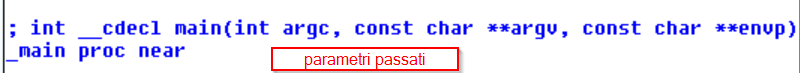
# W24D4PROGETTO

L’ obiettivo sarà quello di analizzare adeguatamente il malware contenuto nella Build\_Week\_Unit\_3.

# 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:**

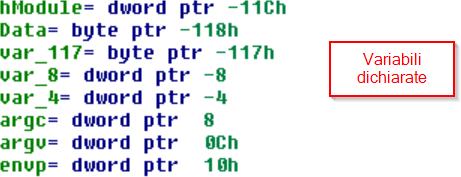
* **Quanti parametri sono passati alla funzione Main()?**



3 parametri in tutto vengono passati:l’intero argc, e le due costanti \*\*argv e \*\*envp.

Parametri passati

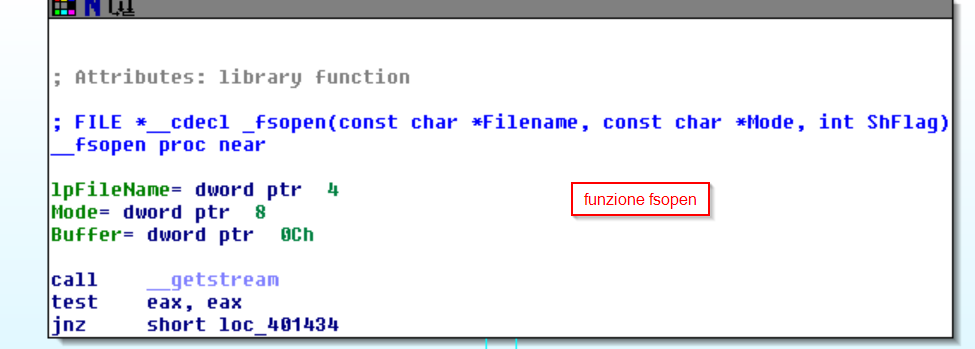
* **Quante variabili sono dichiarate all’interno della funzione Main()?**

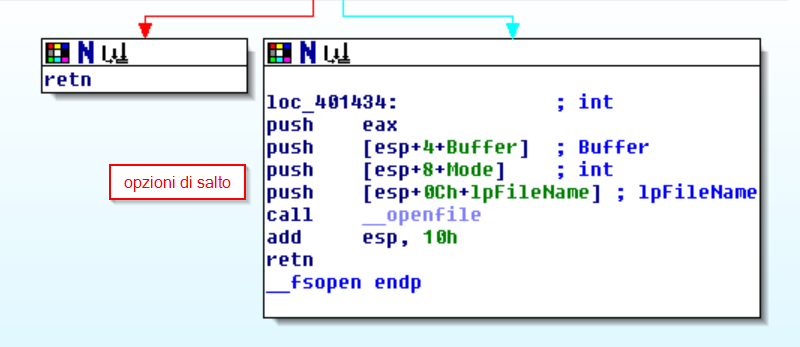


Variabili dichiarate

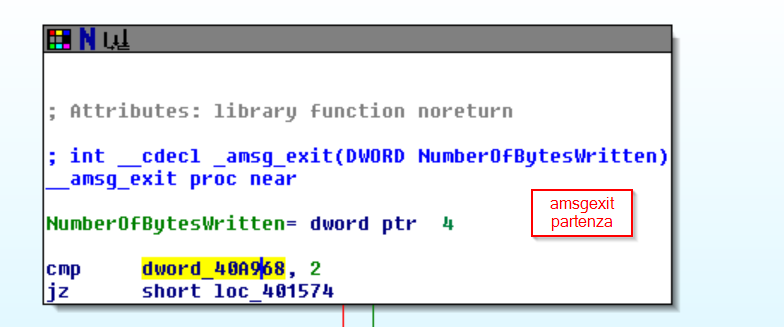
Le variabili sottostanti l’inizio della funzione main riportano, oltre ai parametri passati: hModule, Data, var\_117, var\_8 e var\_4

