all'interno della **cartella «Sysinternals Suite»** del desktop.
Essendo uno strumento da riga di comando, valgono le regole viste in precedenza.
La sintassi del comando è: **strings «file eseguibile».**

L'**output** dell'utility consiste in una lista di stringhe, in cui alcune possono dare indizi sul comportamento del malware, mentre altre sono totalmente da ignorare in quanto prive di significato.
È ruolo dell'analista di sicurezza controllare per bene e dettagliatamente tutte le stringhe per trovare quelle più significative.
Un esempio di una lista di stringhe è la seguente:

- Xro
- Download from malware.com
- 5rrgf
- 39hhfj
- Xd
- Connect to malware.com
- 55.3.22.154

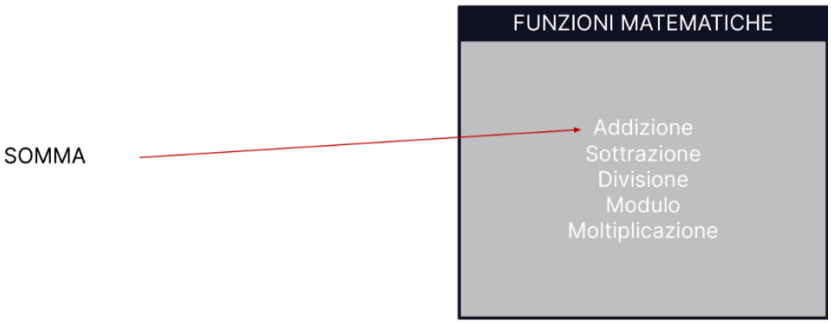
In un caso simile, si può ipotizzare che il malware che si sta analizzando si connetta ad un sito malware.com per scaricare altri malware, e che malware.com sia all'indirizzo IP 55.3.22.154.
Tutto ciò è, ovviamente, una conseguenza logica della fase di analisi statica basica, ma potrebbe rivelarsi un'assunzione sbagliata una volta che si esegue il malware nelle fasi di analisi dinamica.

Definizione dei file eseguibili

Windows utilizza per la maggior parte dei file eseguibili il **formato PE**, **Portable Executable**.
Il formato PE al suo interno contiene delle ***informazioni*** necessarie al sistema operativo ***per capire come gestire il codice del file, come ad esempio le librerie.***

Le **librerie** (anche chiamate moduli) sono insieme di funzioni predefinite pronte per l'uso, senza doverle riscrivere ogni volta.
Quando un programma ha bisogno di una funzione «chiama» una libreria al cui interno è definita la funzione necessaria.
Si dice anche che il programma ha **importato una libreria** (basti ricordare la sintassi iniziale dei programmi Python: import «modulo»).
Di seguito si riporta un esempio pratico.

Si ipotizza di avere un programma eseguibile somma che per funzionare deve importare una libreria «funzioni matematiche» che include, tra le altre, la funzione addizione.

