09/02/2024

S10/L5 PROJECT

EPICODE MALWARE ANALYSIS MODULE

Luca Manna

Sommario

INTRODUZIONE

Traccia:

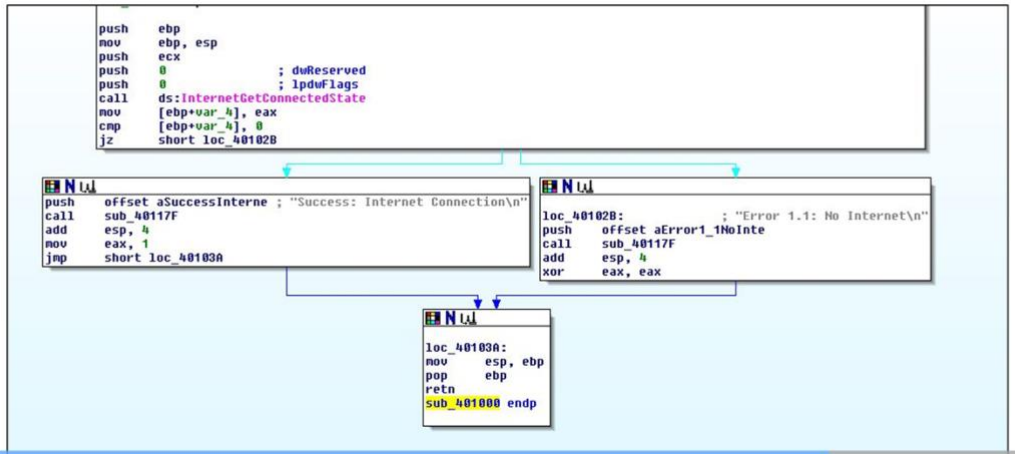
Con riferimento al file Malware_U3_W2_L5 presente all'interno della cartella «Esercizio_Pratico_U3_W2_L5» sul desktop della macchina virtuale dedicata per l'analisi dei malware, rispondere ai seguenti quesiti:

- Quali librerie vengono importate dal file eseguibile?
- Quali sono le sezioni di cui si compone il file eseguibile del malware?

Con riferimento alla figura in slide 3, risponde ai seguenti quesiti:

EPICODE **Esercizio**
Traccia e requisiti

Figura 1



- Identificare i costrutti noti (creazione dello stack, eventuali cicli, costrutti)
- Ipotizzare il comportamento della funzionalità implementata.

Breve Recap Settimanale:
Durante questa settimana abbiamo approfondito diversi aspetti legati all'analisi dei malware. Abbiamo studiato le tecniche di analisi statica e dinamica, così come i tool e le metodologie utilizzate per identificare e comprendere il comportamento dei malware. Inoltre, abbiamo esaminato i concetti fondamentali di sicurezza informatica e le strategie di difesa contro le minacce informatiche. In particolare, abbiamo dedicato del tempo allo studio del linguaggio assembly e al suo ruolo nell'analisi dei malware.

Definizione di Malware e Potenziali Rischi:
Un malware, abbreviazione di "malicious software" appunto software dannoso, è un termine generico che si riferisce a qualsiasi tipo di software progettato per danneggiare, alterare o compromettere un sistema informatico, una rete o i dati al suo interno, senza il consenso dell'utente. I malware possono assumere diverse forme, tra cui virus, worm, trojan, ransomware, spyware e molti altri.

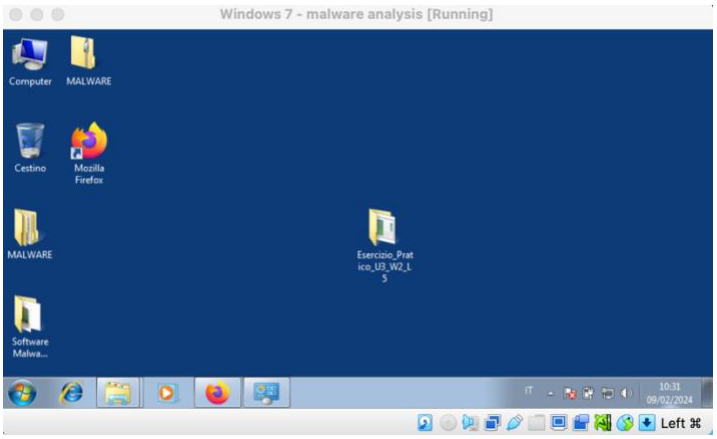
I potenziali rischi associati ai malware sono ampi e diversificati. Alcuni dei rischi principali includono:
Perdita di dati sensibili: I malware possono accedere, modificare o cancellare dati preziosi e sensibili, come informazioni personali, dati aziendali o finanziari, causando gravi danni alla privacy e alla sicurezza.
Danni al sistema: Alcuni malware sono progettati per danneggiare il sistema operativo o i file di sistema, causando malfunzionamenti del computer, crash o addirittura la perdita completa di funzionalità.
Compromissione della sicurezza: I malware possono aprire backdoor nel sistema, consentendo agli hacker di accedere e controllare il computer o la rete a distanza, compromettendo così la sicurezza dell'intera infrastruttura informatica.
Estorsione finanziaria: I ransomware sono malware progettati per crittografare i file dell'utente e richiedere un riscatto in cambio della loro decrittazione. Questo tipo di malware può causare danni finanziari significativi e interruzioni delle attività.
Violazione della privacy: Alcuni malware, come gli spyware, sono progettati per monitorare le attività degli utenti, raccogliendo informazioni

personali senza il loro consenso. Questo può portare a violazioni della privacy e all'abuso delle informazioni raccolte.

