Progetto Modulo M6 – Maria Ludovica Tartaglia

Traccia:

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:

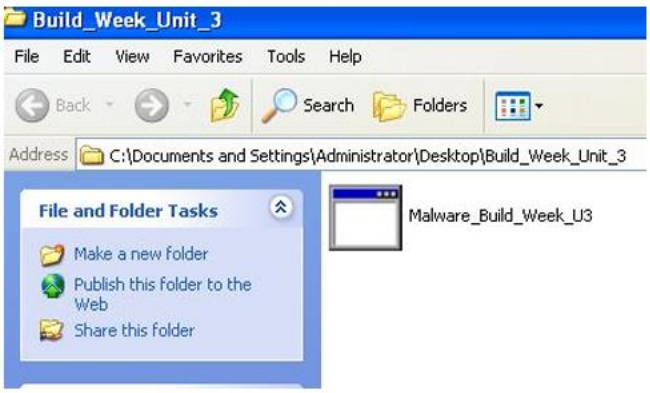
- Quanti parametri sono passati alla funzione Main()?
- Quante variabili sono dichiarate all'interno della funzione Main()?
- Quali sezioni sono presenti all'interno del file eseguibile? Descrivete brevemente almeno 2 di quelle identificate
- Quali librerie importa il Malware? Per ognuna delle librerie 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»?

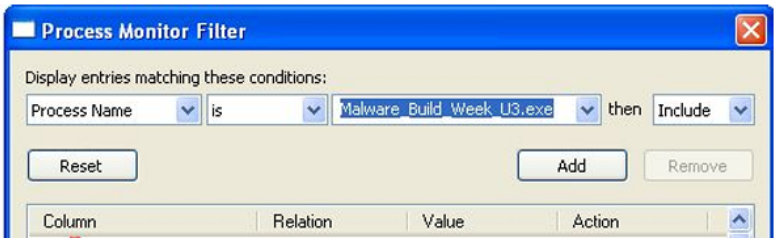
Analisi dinamica

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?

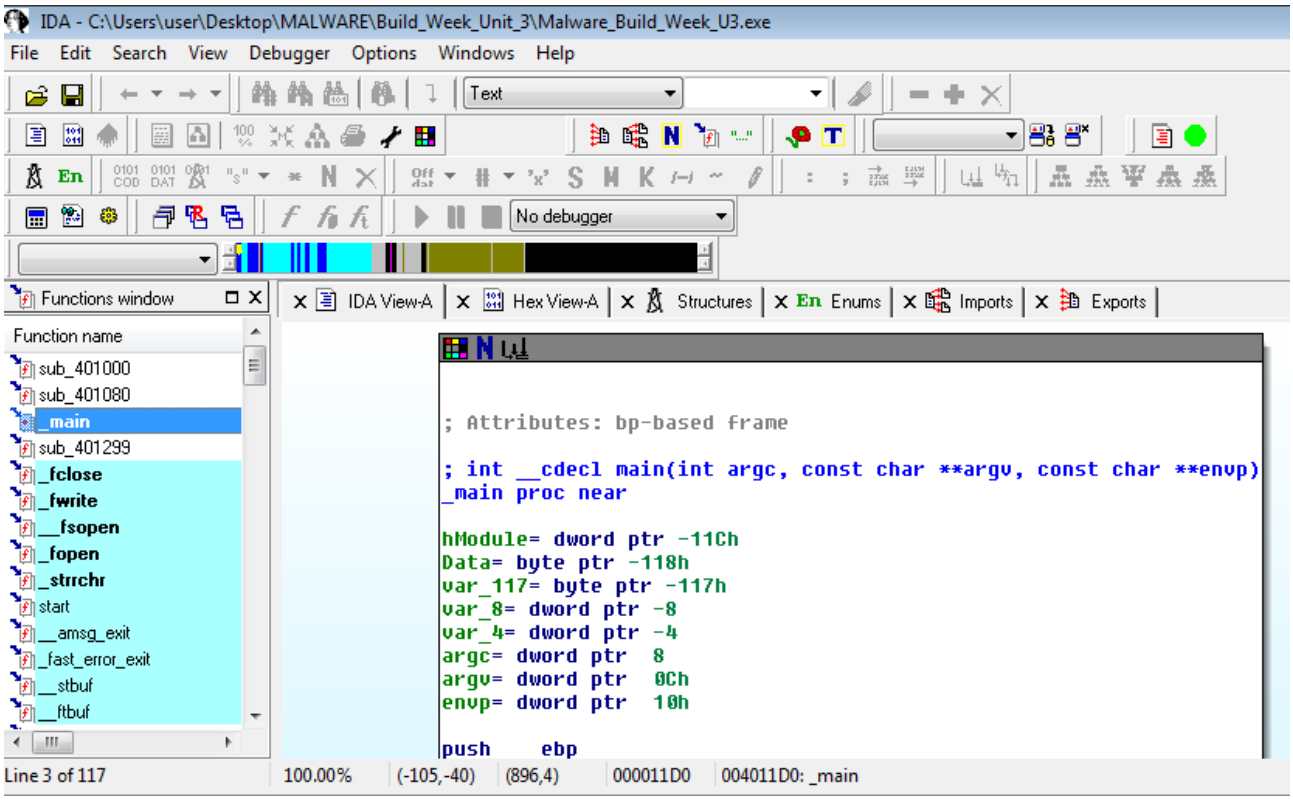
Unite tutte le informazioni raccolte fin qui sia dall'analisi statica che dall'analisi dinamica per delineare il funzionamento del Malware.