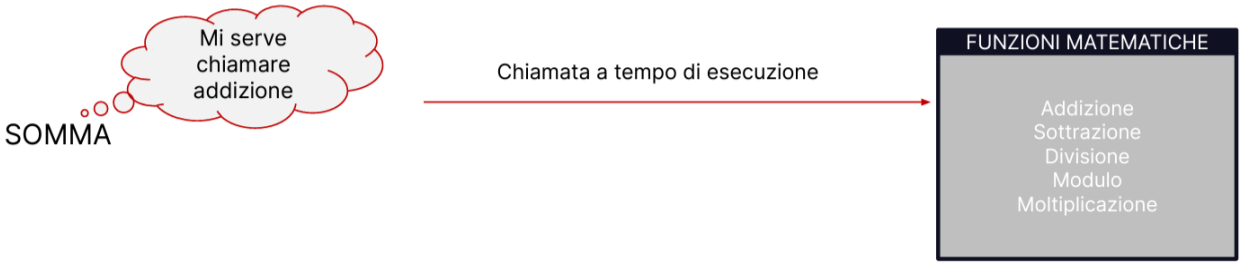


Modalità di importazione delle librerie

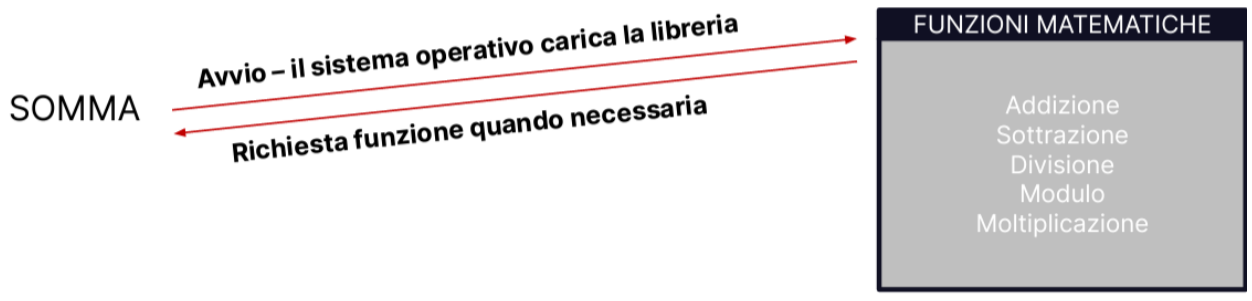
Le librerie e le funzioni possono essere importate in tre modi diversi **dagli eseguibili**:

- **Importazione statica**: quando l'eseguibile non fa altro che ***copiare tutto il contenuto della libreria*** all'interno del proprio codice.
  Di fatto, considerando l'esempio precedente, è come se SOMMA copiasse interamente il contenuto della libreria all'interno del proprio codice, comprese le definizioni delle funzioni sottrazione, divisione, modulo anche se non servono.
  Da un punto di vista pratico, questo approccio incrementa la dimensione di un file.
  Dal punto di vista dell'analista, risulta più complicato, invece, distinguere il codice della libreria dal codice in fase di analisi statica avanzata.

- **Importazione a tempo di esecuzione (Runtime):** in questa casistica ***l'eseguibile richiama la libreria solamente quando necessità di una particolare funzione***.
  Questo comportamento è ampiamente utilizzato dai malware, che importano una determinata funzione solo all'occorrenza, per risultare quanto meno invasivi e rilevabili possibile.
  Per importare la libreria all'occorrenza si utilizzano delle funzioni messe a disposizione dal sistema operativo come «**LoadLibrary**» e «**GetProcAddress**».



- **Importazione dinamica**: questa è la casistica più interessante per gli analisti di sicurezza ed anche la **più comune**.
  Le librerie importate dinamicamente vengono caricate dal sistema operativo quando l'eseguibile è avviato. Quando necessario, la funzione viene chiamata ed eseguita all'interno della libreria.



Librerie

Una competenza fondamentale per l'analisi dei malware è la conoscenza delle librerie e degli scopi per i quali vengono importate nel codice da parte dei file malware, formato PE.
Le librerie più comuni sono:

- **Kernel32.dll**: è una libreria piuttosto comune che contiene le funzioni principali per interagire con il sistema operativo, ad esempio: manipolazione dei file e gestione della memoria.
- **Advapi32.dll**: è una libreria che contiene le funzioni per interagire con i servizi ed i registri del sistema operativo Microsoft.
- **WSock32.dll** e **Ws2_32.dll**: sono librerie che contengono le funzioni di network, come le socket, le funzioni connect, bind. Ogni malware che utilizza funzionalità di rete caricherà certamente queste librerie.
- **Wininet.dll**: libreria che contiene le funzioni per l'implementazione di alcuni protocolli di rete come HTTP, FTP, NTP.
- **Gdi32.dll**: libreria che contiene le funzioni per l'implementazione e manipolazione della grafica e dei suoi effetti
- **MSVCRT.dll**: libreria che contiene funzioni per la manipolazione stringhe, allocazione memoria e altro, come chiamate per input/output in stile linguaggio C.

Oltre a quelle citate, ci sono molte altre librerie utilizzate dai malware. La pratica e l'esperienza diretta sono di grande supporto nella comprensione delle librerie e delle funzioni più utilizzate dai malware.
È fondamentale sapere che le informazioni circa le librerie e le funzioni richieste dall'eseguibile sono contenute nell'**header del formato PE** (Portable Executable).
Infatti, controllando l'header del PE, è **possibile sapere quali librerie e funzioni sono importate**, il che è fondamentale **per capire lo scopo dei malware**.