Ruolo del Linguaggio Assembly nell'Analisi dei Malware:
Il linguaggio assembly svolge un ruolo cruciale nell'analisi dei malware. Poiché i malware spesso operano a un livello molto basso del sistema operativo, l'analisi del loro comportamento richiede la comprensione delle istruzioni a livello di linguaggio macchina. Il linguaggio assembly fornisce un'interfaccia per comunicare direttamente con l'hardware del computer, consentendo agli analisti di comprendere in dettaglio le azioni intraprese dal malware e le tecniche utilizzate per evadere le misure di sicurezza.
Gli analisti di malware utilizzano spesso il linguaggio assembly per esaminare il codice del malware, identificare le sue funzionalità, individuare eventuali vulnerabilità e sviluppare contro misure di difesa.
Inoltre, l'analisi del linguaggio assembly consente di comprendere meglio le tecniche di evasione e di attacco impiegate dai malware, contribuendo così a migliorare le strategie di protezione e prevenzione.
In conclusione, il linguaggio assembly è uno strumento essenziale nell'analisi dei malware, fornendo agli analisti la capacità di scrutare il codice del malware a un livello molto dettagliato e comprendere le sue funzionalità e il suo comportamento. Una competenza solida nel linguaggio assembly è pertanto fondamentale per gli analisti di sicurezza informatica impegnati nella lotta contro le minacce informatiche.

ANALISI STATICA DI BASE

Come primo step di questo progetto inizieremo con l'Analisi statica di base del malware contenuto nella cartella U3 W2 L5 presente sulla nostra macchina virtuale Windows 7 adattata alla simulazione ed esecuzione di malware.

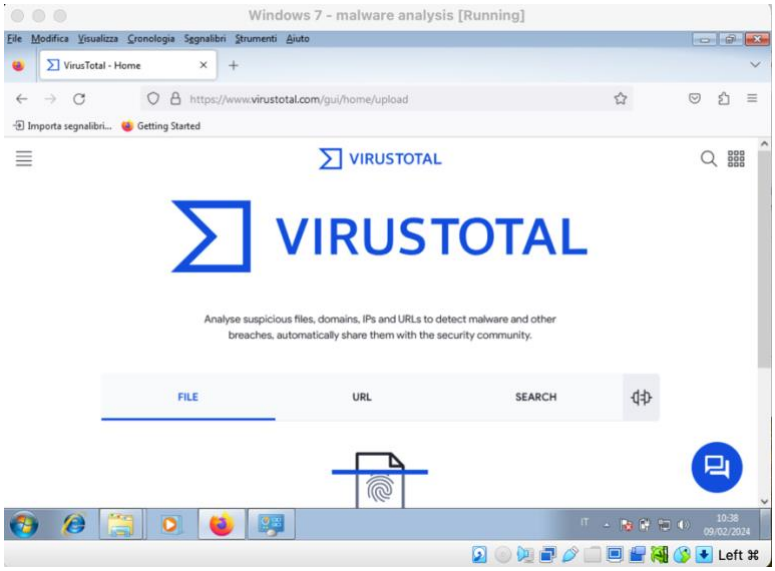


Ancor prima di svolgere l'esercizio chiariamo i concetti di Analisi statica di base e i tool necessari per effettuarla.

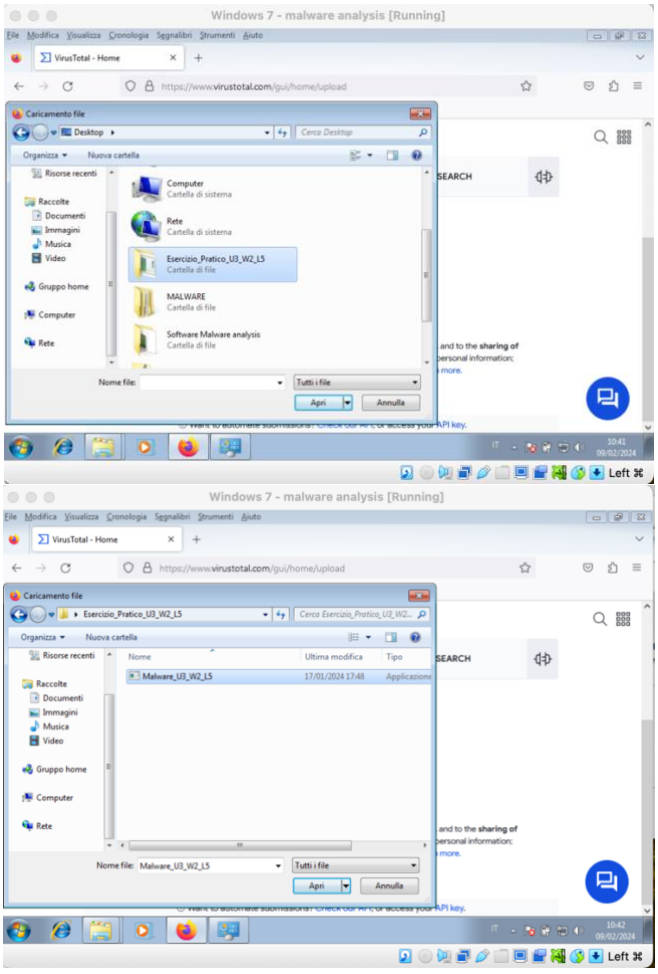
Un'analisi statica di base è un metodo di analisi dei malware che si concentra sull'esame del file binario senza eseguirlo. Durante questa analisi, vengono esaminate le caratteristiche del file, come le firme digitali, le intestazioni PE (Portable Executable), le stringhe di testo, i metadati e altre informazioni, al fine di identificare eventuali comportamenti sospetti o indicatori di compromissione.

VirusTotal è un servizio online gratuito che consente di analizzare file e URL sospetti utilizzando più di 70 motori antivirus e tool di analisi malware. Questo strumento può essere utilizzato per eseguire rapidamente un'analisi statica di base su un file, confrontando i risultati di diversi motori antivirus e rilevando potenziali minacce.

