Progetto finale modulo 6

MALWARE ANALYSIS

Mohamed Asry

Malware Analysis

Il Malware da analizzare è nella cartella Build_Week_Unit_3 presente sul desktop della macchina virtuale dedicata.

Analisi statica

Con riferimento al file eseguibile Malware_Build_Week_U3, rispondere ai seguenti quesiti utilizzando i tool e le tecniche apprese nelle lezioni teoriche:

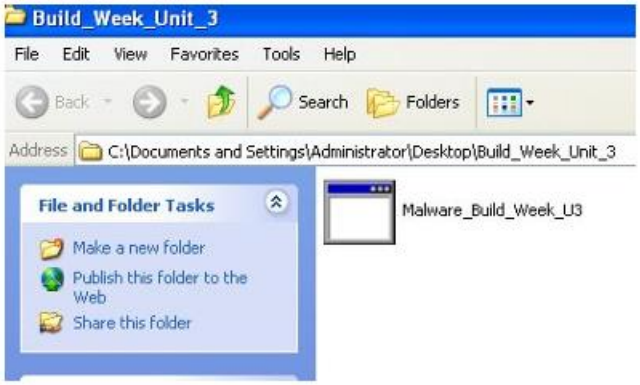
- Quanti parametri sono passati alla funzione Main()?
- Quante variabili sono dichiarate all'interno della funzione Main()?
- Quali sezioni sono presenti all'interno del file eseguibile? Descrivete brevemente almeno 2 di quelle identificate
- Quali librerie importa il Malware? Per ognuna delle librerie importate, fate delle ipotesi sulla base della sola analisi statica delle funzionalità che il Malware potrebbe implementare. Utilizzate le funzioni che sono richiamate all'interno delle librerie per supportare le vostre ipotesi.

Con riferimento al Malware in analisi, spiegare:

- Lo scopo della funzione chiamata alla locazione di memoria **00401021**
- Come vengono passati i parametri alla funzione alla locazione **00401021**;
- Che oggetto rappresenta il parametro alla locazione **00401017**
- Il significato delle istruzioni comprese tra gli indirizzi **00401027** e **00401029**.
- Con riferimento all'ultimo quesito, tradurre il codice Assembly nel corrispondente costrutto C.
- Valutate ora la chiamata alla locazione **00401047**, qual è il valore del parametro «ValueName»?

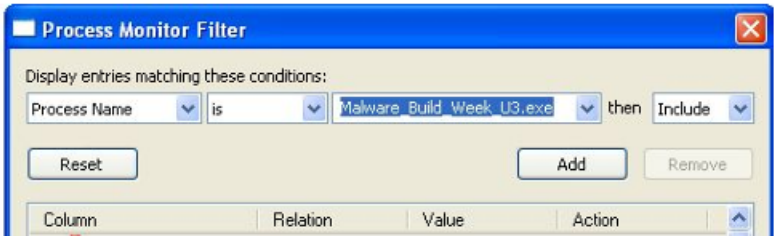
Analisi dinamica

Preparate l'ambiente ed i tool per l'esecuzione del Malware (suggerimento: avviate principalmente Process Monitor ed assicurate di eliminare ogni filtro cliccando sul tasto «reset» quando richiesto in fase di avvio). Eseguite il Malware, facendo doppio click sull'icona dell'eseguibile



- Cosa notate all'interno della cartella dove è situato l'eseguibile del Malware? Spiegate cosa è avvenuto, unendo le evidenze che avete raccolto finora per rispondere alla domanda

Analizzate ora i risultati di Process Monitor (consiglio: utilizzate il filtro come in figura sotto per estrarre solo le modifiche apportate al sistema da parte del Malware). Fate click su «ADD» poi su «Apply» come abbiamo visto nella lezione teorica.



Filtrate includendo solamente l'attività sul registro di Windows.

- Quale chiave di registro viene creata?
- Quale valore viene associato alla chiave di registro creata?

Passate ora alla visualizzazione dell'attività sul file system.

- Quale chiamata di sistema ha modificato il contenuto della cartella dove è presente l'eseguibile del Malware?