Svolgimento:

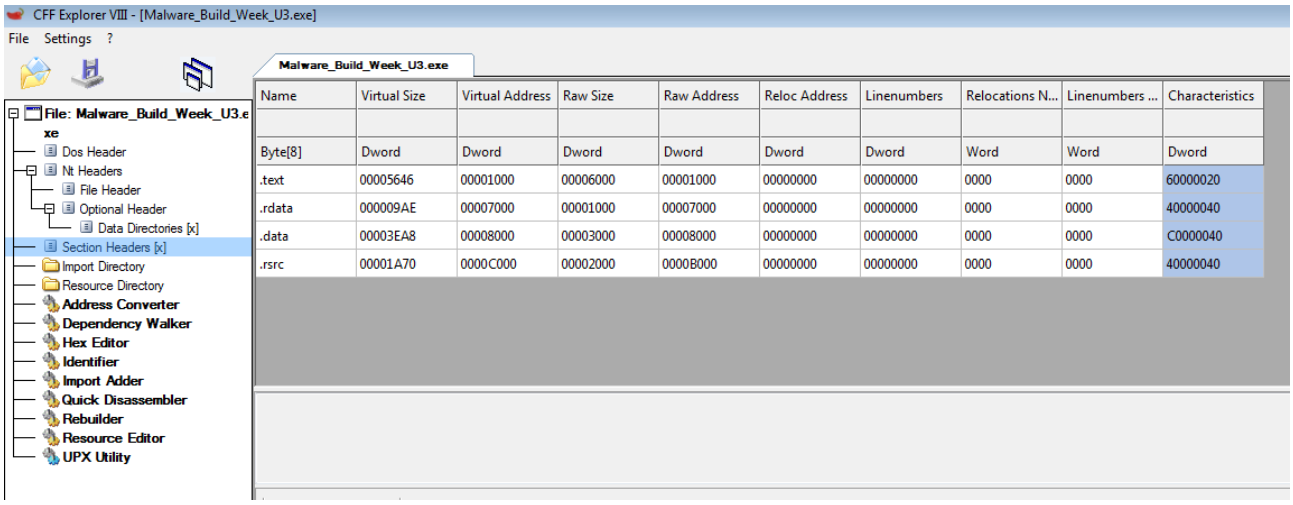
Effettuare l'analisi dei malware è una pratica di fondamentale importanza, perché permette di studiarne il comportamento. Esistono due tipologie di analisi che possiamo effettuare per studiare il malware: una di tipo statico, che analizza il programma malevolo senza eseguirlo, ed una di tipo dinamico, grazie alla quale il malware viene eseguito in un ambiente protetto (c.d. "sandbox") e fa sì che si possa osservarne il comportamento e le modalità di azione e di interazione con il sistema con cui entra in contatto.
A loro volta, entrambe le tipologie di analisi possono essere di tipo basico o di tipo avanzato, con delle sostanziali differenze: nell'analisi statica basica, il file viene eseguito senza entrare nel dettaglio delle singole istruzioni, di modo da poter determinare se sia malevolo o meno e carpire informazioni generali sulla sua operatività. L'analisi dinamica basica tenterà, eseguendo il file nell'ambiente protetto di cui parlavamo sopra, di neutralizzare il file infetto; l'analisi statica avanzata, invece, attraverso strumenti di reverse engineering cercherà di identificare il comportamento del malware attraverso l'analisi delle sue istruzioni, avvalendosi dell'utilizzo di tools come i disassembler, che permettono di convertire il file in linguaggio assembly, lavorando pertanto sul codice sorgente del malware stesso. L'analisi dinamica avanzata, infine, raccoglierà le informazioni su come il file malevolo agisce e si interseca nel sistema attraverso un'azione di