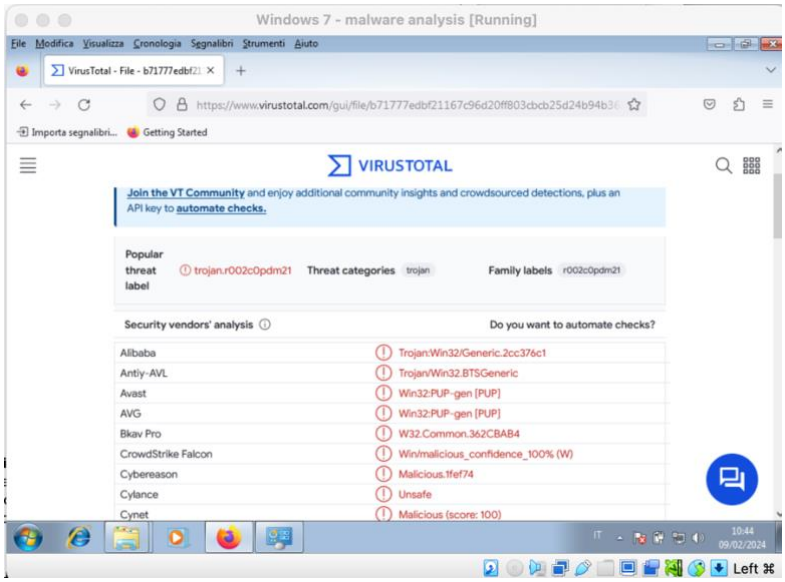
Passaggi per eseguire un'analisi statica di base con VirusTotal:
Accedi al sito web di VirusTotal: Apri il browser web sulla tua macchina virtuale e vai al sito web di VirusTotal all'indirizzo https://www.virustotal.com.



Carica il file malware: Una volta sul sito web di VirusTotal, trova l'opzione per caricare un file e seleziona il file del malware contenuto nella cartella "Malware_U3_W2_L5" sulla tua macchina virtuale. Attendi che il file venga caricato.



Avvia l'analisi: Dopo aver caricato il file, VirusTotal inizierà automaticamente l'analisi utilizzando i suoi motori antivirus e tool di analisi. Questo processo potrebbe richiedere alcuni minuti a seconda delle dimensioni del file e della carica del servizio.



Esamina i risultati: Una volta completata l'analisi, VirusTotal visualizzerà i risultati ottenuti dai vari motori antivirus e tool di analisi. I risultati mostreranno se il file è stato identificato come malware da uno o più motori antivirus, insieme ad altre informazioni rilevanti come le stringhe di testo presenti nel file, le risorse incorporate, le firme digitali, ecc.

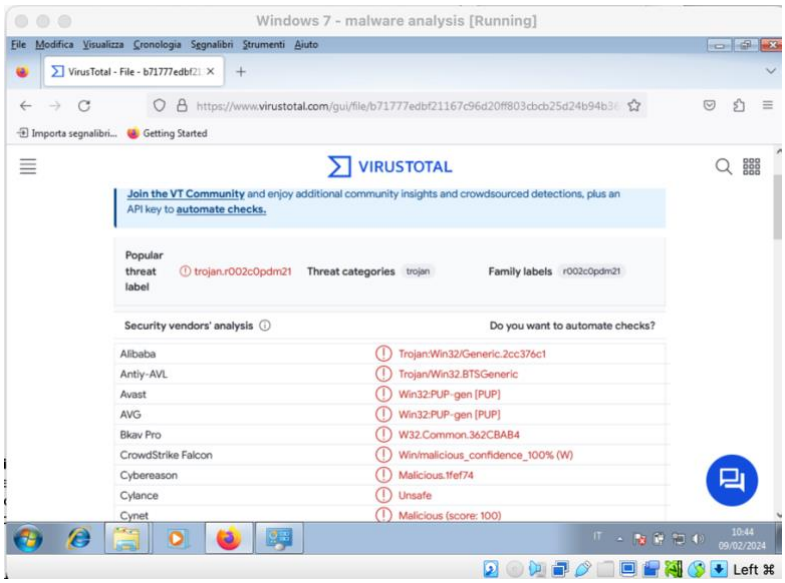
L'utilizzo di VirusTotal ci permetterà di ottenere una visione più chiara del file malware e delle sue caratteristiche. I risultati dell'analisi ci permetteranno di:

- Identificare se il file è stato precedentemente rilevato come malware da altri motori antivirus.
- Esaminare le stringhe di testo e altre informazioni presenti nel file per individuare eventuali indicatori di compromissione.
- Ottenere un'idea generale delle funzionalità e del comportamento del malware.
- Valutare il rischio potenziale che il file malware possa rappresentare per il sistema.

In conclusione, l'utilizzo di VirusTotal per eseguire un'analisi statica di base del malware ci fornirà informazioni cruciali per comprendere la

natura e il pericolo del file, aiutandoci così a pianificare le prossime fasi dell'analisi e adottare misure di difesa appropriate.

Dopo l'esame del file tramite VirusTotal, è emerso che il file è stato identificato come un malware. In particolare, VirusTotal ha rilevato la presenza del trojan "Trojan.R002C0PDM21".



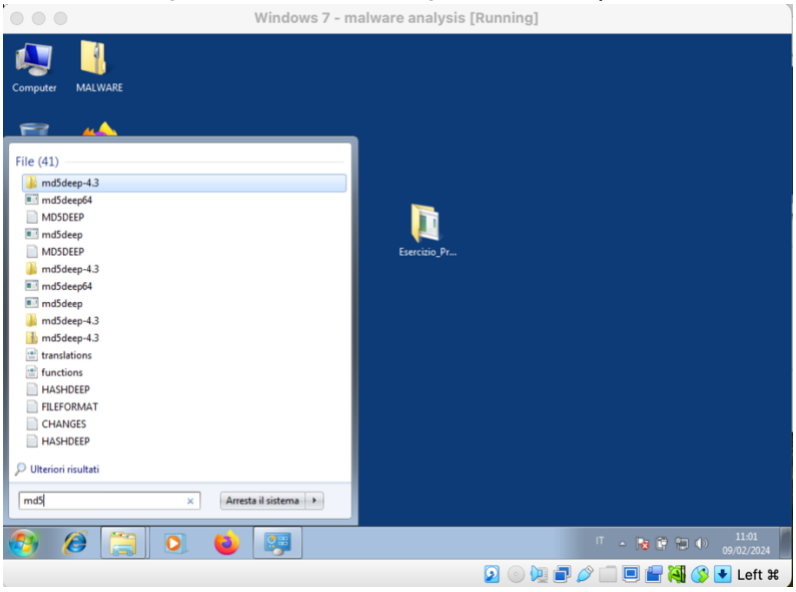
La presenza di questo tipo di trojan indica chiaramente che il file è compromesso e rappresenta una minaccia potenziale per la sicurezza del sistema.
L'identificazione di un trojan come "Trojan.R002C0PDM21" suggerisce che il malware potrebbe essere progettato per attività dannose, come il furto di informazioni personali, l'installazione di backdoor nel sistema, la distribuzione di altri malware o altre azioni dannose. Pertanto, è fondamentale adottare misure immediate per isolare e rimuovere il malware dal sistema al fine di prevenire eventuali danni e proteggere l'integrità dei dati e della rete.
L'analisi statica di base tramite VirusTotal ha quindi giocato un ruolo fondamentale nel rivelare la natura dannosa del file, fornendo informazioni preziose per avviare azioni correttive e mitigare il rischio associato alla presenza di questo malware nel sistema.