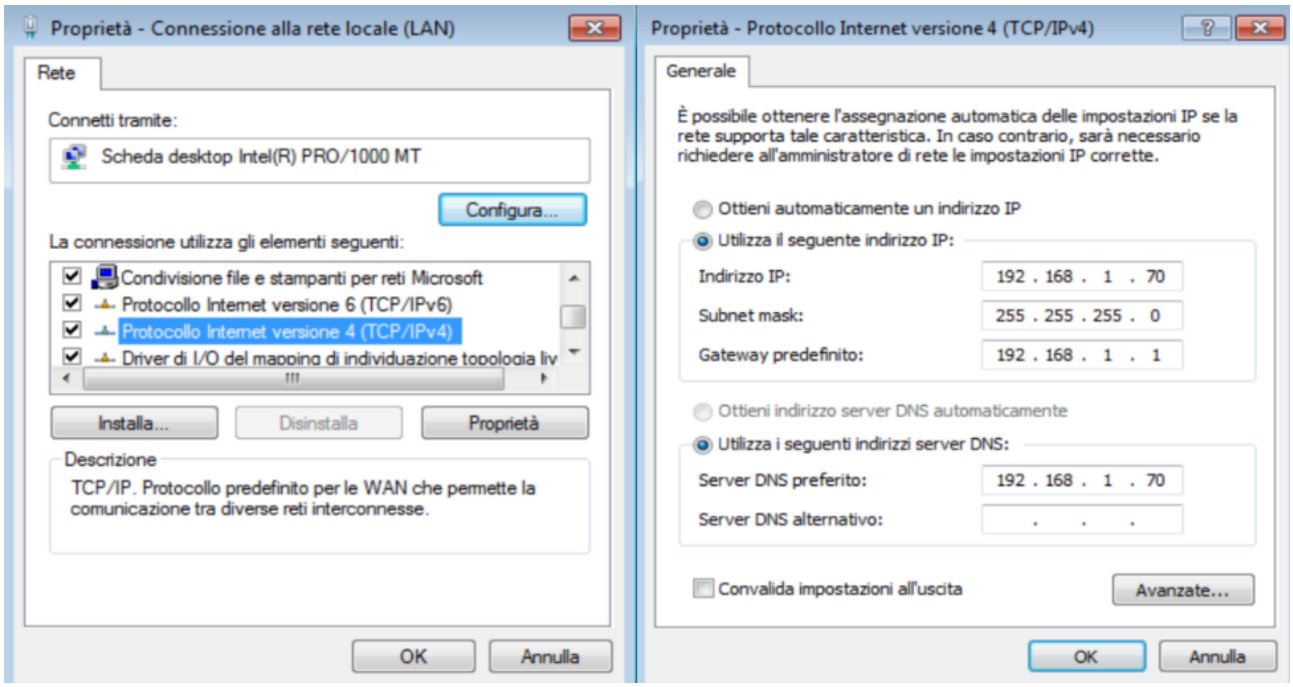
Unite tutte le informazioni raccolte fin qui sia dall'analisi statica che dall'analisi dinamica per delineare il funzionamento del Malware.

ANALISI STATICA

Ricreo un'istantanea da virtualbox della macchina windows prima di iniziare, per poter ripristinare in caso di problemi, e visto che andrò ad eseguire il malware mi assicuro di rispettare i seguenti accorgimenti:

- disattivare controller usb
- disattivare comunicazione con la rete (solo int)
- disabilitare la condivisione delle cartelle
- disabilitare appunti condivisi (copia/incolla)

Prima di avviare il malware, mi assicuro di andare a catturare tutti gli eventi aprendo Procmon, al fine di identificare eventuali azioni del malware su processi e thread, e modifiche registro; avvio Regshot per confrontare tramite screenshot (prima/dopo) le modifiche che avverranno a livello di sistema, ed imposto un ip statico per vedere tramite Apatedns le chiamate che il malware andrà a fare nel web . Avendolo isolato dalla rete ovviamente non potrà eseguire tutte le sue funzioni.