Esportazione e importazione di funzioni
Oltre alle funzioni importate, un file eseguibile può ***esportare funzioni***. Ovvero, può mettere a disposizione di altri programmi o dell'utente delle funzioni da «chiamare».

Quindi l'header del formato PE contiene:

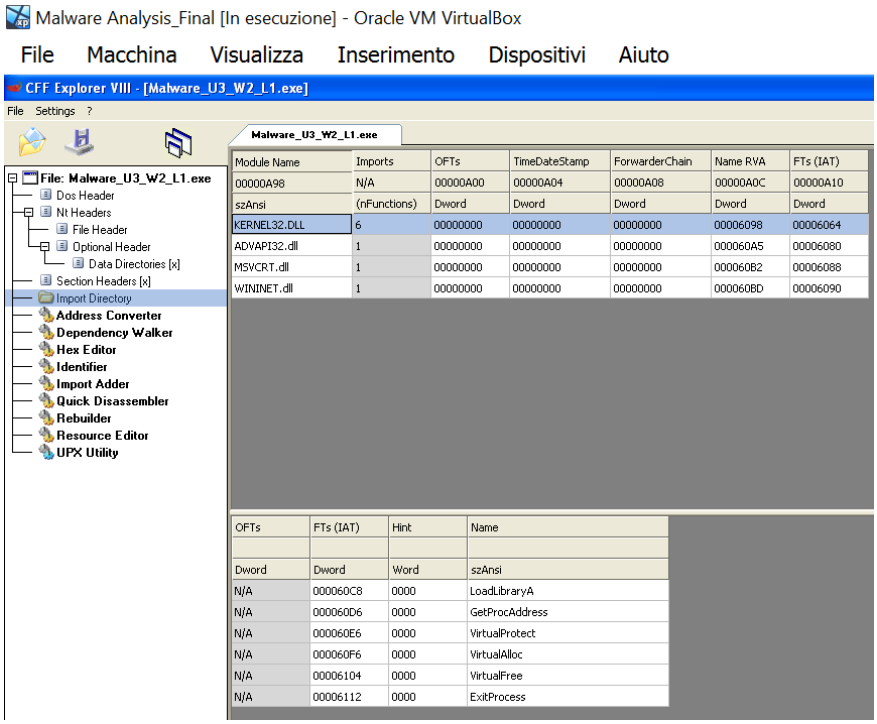
- sia l'elenco delle **funzioni importate**
- che l'elenco delle funzioni **esportate** dall'eseguibile.

CFF Explorer

Si tratta di un software utilizzato per la ***visualizzazione, l'analisi e la modifica della struttura e del contenuto interno dei file eseguibili***, quali i file formato PE.
In particolare, CFF Explorer consente di ***individuare*** le librerie importate dall'eseguibile, e ***le funzioni importate ed esportate da un malware***.
Questo **tool** è installato sulle VMs (Virtual Machines) dedicate all'analisi dei malware, come quella fornita, in cui è presente sul desktop.

Una volta avviato il tool, con la scelta dell'eseguibile del quale si vuole esaminare l'header del formato PE, si aprirà una schermata con un menù sulla sinistra che comprende varie opzioni. Ci si deve spostare sull'opzione «import directory» per controllare le librerie e le funzioni importate.

A quel punto, nella schermata, si vedranno le informazioni sulle librerie importate mentre, per ognuna delle librerie, un pannello inferiore mostrerà una lista delle funzioni richieste all'interno della libreria selezionata.



Sezioni

L'header del formato PE fornisce, oltre alle funzioni/librerie importate ed esportate, molte altre informazioni importanti, quali le sezioni di cui si compone il software.

Ogni sezione ha un preciso scopo, e conoscerle è una preziosa informazione per le analisi.
Le più comuni ed interessanti sezioni in un file PE sono:

- **.text**: la sezione «text» contiene le istruzioni (le righe di codice) che la CPU eseguirà una volta che il software sarà avviato. Generalmente questa è l'unica sezione di un file eseguibile che viene eseguita dalla CPU, in quanto tutte le altre sezioni contengono dati o informazioni a supporto.