“debugging”. Compito di un analyst in tal senso sarà combinare l’utilizzo delle diverse tipologie di analisi al fine di avere un quadro completo del malware che ci si presenti davanti e delle sue funzionalità a 360 gradi.

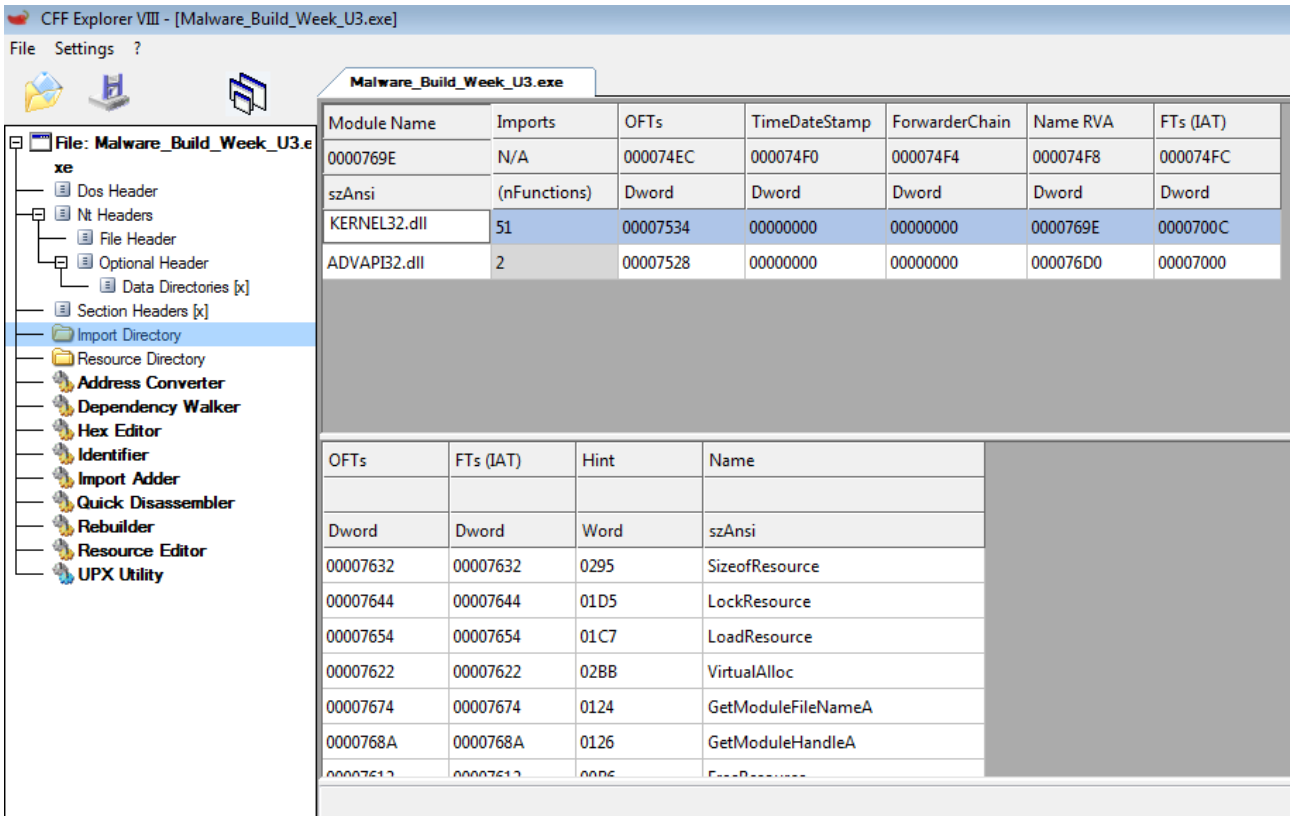
Cominciando con ciò che viene richiesto dalla traccia, partiamo con l’eseguire un’analisi statica del malware che ci viene sottoposto.
Utilizzeremo alcuni tools molto utili per effettuare il nostro lavoro. Uno di questi è IDA Pro, che permette di individuare la funzione Main ed i suoi parametri come richiesto. I parametri passati dalla funzione sono tre: argc, argv e envp. Notiamo poi la presenza di cinque variabili dichiarate, e di un int e due char:



Cerchiamo poi le sezioni di file eseguibile, che vediamo essere di tutte le tipologie studiate:
- .text, che contiene le righe di codice (cioè le istruzioni) che la CPU eseguirà una volta che il software sarà attivato;
- .rdata, che include solitamente le informazioni sulle librerie e le funzioni che vengono importate dal file eseguibile;
- .data, che conterrà i dati e le variabili globali dell’eseguibile e che sono disponibili in qualsiasi parte del programma;
- .rsrc, che include tutte le risorse utilizzate dal file, quali immagini o stringhe che non fanno parte dell’eseguibile stesso.



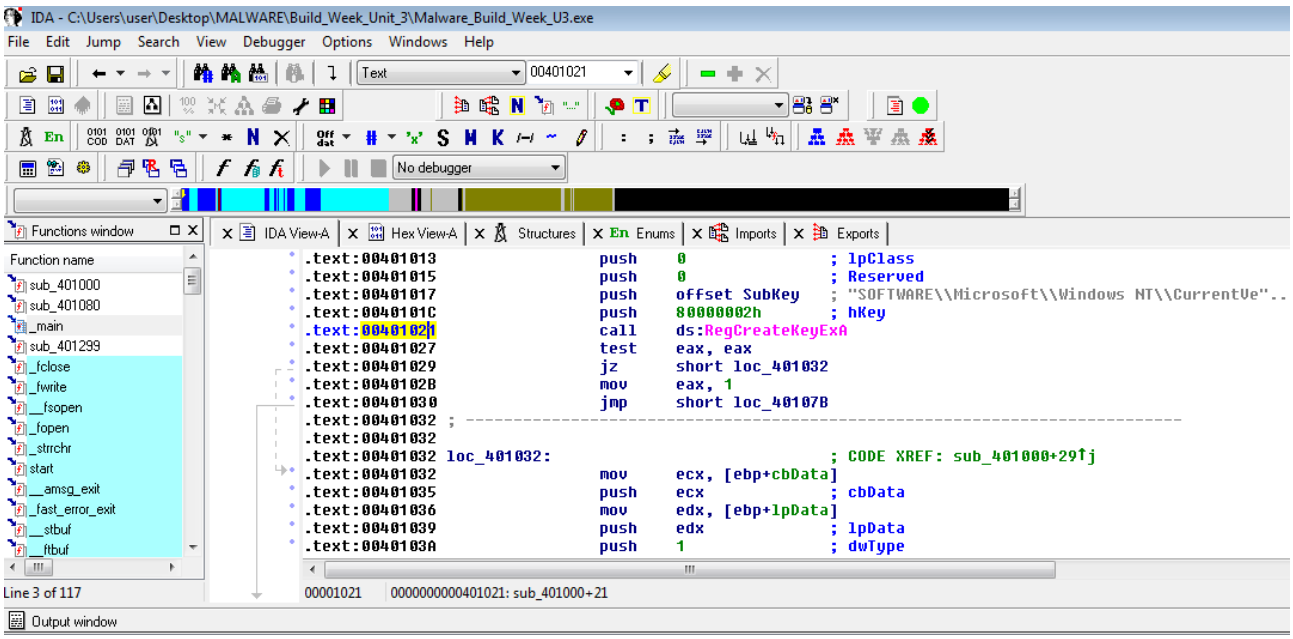
Utilizzando un altro importante strumento per la malware analysis quale è CFF Explorer, vediamo poi le librerie importate dal malware, che risultano essere KERNEL32 e ADVAPI32:



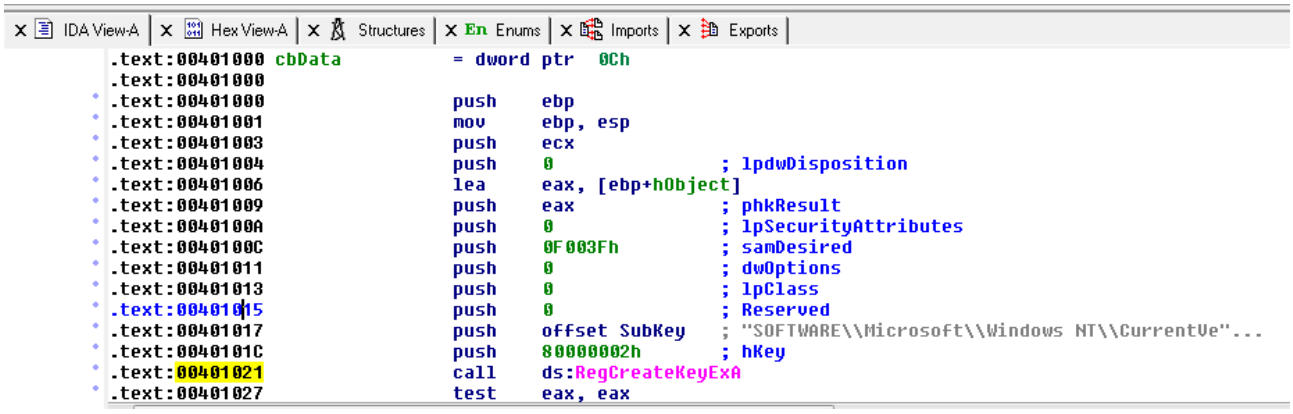
In base alle funzioni che vediamo richiamate poco sotto, possiamo pensare si tratti di un programma malevolo che contiene un malware, detto **dropper.** I dropper sono programmi malevoli che contengono al loro interno un malware, estraendolo per salvarlo su disco. Per effettuare ciò, utilizzeranno delle specifiche APIs, come ad esempio FindResource() LoadResource()

LockResource() SizeOfResource(); ognuna di esse permetterà di localizzare nella sezione “risorse” il malware da estrarre, salvandolo successivamente per poterlo poi eseguire.