Un altro metodo di analisi statica di base che possiamo usare in alternativa a VirusTotal può comprendere il calcolo del codice hash del file dannoso.

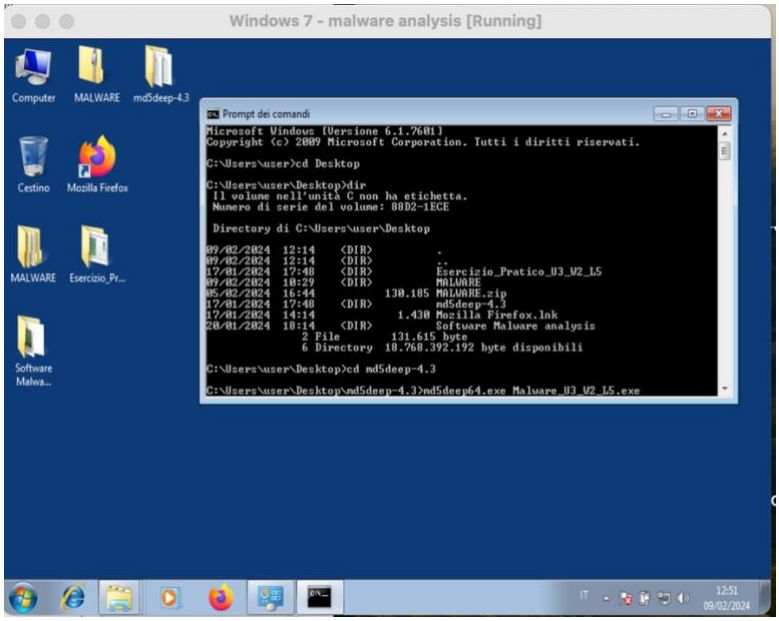
Il calcolo dell'hash è un processo che converte i dati di un file in una stringa di lunghezza fissa, chiamata "hash value" o "digest", utilizzando un algoritmo hash crittografico. Questo valore hash è unico per ogni file e può essere utilizzato per identificare un file specifico in modo univoco.

Il calcolo dell'hash è utile per verificare l'integrità dei file, confrontare file per identificare duplicati, e identificare file noti o sospetti tramite database di hash conosciuti di malware. Inoltre, può essere utilizzato per rilevare anche le più piccole modifiche apportate a un file.
In questo caso utilizzeremo md5deep4.3 installato sulla macchina virtuale Windows 7 di default.
Il programma md5deep è un'utilità a riga di comando che permette di calcolare gli hash di file e di confrontarli con un database di hash noti.
Ecco come utilizzarlo:

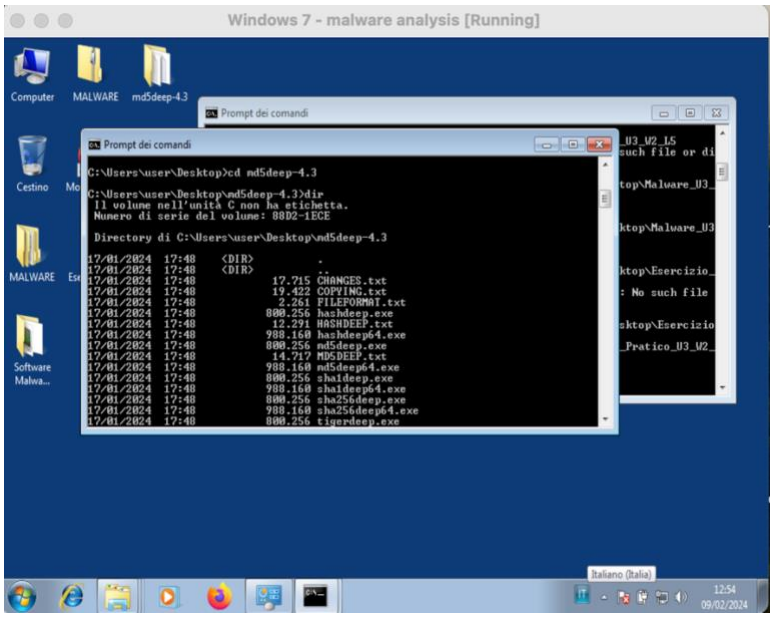
2. **Installazione di md5deep**: Se md5deep non è già installato sul tuo sistema, puoi scaricarlo e installarlo dalla sua pagina GitHub (https://github.com/jessek/hashdeep/releases).