- **.rdata**: la sezione «rdata» include generalmente le informazioni circa le librerie e le funzioni importate ed esportate dall'eseguibile, informazione che si possono ricavare con CFF Explorer.

- **.data**: la sezione «data» contiene tipicamente i dati/le variabili globali del programma eseguibile, che devono essere disponibili da qualsiasi parte del programma.
  Una variabile si dice globale quando non è definita all'interno di un contesto di una funzione, ma è globalmente dichiarata ed è, di conseguenza, accessibile da qualsiasi funzione dell'eseguibile.

- **.rsrc**: la sezione «rsrc» include le risorse utilizzate dall'eseguibile come ad esempio icone, immagini, menu e stringhe che non sono parte dell'eseguibile stesso.

Per controllare le sezioni di un file eseguibile, si avvia CFF Explorer, si seleziona il file eseguibile del quale si vogliono controllare le sezioni e ci si sposta nel pannello a sinistra, nella sezione «section headers».
Il pannello principale a destra mostrerà le informazioni circa le sezioni di cui si compone l'eseguibile.
Tra queste informazioni, altre fondamentali sono:

- **Virtual size**: indica lo spazio allocato per la sezione durante il processo di caricamento dell'eseguibile in memoria.
- **Raw size**: indica lo spazio occupato dalla sezione quando è sul disco.

Un tool alternativo per la visualizzazione delle librerie, funzioni e sezioni è **ExeinfoPE.**

Si è appena visto come, utilizzando un set di tool piuttosto semplici, si possono recuperare importanti informazioni circa un eseguibile malware, ed iniziare ad avere un'idea sul suo comportamento.

Tuttavia, l'analisi statica è un processo piuttosto semplice ma altresì inefficace contro i tipi di malware più avanzati che tendono, per esempio, ad oscurare il nome delle sezioni, oppure a configurare le sezioni con un nome senza un senso logico.
Alcuni malware utilizzano il caricamento delle librerie durante l'esecuzione (runtime import), nascondendo di fatto all'analisi statica le funzioni e le librerie importate.
Questi malware sono riconoscibili in quanto hanno generalmente poche entry nella sezione import, e tra esse figurano le funzioni «LoadLibrary e GetProcAddress» che vengono utilizzate per caricare funzioni addizionali durante l'esecuzione.

Svolgimento esercitazione

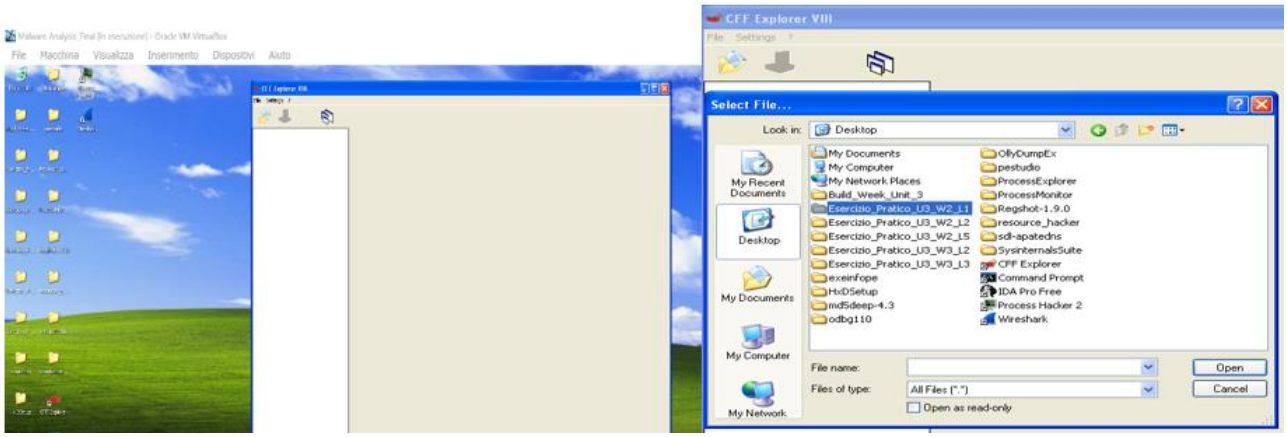
La richiesta della traccia è di recuperare informazioni su un file eseguibile tramite l'analisi statica basica.
In particolare, in relazione al file eseguibile contenuto nella cartella «Esercizio_Pratico_U3_W2_L1» presente sul desktop della macchina virtuale dedicata all'analisi dei malware, la traccia chiedeva di: indicare le librerie importate dal malware, fornendo una descrizione per ognuna di esse, Indicare le sezioni di cui si compone il malware, fornendo una descrizione per ognuna di essa e infine di aggiungere una considerazione finale sul malware in analisi in base alle informazioni raccolte.