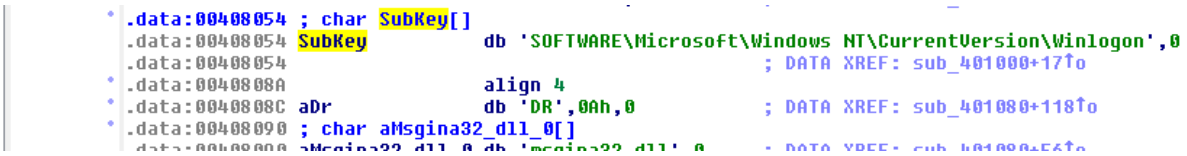
La funzione chiamata all’indirizzo di memoria 00401021, come richiesto dalla traccia, utilizza **RegCreateKeyExA** nella libreria ADVAPI32. Questa funzione va, come il nome suggerisce, a creare una chiave di registro. I parametri richiamati dalla funzione saranno “**hKey**”, “**subKey**”. “**lpSecurityAttributes**”



I parametri alla funzione alla locazione 00401021 vengono passati attraverso la creazione di uno stack, attraverso delle push, come si evince dallo screenshot a seguire:



Sempre nello screenshot di cui sopra, notiamo che il parametro alla locazione 00401017 rappresenta la creazione dell’oggetto “SubKey”, rappresentante la sottocartella (o, letteralmente, “Sottochiave”) che sarà poi creata dal malware nel sistema. Andiamo a comprovare quanto appena spiegato nello screen che segue:



Le istruzioni comprese tra gli indirizzi 00401027 e 00401029 rappresentano, rispettivamente, una test e una jz. L'istruzione test si comporta come una AND, senza però andare a modificare il valore degli operandi. La jz (Jump if Zero) è invece un salto condizionale, facendo si che il programma salti ad una nuova posizione di memoria solamente se l'operazione precedente abbia dato un risultato nullo.

La traccia richiede poi una trasformazione da Assembly nel corrispondente C, può essere sintetizzato come una if/else, dove:

if (EAX ==0) {

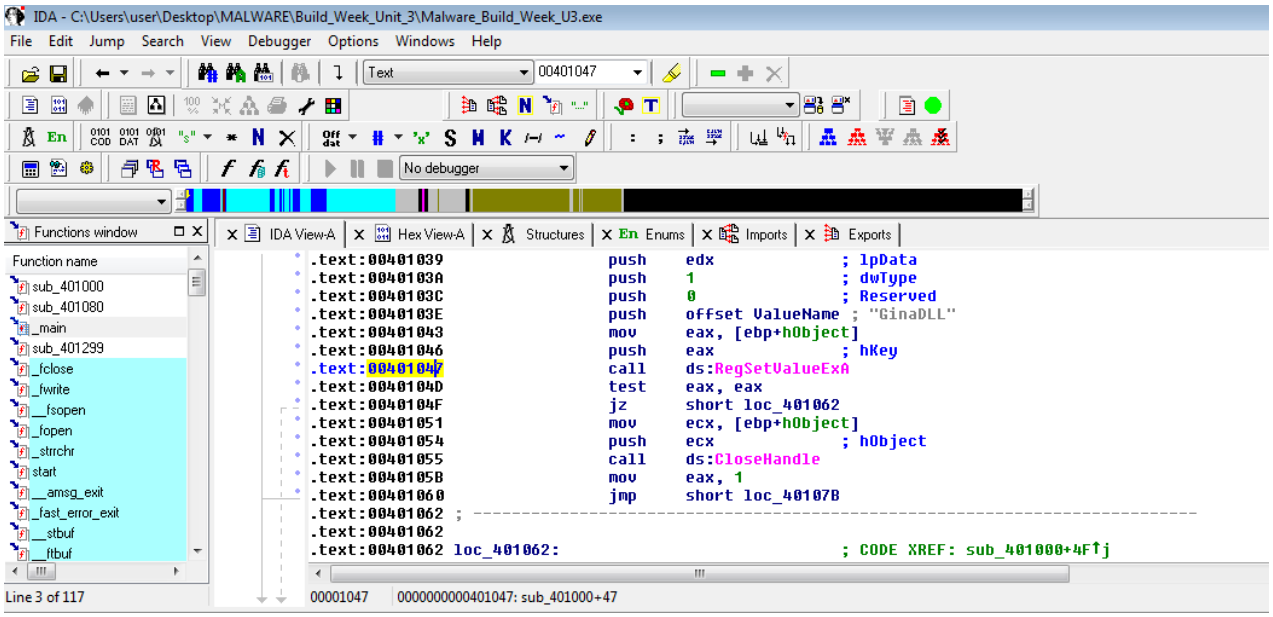
// Fai una cosa

} else {

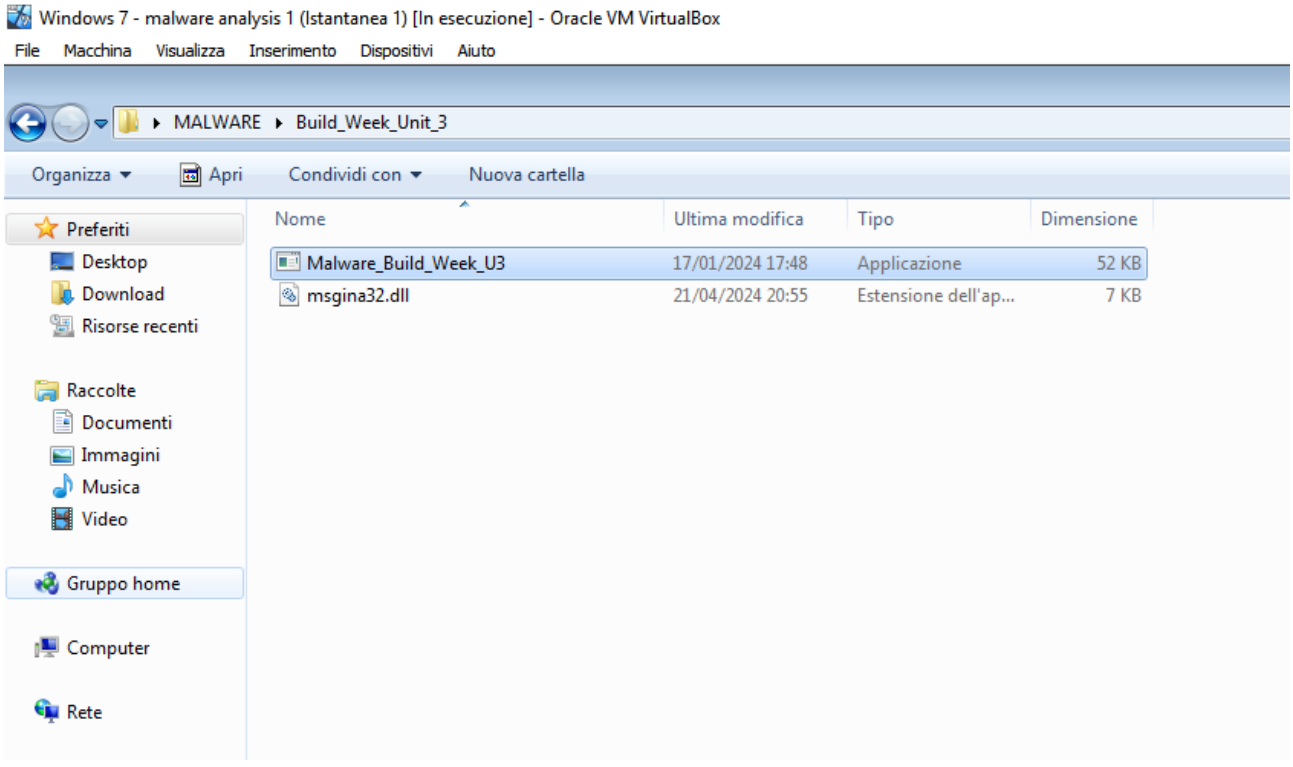
// Fai una cosa diversa

}

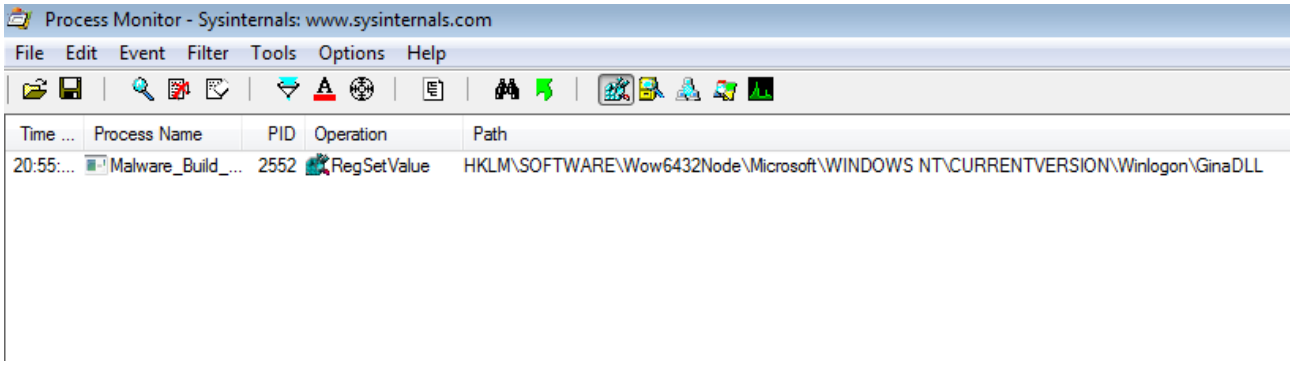
Da ultimo, per quanto riguarda questa prima parte, andiamo ad individuare il valore del parametro "ValueName", e troveremo una corrispondenza con "gina.dll". Possiamo pertanto valutare che la funzione stia impostando il valore di una chiave di registro denominata come sopra.



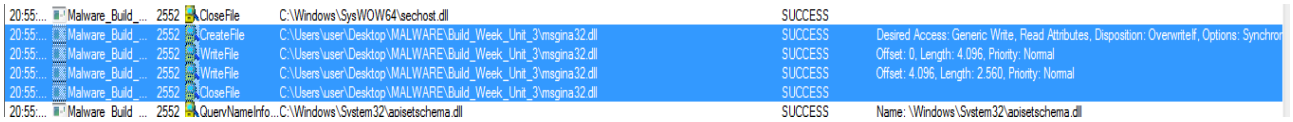
Passiamo ora all'analisi dinamica del malware. Notiamo subito, aprendo la cartella indicata nella traccia ed eseguendo il file malevolo, che viene creata l'estensione ***msgina32.dll.***
Msgina.dll è un file di libreria considerato di fondamentale importanza per il processo di login di Windows. Principalmente si occupa di visualizzare la schermata di login, con inserimento del nome utente e password per accedere al dispositivo, e conseguentemente gestire l'autenticazione dell'utente.



Avviamo Process Monitor, impostando il filtro “Process name is”, in maniera tale da analizzare il file tramite il suo stesso nome Malware_Build_Week_U3.exe.



Come prima cosa, notiamo che ritroviamo le chiavi di registro RegSetValue precedentemente incontrate con l’utilizzo di IDA Pro. A seguire, vediamo che viene effettuata una “create file”, ossia una chiamata di sistema che andrà a creare la msgina.dll incontrata poco sopra nella cartella del malware. Subito sotto, notiamo la “write file” che andrà letteralmente a scrivere e quindi inserire il codice malevolo nel programma, per poi concludere il suo lavoro con una “close file”.



Abbiamo pertanto effettuato l’analisi del malware da ogni prospettiva possibile. Le considerazioni che possiamo fare, al netto di quanto scoperto e analizzato, è che il malware intende creare e aprire una chiave di registro che incontriamo negli screen precedenti, “Winlogon”, di modo da potervi inserire il valore necessario per arrivare alla libreria malevola msgina32.dll che è stata creata dopo averlo eseguito. Come abbiamo visto, la libreria a cui ci riferiamo è di cruciale importanza per l’autenticazione degli utenti nel sistema Windows, quindi possiamo ipotizzare che lo scopo del malware fosse quello di agire per registrare le credenziali di accesso degli utenti.