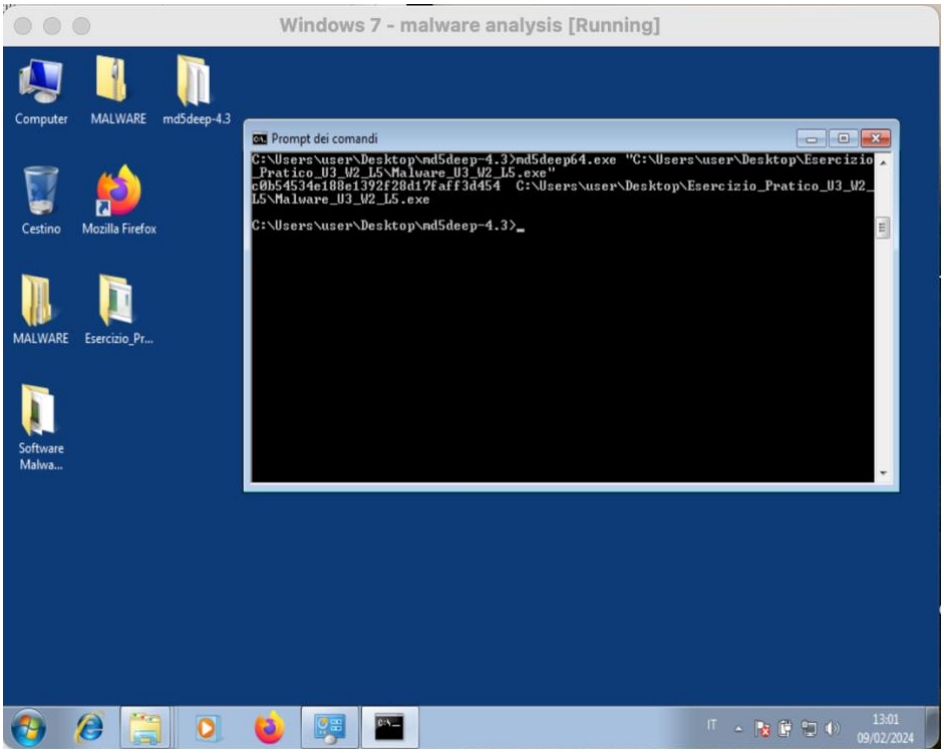
3. **Calcolo dell'hash:** per eseguire il programma md5deep-4.3 dobbiamo aprire il prompt dei comandi, redirigerci nella directory del Desktop dove sono presenti i programmi eseguibili sia del Malware (Malware_U3_W2_L5) sia di md5deep-4.3.



Sempre con il comando cd md5deep-4.3 ci redigiamo all'interno della directory dei programmi e potremo notare che vi sono vari programmi eseguibili con diverse versioni per il calcolo degli hash.



In tal caso sappiamo che la versione del file del nostro malware è di 64bit, dunque, eseguiremo direttamente md5deep64.exe che supporta appunto file con questa versione.
Possiamo procedere con l'esecuzione del programma md5deep64.exe per analizzare gli hash del Malware_U3_W2_L5 all'interno della cartella Esercizio_Pratico_U3_W2_L5 con i seguenti comandi
md5deep64.exe"C:\Users\NomeUtente\Desktop\Esercizio_Pratico_U3_W2_L5\Malware_U3_W2_L5.exe"
Da notare che gli apici "" sono quasi essenziali se alcuni file contengono caratteri speciali come in questo caso (_).



Come risultato il calcolo dell'hash del file malware è:
c0b54534e188e1392f28d17faff3d454

Scopo e beneficio:
Il calcolo dell'hash con md5deep fornisce un metodo rapido ed efficace per identificare in modo univoco un file e confrontarlo con database noti di malware. Questo ci consente di determinare se un file è già noto come malware e di prendere misure appropriate di conseguenza.
Utilizzando md5deep, siamo in grado di ottenere un'indicazione preliminare sull'integrità e sulla sicurezza del file malware attraverso l'analisi statica di base, aiutandoci così a comprendere meglio la natura e il rischio associato al file.

Appunto, il confronto con il database ci consente di percepire subito se siamo difronte ad un file malevolo, nel caso in cui dobbiamo effettuare un confronto con un database già presente all'interno delle macchine possiamo utilizzare il seguente comando:

md5deep64.exe -r -k "C:\percorso\database_hash.txt" "C:\Users\NomeUtente\Desktop\Esercizio_Pratico_U3_W2_L5\Malware_U3_W2_L5"

Naturalmente questo comando deve essere personalizzato in base al nome del database.

Nello specifico **md5deep -r -k database_hash.txt Malware_U3_W2_L5** esegue un confronto tra gli hash dei file presenti nella cartella specificata ("Malware_U3_W2_L5") e quelli presenti nel database di hash noti ("database_hash.txt"). L'opzione "-r" indica ancora una volta che l'operazione deve essere eseguita in modo ricorsivo su tutte le sottocartelle.
Utilizzando l'opzione "-k", il comando cerca corrispondenze tra gli hash dei file nella cartella specificata e quelli presenti nel database di hash noti. Se trova una corrispondenza, significa che il file ha un hash noto nel database e potrebbe essere un file conosciuto come malware o con altre caratteristiche note.

Invece se siamo in presenza di una cartella con più file e non siamo sicuri della loro origine e della loro funzione, possiamo esaminare gli hash di tutti i file con il seguente comando:

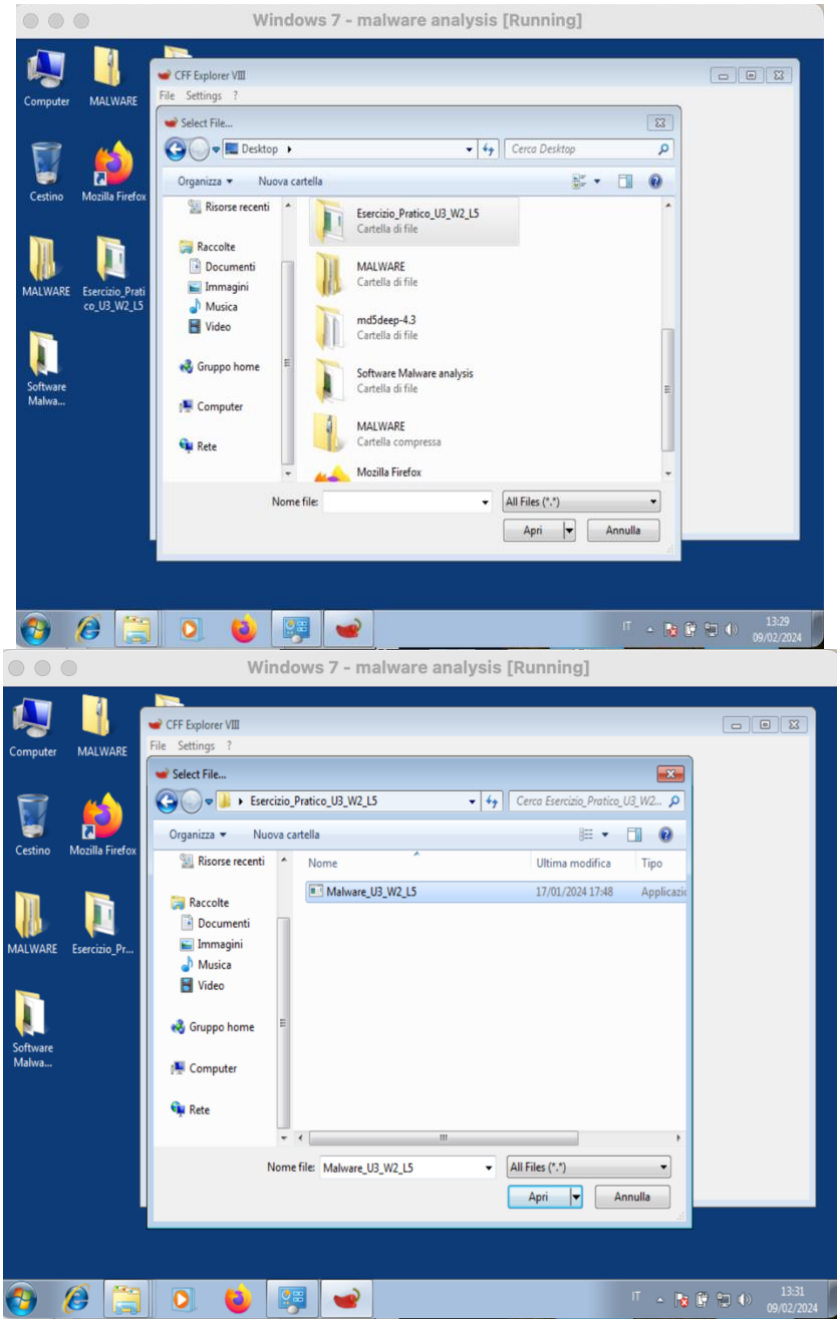
md5deep64.exe -r "C:\Users\NomeUtente\Desktop\Esercizio_Pratico_U3_W2_L5\Malware_U3_W2_L5"
Questo comando esegue un'operazione di hashing ricorsiva sulla cartella specificata, in questo caso "Malware_U3_W2_L5". L'opzione "-r" indica che l'operazione di hashing deve essere eseguita in modo ricorsivo, cioè deve includere tutti i file presenti nella cartella specificata e nelle sue sottocartelle.
L'hashing viene effettuato su ciascun file individuale all'interno della cartella e delle sottocartelle, generando un valore hash univoco (tipicamente MD5) per ciascun file. Questo può essere utile per verificare

l'integrità dei file all'interno della cartella e identificare eventuali modifiche o corruzioni.

Proseguendo con l'esercizio in questa settimana abbiamo studiato e ripetuto cosa sono le librerie, queste funzioni che vengono importate all'interno di un programma e quale relazione ci sia tra librerie e malware. Le librerie, nel contesto informatico, sono insiemi di codice predefinito che contengono funzioni e procedure comuni che possono essere richiamate da altri programmi. Esse permettono di riutilizzare il codice, migliorando l'efficienza e la manutenibilità del software. Nel caso dei malware, le librerie possono essere utilizzate per sfruttare vulnerabilità nel sistema, eseguire azioni dannose o aggirare misure di sicurezza.
Comprendere le librerie utilizzate da un malware è essenziale per identificarne le capacità e il comportamento. Le funzioni all'interno delle librerie possono fornire informazioni cruciali sulle azioni che il malware è in grado di compiere, consentendo agli analisti di comprendere meglio il suo funzionamento e sviluppare contro-misure appropriate.

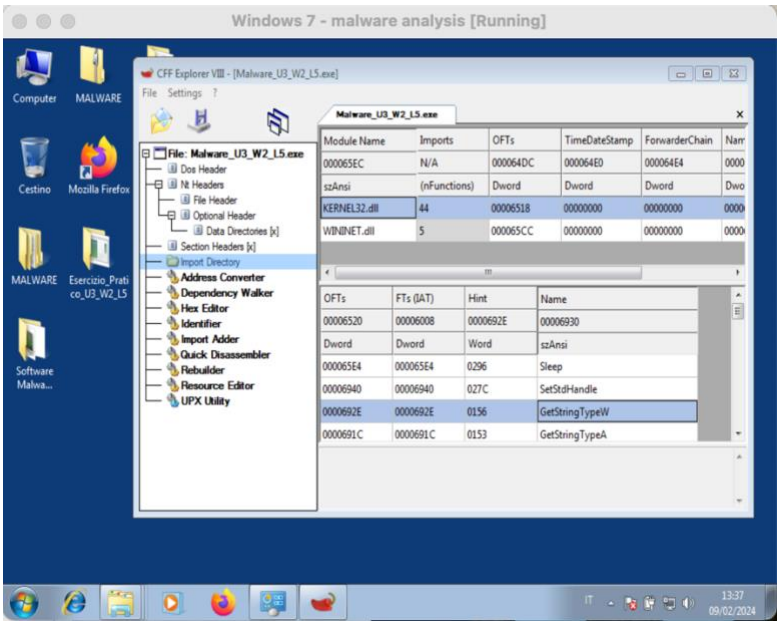
Anche in questo caso ricorreremo ad un tool installato nella macchina virtuale di default per controllare le librerie presenti all'interno del malware.
CFF Explorer è uno strumento molto utile per l'analisi dei malware e delle applicazioni in generale. Si tratta di un programma gratuito progettato per l'analisi delle strutture dei file eseguibili, compresi i file PE (Portable Executable) utilizzati nei sistemi Windows.
Questo strumento consente agli analisti di esaminare dettagliatamente i file eseguibili, inclusi i malware, e di visualizzare informazioni come le librerie importate, le funzioni utilizzate, le risorse incorporate e altro ancora.
Utilizzando CFF Explorer, saremo in grado di analizzare in dettaglio le librerie importate e le funzioni utilizzate dal malware "Malware_U3_W2_L5", fornendo così una panoramica più approfondita del suo comportamento e delle sue capacità.
Continuiamo con l'analisi del malware utilizzando CFF Explorer per ottenere ulteriori informazioni sulle sue caratteristiche e funzionalità.

Istruzioni per l'applicazione di CFF Explorer all'esercizio:
Avviamo CFF Explorer e carichiamo il file eseguibile del malware "Malware_U3_W2_L5" su CFF Explorer.