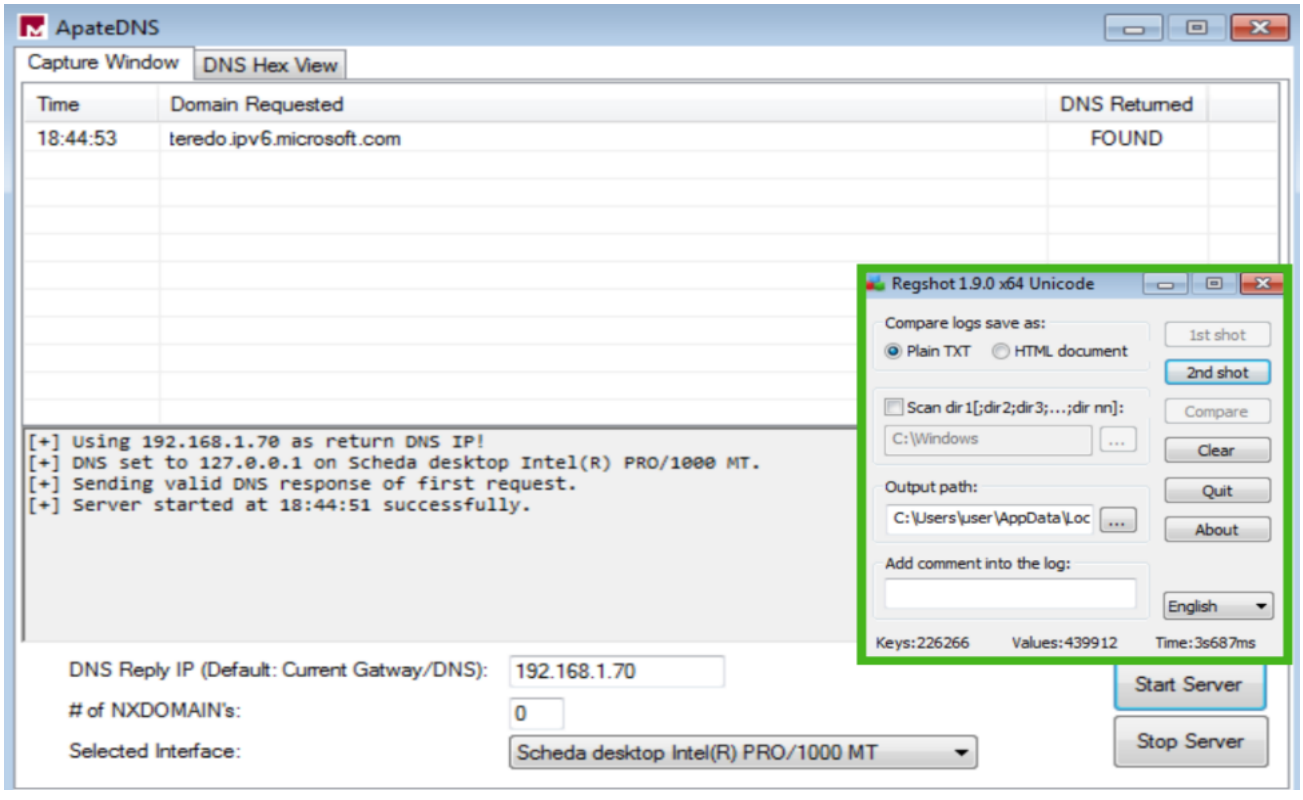

```
C:\Users\user>ipconfig

Configurazione IP di Windows

Scheda Ethernet Connessione alla rete locale (LAN):

   Suffisso DNS specifico per connessione:
   Indirizzo IPv6 locale rispetto al collegamento . : fe80::e055:3d47:f6e7:dfee%
11
   Indirizzo IPv4. . . . . . . . . . . . : 192.168.1.70
   Subnet mask . . . . . . . . . . . . . : 255.255.255.0
   Gateway predefinito . . . . . . . . . : 192.168.1.1

Scheda Tunnel isatap.{9AC0F9A2-BF82-47F4-8F6D-C16B1D79E059}:
```



Per identificare i parametri passati alla funzione main(), abbiamo impiegato IDA Pro, uno strumento avanzato di disassemblaggio, è emerso che la funzione main() accetta tre parametri principali:

- **argc:** Un intero (int) che indica il numero di argomenti forniti al programma tramite la riga di comando. Questo parametro fornisce il conteggio degli argomenti.
- **argv:** Un array di stringhe (puntatori a char) che contiene gli argomenti passati tramite la riga di comando. Ogni elemento dell'array corrisponde a un singolo argomento.
- **envp:** Un array di stringhe che rappresentano le variabili d'ambiente dell'utente. Queste variabili possono influenzare il comportamento del programma e determinare le condizioni di esecuzione.