Si è visto nell'introduzione che il tool CFF Explorer, di analisi dei malware, consente di ***individuare*** le ***librerie*** e ***le funzioni, importate ed esportate da un malware, e le sezioni che lo compongono***.

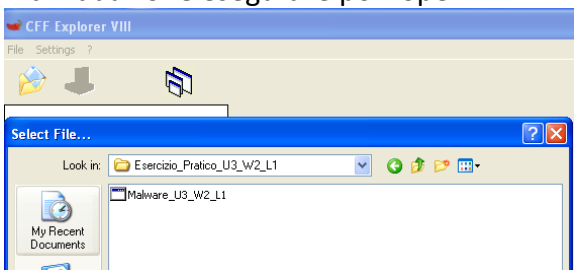
Di seguito si riportano i passaggi che sono stati eseguiti sul tool per ottenere le informazioni sulle librerie importate dal malware e sulle sezioni che lo compongono.

Una volta cliccato sull'icona rossa del tool, nella schermata che si apre, si ricerca il file eseguibile di cui si vuole analizzare il contenuto: nel caso preso in esame nell'esercitazione odierna, l'eseguibile è rappresentato dal file "Malware_U3_W2_L1", contenuto nella cartella del desktop "Esercizio_Pratico_U3_W2_L1".

Schermata iniziale e ricerca cartella del desktop

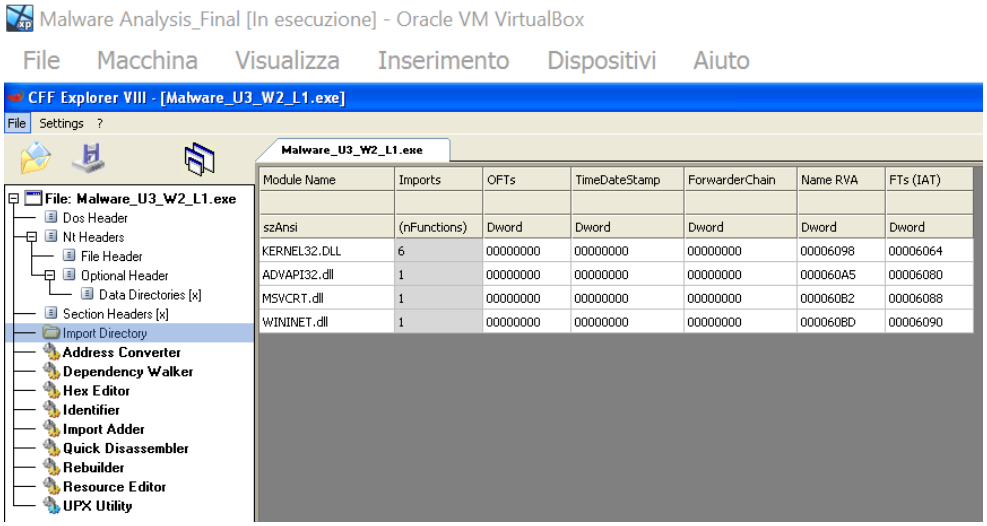


Individuazione eseguibile poi "open"



Librerie importate dall'eseguibile

Schermata iniziale CFF Eplorer relativa al file eseguibile

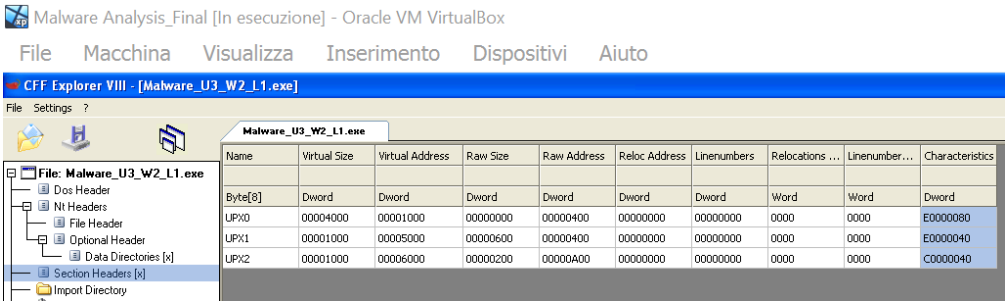


Nella schermata principale, a questo punto, si possono vedere le librerie importate dall'eseguibile, che sono:

- **KERNEL32.dll**: è una delle principali librerie di sistema di Windows che gestisce funzioni di basso livello, come la gestione della memoria, operazioni di input/output e creazione di processi.
  In sostanza, viene utilizzata per le funzioni di base del sistema operativo.
  ADVAPI32.dll: Fornisce funzionalità relative alla sicurezza, al registro di sistema e ai servizi.
  Questa libreria è spesso utilizzata per eseguire operazioni che richiedono privilegi elevati, come la modifica delle impostazioni di sicurezza o l'accesso a parti sensibili del registro di sistema.
  MSVCRT.dll: Il Microsoft Visual C++ Run-Time, è una libreria che fornisce funzioni standard di C come manipolazione di stringhe, matematica e altre operazioni di base.
  Viene utilizzata per accedere a funzioni standardizzate senza doverle scrivere da zero.
  WININET.dll: è una libreria che fornisce funzioni di alto livello per l'accesso a protocolli Internet come HTTP e FTP. WININET è spesso utilizzata per la comunicazione di rete, inclusa la possibilità di scaricare o caricare dati da e verso Internet.

Quindi, queste librerie sono progettate per l'utilizzo in ambiente Windows legittimo, ma, identificate in un eseguibile, potrebbero indicare le potenziali capacità del malware, come:
- Modificare o leggere dati sensibili dal sistema.
- Interagire con il sistema operativo a livello di rete o di interfaccia utente.
- Manipolare servizi e processi.

Sezioni dell’eseguibile



Per visualizzare le sezioni del file eseguibile ci si sposta nella “section header” del menù a sinistra dell’interfaccia principale.

Ciò che appare evidente, è che invece delle classiche sezioni .text, .rdata e .data, sono presenti le sezioni UPX0, UPX1, UPX2.
Questo significa che il file eseguibile è stato compresso utilizzando il pacchetto di compressione UPX (Ultimate Packer for eXecutables).
UPX è uno strumento di compressione per eseguibili che riduce le dimensioni del file compresso, consentendo un trasferimento più efficiente e una distribuzione più veloce.
Durante il processo di compressione, UPX crea nuove sezioni nel file eseguibile per contenere i dati dell’eseguibile compressi e le informazioni necessarie per decomprimere e ripristinare il programma originale, una volta eseguito.

Considerazioni sul malware in analisi

Analizzando le librerie importate, come KERNEL32.dll, ADVAPI32.dll, MSVCRT.dll e WININET.dll, si nota che il file eseguibile sembra sfruttare funzionalità di basso e alto livello del sistema operativo Windows.
In particolare, KERNEL32.dll e ADVAPI32.dll indicano la possibilità che il file esegua operazioni avanzate a livello di sistema, manipolando processi, accedendo a risorse sensibili e ottenendo privilegi elevati.
La presenza di WININET.dll suggerisce, inoltre, la possibilità che il comunichi o scarichi ulteriori payload da Internet, ampliando le sue capacità.

La presenza delle sezioni UPX0, UPX1 e UPX2 suggerisce che il file è stato compresso tramite UPX, una pratica spesso adottata dai malware per nascondere la propria presenza e complicare l'analisi statica.
Questa compressione, pur essendo pensata per ridurre le dimensioni del file, indica una potenziale sofisticatezza del malware.

In conclusione, la combinazione di librerie di sistema, la compressione tramite UPX e l'impiego di funzionalità avanzate suggerisce che il file eseguibile potrebbe essere un malware sofisticato.
Le potenziali azioni includono la manipolazione del sistema, l'interazione con la rete e la modifica delle impostazioni di sicurezza.
L'analisi dinamica del comportamento del malware sarà essenziale per comprendere appieno le minacce potenziali e implementare strategie di mitigazione efficaci.