Una volta aperto il file, cerca la sezione delle librerie importate o delle funzioni utilizzate. Questa sezione mostrerà le librerie utilizzate dal malware e le funzioni chiamate all'interno di esse.
Cliccando su import noteremo quali tipo di librerie sono presente e in questo caso ci imbatteremo in librerie KERNEL32.dll che interagiscono con

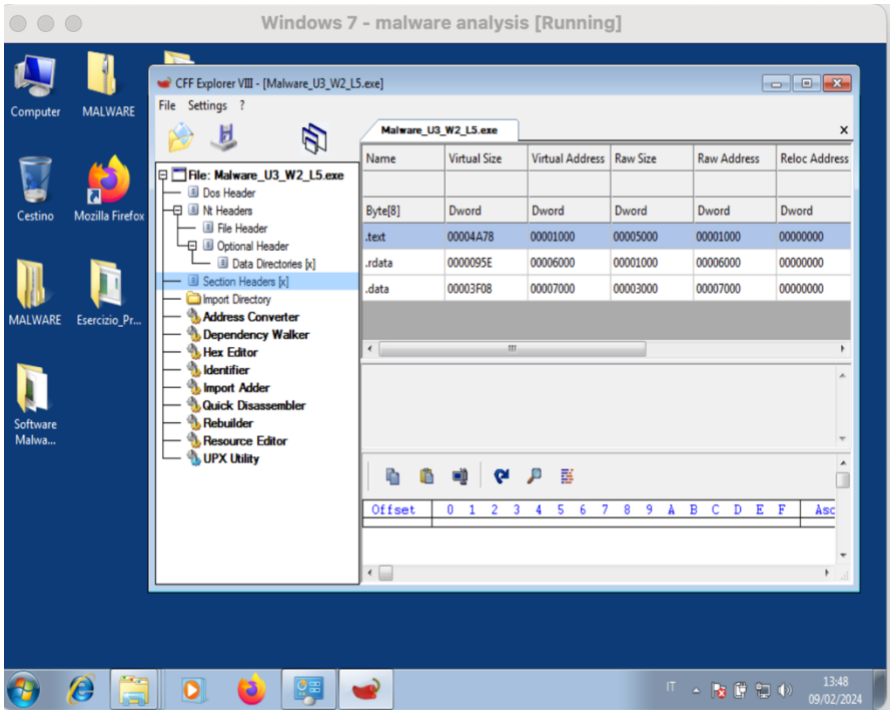
il sistema operativo e la libreria WININET.dll che implementa alcuni protocolli di rete come HTTP, FTP, NTP.



RISPOSTE ALLE DOMANDE DELLA TRACCIA

Possiamo dunque rispondere al primo quesito della traccia:
Quali librerie vengono importate dal file eseguibile?
Dal tuo utilizzo di CFF Explorer, hai scoperto che il malware utilizza le seguenti librerie:
KERNEL32.dll
WININET.dll
Queste librerie sono fondamentali per il funzionamento del malware e possono essere utilizzate per svolgere una varietà di operazioni, tra cui l'accesso alle risorse di sistema, la comunicazione in rete e altro ancora.

Per rispondere al secondo quesito invece, ci spostiamo nella sezione Section Headers nel nostro tool CFF Explorer e noteremo:



Quali sono le sezioni di cui si compone il file eseguibile del malware?
Le sezioni di cui si compone il file eseguibile del malware sono:
.text
.rdata
.data
Spiegazione delle sezioni:
.text:
La sezione .text contiene il codice eseguibile del programma. Questa è la parte del file che contiene le istruzioni del programma che vengono eseguite dal processore. Il codice in questa sezione è responsabile dell'esecuzione delle operazioni principali del malware, come l'avvio dei processi, l'infezione di altri file e altre attività dannose.
.rdata:
La sezione .rdata contiene dati di sola lettura. Questi dati sono utilizzati dal malware e possono includere stringhe, costanti o altri dati che non devono essere modificati durante l'esecuzione del programma. La presenza di dati di sola lettura può essere importante per capire meglio il comportamento del malware e le informazioni che utilizza o memorizza durante l'esecuzione.
.data:
La sezione .data contiene dati inizializzati, cioè dati che hanno un valore predefinito assegnato nel codice sorgente del programma. Questi dati possono includere variabili globali, strutture di dati e altre informazioni utilizzate dal malware durante l'esecuzione. Identificare questa sezione è

cruciale per comprendere quali dati vengono utilizzati e modificati dal malware durante la sua esecuzione.

Importanza dell'identificazione delle sezioni:

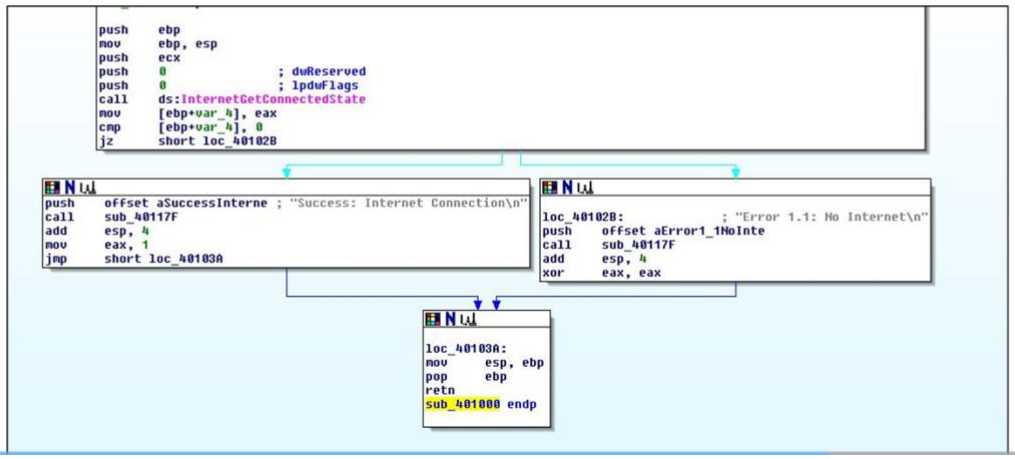
Identificare le sezioni all'interno di un file eseguibile è cruciale per comprendere il suo funzionamento e il suo comportamento. Le sezioni forniscono informazioni fondamentali sulle parti del programma, come il codice eseguibile, i dati utilizzati e le risorse incorporate. Queste informazioni sono essenziali per gli analisti dei malware, poiché consentono loro di identificare le aree del programma che potrebbero essere coinvolte in attività dannose, di isolare il comportamento sospetto e di sviluppare contro-misure appropriate.