All'interno della funzione main(), sono dichiarate cinque variabili locali, utilizzate per gestire dati temporanei e facilitare le operazioni durante l'esecuzione del malware. Le variabili dichiarate sono:

- **hmodule**
- **data**
- **var_117**
- **var_8**
- **var_4**

Queste variabili vengono allocate nello stack e gestite tramite istruzioni di manipolazione dello stack, come push e pop.

L'eseguibile del malware è organizzato in diverse sezioni, ognuna con uno scopo specifico. Le sezioni principali identificate sono:

- **.text:** Contiene il codice eseguibile del malware, ovvero le istruzioni che la CPU eseguirà quando il malware verrà avviato. Questa sezione è fondamentale, poiché racchiude la logica del malware e le sue operazioni principali.
- **.rdata:** Comprende dati di sola lettura, come le informazioni sulle librerie importate e le stringhe di testo. Questa sezione può contenere nomi di funzioni e variabili utilizzate dal malware durante l'esecuzione, ed è utile per identificare le sue dipendenze e le risorse necessarie.
- **.data:** Contiene variabili globali e dati accessibili da qualsiasi parte del programma. Viene utilizzata per memorizzare informazioni che devono persistere durante l'esecuzione del malware.
- **.rsrc:** Include risorse non di codice, come icone, immagini e altri dati non eseguibili che sono necessari per l'interfaccia utente del malware o per altre funzioni.

Le librerie importate sono KERNEL32.dll e ADVAPI32.dll.

KERNEL32.dll è un elemento vitale del sistema operativo Windows. È una libreria di collegamento dinamico (DLL), che è una raccolta di funzioni utilizzate da vari programmi sul tuo computer.
Nel dettaglio, le operazioni che gestisce KERNEL32.dll sono:
● Gestione della memoria;
● Input/Output (I/O);
● Gestione dei file: KERNEL32.dll è anche responsabile della gestione dei file. Questo include la creazione, la lettura e la scrittura di file sul disco rigido;

● Comunicazione con l’hardware: KERNEL32.dll facilita la comunicazione tra i programmi e l’hardware del computer. Questo è essenziale per le operazioni come l’invio di comandi alla CPU o la lettura dei dati dalla RAM.

ADVAPI32.dll è una libreria di collegamento dinamico (DLL) fondamentale nei sistemi Windows, che fornisce funzioni essenziali per sicurezza, autenticazione e controllo degli accessi. Agisce come ponte tra le applicazioni e i meccanismi di sicurezza sottostanti del sistema operativo. Tra le funzionalità, di particolare importanza è il controllo dell'accesso al registro, che permette la gestione delle autorizzazioni di accesso per le chiavi e i valori del registro, garantendo la sicurezza dei dati.

ANALISI DINAMICA

L'analisi dinamica comporta l'esecuzione del malware in un ambiente controllato per osservare direttamente il suo comportamento. Questa fase fornisce dettagli sulle modifiche apportate al file system, alle chiavi di registro e alle connessioni di rete.

A. Osservazioni nella cartella del malware
Dopo l'esecuzione del malware, sono state rilevate modifiche nella cartella contenente l'eseguibile. I principali cambiamenti osservati includono:

- Creazione di nuovi file e cartelle: Il malware può generare nuovi file o cartelle all'interno della propria directory per memorizzare ulteriori payload o configurare il sistema in modo che il malware venga eseguito automaticamente.
- Ridenominazione di file: Il malware potrebbe rinominare file esistenti o l'eseguibile stesso per nascondere la propria presenza, rendendo più difficile la sua identificazione. Questa tecnica può essere utilizzata per eludere il rilevamento da parte degli strumenti di sicurezza.

B. Risultati di Process Monitor
Process Monitor (ProcMon) è stato impiegato per monitorare le attività del malware durante l'esecuzione. I risultati dell'analisi hanno mostrato:

- Modifiche al registro: Il malware ha alterato chiavi di registro per ottenere persistenza. Questo può includere la creazione di nuove chiavi o la modifica di chiavi esistenti per garantire che il malware venga eseguito automaticamente all'avvio del sistema.
- Attività sui file: Il malware potrebbe aver creato, modificato o eliminato file per nascondere la propria presenza o preparare il sistema per ulteriori attacchi. Queste modifiche possono includere la creazione di file di log, la scrittura di dati temporanei o la cancellazione di file per evitare la scoperta.