In definitiva, l'identificazione accurata delle sezioni di un file eseguibile del malware fornisce una visione dettagliata del suo funzionamento interno, consentendo agli analisti di comprendere meglio il malware e di mitigare i suoi effetti dannosi.

Per rispondere invece alla seconda sezione del progetto analizziamo l'immagine che ci è stata data nella traccia

EPICODE **Esercizio**
Traccia e requisiti

Figura 1



1. **Identificazione dei costrutti noti:**

Creazione dello stack:
Si nota l'uso delle istruzioni **push** e **pop** per manipolare lo stack. Ad esempio, **push** viene utilizzato per inserire valori nello stack e **pop** per estrarli.
Il "lo stack" è un'area di memoria utilizzata dal programma per l'archiviazione temporanea dei dati e delle istruzioni durante l'esecuzione. Le istruzioni **push** e **pop** vengono utilizzate per gestire lo stack. Quando si esegue un'istruzione **push**, il valore viene inserito nello stack, mentre un'istruzione **pop** estrae il valore dallo stack.
I registri **ebp** ed **esp** sono comunemente utilizzati per riferirsi allo stack. **ebp** (base pointer) viene utilizzato per accedere ai parametri e alle variabili locali all'interno delle funzioni, mentre **esp** (stack pointer) indica l'indirizzo di memoria corrente dello stack. La manipolazione di questi registri consente al programma di accedere ai dati nello stack in modo efficiente.

Le istruzioni **ebp** ed **esp** sono spesso coinvolte nella gestione dello stack. **ebp** viene utilizzato come registro del frame del puntatore base ed **esp** come puntatore dello stack.
Altri costrutti:
Si nota l'utilizzo di istruzioni di confronto (**cmp**) e di salto condizionato (**jz**), che sono spesso utilizzate per implementare strutture decisionali.

È presente anche l'utilizzo di chiamate a funzioni (**call**), che potrebbero indicare l'invocazione di funzionalità del sistema operativo o di librerie importate.
Le istruzioni **cmp** (compare) vengono utilizzate per confrontare due valori. Questo è spesso utilizzato insieme a istruzioni di salto condizionale come **jz** per implementare strutture decisionali.
Le chiamate a funzioni (**call**) sono utilizzate per invocare altre funzioni all'interno del programma. Queste chiamate possono essere rivolte a funzionalità del sistema operativo o a librerie importate. Ad esempio, nel codice fornito, sembra che sia stata fatta una chiamata alla funzione **InternetGetConnectedState** per verificare lo stato della connessione Internet.

IPOTESI SUL COMPORTAMENTO DEL MALWARE

2. **Formulazione di ipotesi sul comportamento delle funzionalità implementate**

Dalle istruzioni presentate, sembra che il malware stia cercando di verificare lo stato della connessione Internet utilizzando la funzione **InternetGetConnectedState**. In base al risultato ottenuto, il malware potrebbe procedere con diversi comportamenti:

a. Se la connessione Internet fosse attiva (**Success: Internet Connection\n**), potrebbe continuare con una sequenza di operazioni.
b. Se la connessione Internet non è attiva (**Error 1.1: No Internet\n**), potrebbe eseguire un altro ramo di codice per gestire questo caso.

Inoltre, ci sono anche istruzioni di manipolazione dello stack (**push**, **pop**, **esp**, **ebp**) e di calcolo (**add**, **sub**, **xor**) che potrebbero essere utilizzate per gestire parametri e variabili locali, nonché per eseguire operazioni aritmetiche.
In conclusione, basandoci sul codice assembly fornito, possiamo ipotizzare che il comportamento del malware sia orientato al controllo dello stato della connessione Internet e all'esecuzione di azioni specifiche in base a questo stato.

CONCLUSIONE E SCOPO DEL PROGETTO

Conclusione:
In conclusione, questo progetto si è concentrato sull'analisi statica di base di un malware utilizzando diverse tecniche e strumenti di analisi. Attraverso l'approfondimento della traccia fornita e l'applicazione di concetti teorici appresi durante il corso, siamo stati in grado di esaminare dettagliatamente il comportamento e le caratteristiche del malware "Malware_U3_W2_L5". Dalle librerie importate al calcolo dell'hash, dall'analisi con VirusTotal all'utilizzo di strumenti come CFF Explorer, abbiamo esplorato una serie di metodologie utili per comprendere la natura e il potenziale impatto di un malware sul sistema.
Le risposte fornite alle domande della traccia, insieme alle ipotesi sul comportamento del malware, hanno permesso di ottenere una visione più completa delle funzionalità implementate e delle possibili azioni intraprese dal malware. Questo processo di analisi è essenziale per gli analisti della sicurezza informatica, poiché consente di identificare le minacce, sviluppare contro-misure adeguate e proteggere i sistemi da potenziali attacchi.
Scopo del Progetto:
Lo scopo principale di questo progetto è stato quello di acquisire competenze pratiche nell'analisi dei malware attraverso l'applicazione di concetti teorici e l'utilizzo di strumenti specifici. Attraverso l'esame del malware e l'implementazione delle procedure di analisi statica di base, abbiamo mirato a:

- Comprendere il funzionamento interno di un malware e identificarne le caratteristiche distintive.
- Utilizzare strumenti come VirusTotal e CFF Explorer per eseguire un'analisi approfondita delle funzionalità del malware.
- Formulare ipotesi sul comportamento del malware e suggerire possibili azioni intraprese dallo stesso.
- Fornire risposte dettagliate e ragionate alle domande poste nella traccia del progetto.

In sintesi, il progetto ha offerto un'opportunità preziosa per applicare le conoscenze teoriche acquisite durante il corso e per sviluppare competenze pratiche nel campo dell'analisi dei malware, preparandoci ad affrontare sfide reali nel campo della sicurezza informatica.