C. Filtraggio dell’attività del registro
Filtrando le attività di registro con Process Monitor, è possibile ottenere informazioni dettagliate su:

- Chiave di registro creata: Ad esempio, una chiave potrebbe essere HKCU\Software\Microsoft\Windows\CurrentVersion\Run\Malware.
- Valore associato alla chiave di registro: Questo valore potrebbe rappresentare il percorso dell'eseguibile del malware, come C:\ProgramData\Malware.exe. Questa chiave viene utilizzata per assicurare che il malware venga eseguito ad ogni avvio del sistema.

D. Modifiche al file system
Durante l'analisi del file system, è emerso che: chiamata di sistema che ha modificato la cartella del malware: Chiamate di sistema come MoveFileEx o CreateFile sono state utilizzate per rinominare, creare o modificare file. Queste operazioni fanno parte della strategia del malware per offuscare la propria presenza, rendendo più difficile il rilevamento e la rimozione.

Analisi Dettagliata della Funzione Specifica

A. Scopo della funzione alla locazione di memoria 00401021
Alla locazione di memoria 00401021, il malware invoca la funzione RegCreateKeyExA, utilizzata per creare una nuova chiave di registro o aprirne una esistente. Il principale obiettivo di questa funzione è ottenere persistenza, creando o modificando chiavi nel registro di sistema per assicurare che il malware venga eseguito automaticamente all'avvio del sistema.

B. Passaggio dei parametri alla funzione
I parametri vengono passati alla funzione RegCreateKeyExA tramite lo stack. Alla locazione 00401017, l'indirizzo della sottochiave di registro viene caricato nello stack. I parametri vengono letti dalla funzione in ordine inverso, quindi il valore più recente è il primo ad essere utilizzato.

C. Oggetto del parametro alla locazione 00401017
Alla locazione 00401017, il parametro rappresenta il nome della sottochiave di registro da creare. Questo parametro è un puntatore a una stringa che specifica il nome della chiave, e può influenzare il comportamento del malware modificando le impostazioni di esecuzione.

D. Significato delle istruzioni tra 00401027 e 00401029
Le istruzioni tra 00401027 e 00401029 includono:

- test (00401027): Questa istruzione verifica se un valore è zero. Non modifica il valore, ma imposta il flag ZF (zero flag) a 1 se il risultato dell'operazione è zero.
- JZ (00401029): Salta a un'altra locazione di memoria se il flag ZF è impostato, indicando che il risultato dell'istruzione precedente era zero.

E. Valore del parametro ValueName alla locazione 00401047

Alla locazione 00401047, il parametro ValueName è la stringa "ginadll". Questo valore rappresenta il nome della voce di registro che il malware sta cercando di impostare. L'uso di nomi come "ginadll" può essere un tentativo di mascherare l'attività malevola, rendendo più difficile per gli utenti e per i software di sicurezza identificare la vera natura del malware.

CONCLUSIONI

Guardando nel dettaglio le funzioni importate dalle singole librerie, possiamo ipotizzare che il malware modifichi il registro di sistema di Windows, creando delle nuove chiavi e assegnando loro un valore, in questo modo ipotizziamo che il malware riesca ad ottenere la persistenza nel nostro sistema operativo.
Inoltre, tra le funzioni importate dalla libreria Kernel32.dll troviamo le più comuni per la realizzazione di un dropper.
L'analisi combinata, sia statica che dinamica, del malware ha rivelato diverse tecniche impiegate per garantire la persistenza e nascondere le sue attività. Il malware utilizza funzioni del registro di sistema per mantenere la propria esecuzione, apportando modifiche alle chiavi e ai valori del registro. Inoltre, il comportamento osservato, come la creazione di file e le modifiche al file system, indica che il malware adotta strategie per mascherare la sua presenza e rendere più difficile il rilevamento. Il malware sfrutta tecniche di persistenza basate sul registro e altera il file system per assicurarsi la continua esecuzione e occultare le sue tracce. Questa comprensione è fondamentale per sviluppare contromisure efficaci e proteggere i sistemi compromessi da ulteriori attacchi.