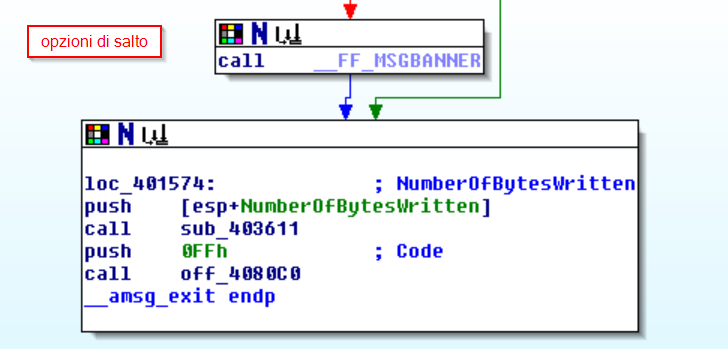
* **Quali sezioni sono presenti all’interno del file eseguibile? Descrivete brevemente almeno 2 di quelle identificate.**





La funzione fopen sembra essere una wrapper function per aprire un file in stream (lettura o scrittura).LpFileName è un puntatore a una stringa che rappresenta il nome del file. Mode specifica la modalità di apertura, mentre Buffer serve a memorizzare il file aperto. La funzione getstream verifica il valore restiuito in eax. Se eax è diverso da zero, verrà effettuato un salto alla locazione 401434, altrimenti Viene eseguito un ritorno.





OPZIONI DI SALTO

\_amsg\_exit è una funzione della libreria di runtime C/C++. Viene chiamata quando si verifica un errore di runtime all’interno di un programma. Il suo compito principale è generare un messaggio di errore e terminare l’applicazione.

La funzione parte con la verifica di word\_40A968, una variabile globale. Se dword\_40A968 è uguale a 2, salta a loc\_401574, altrimenti chiama FF\_MSGBANNER.

* **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.**



LIBRERIE IMPORTATE

Le funzioni nella lista “imports” sono state importate dalle librerie kernel32 e advapi32.

La prima libreria include funzioni riconducibili all’interazione con i servizi e i registri. RegSetValueExA, ad esempio, imposta dati e tipo di un

Valore all’interno di una chiave di registro del sistema, mentre

RegCreateKeyExA si occupa di creare la chiave specificata, a patto che quest’ultima non esista di già, in quel caso viene aperta. Per mezzo di queste funzioni, il malware potrebbe compiere diverse azioni dannose:

* Potrebbe utilizzare RegSetValueExA per creare o modificare valori all’interno delle chiavi di registro di avvio automatico, come

HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVer sion\Run. In questo modo otterrebbe la persistenza.

* Potrebbe modificare il comportamento del sistema, come ad esempio il browser predefinito,potrebbe indirizzare l’utente verso siti web malevoli o pubblicitari.
* Potrebbe utilizzare i valori modificati/creati per occultare la propria presenza o negare il corretto funzionamento degli strumenti di sicurezza.

La seconda libreria porta con sé funzioni per modificare i file e gestire la memoria.Tra le varie funzioni importate, prendiamo in considerazione: SizeOfResource, LockOfResource e

LoadOfResource:

* Il primo restituisce la dimensione di una risorsa

specificata.la si può utilizzare per determinare quanto spazio di memoria è necessario per archiviare una risorsa.

* LockOfResource: Restituisce un puntatore al primo byte dei dati associati a una risorsa. Dopo aver caricato una risorsa con LoadResource, è possibile utilizzarlo per accedere ai dati della risorsa.
* LoadResource, infine, può essere utilizzato per ottenere un puntatore al primo byte della risorsa specificata in memoria.

È possibile utilizzarla per caricare una risorsa da un file eseguibile e ottenere un handle per i dati associati alla risorsa.

LoadResource restituisce un handle di tipo HGLOBAL, ma non deve essere passato alle funzioni GlobalLock o GlobalFree. Per ottenere un puntatore ai dati della risorsa, è necessario chiamare LockResource. Per ottenere le dimensioni della risorsa, è possibile chiamare SizeofResource.

Ipotizzando dei possibili scenari, un malware potrebbe:

* Utilizzare LoadResource per accedere a risorse all’interno di un file eseguibile o di una libreria dinamica. Ad esempio, potrebbe cercare di estrarre informazioni sensibili come chiavi di

crittografia, password o dati di configurazione.

* Utilizzando LockResource, un malware potrebbe ottenere un puntatore ai dati di una risorsa. Successivamente, potrebbe

Sovrascrivere i dati con codice malevolo e iniettarlo nel processo in esecuzione. Questo potrebbe portare a comportamenti imprevisti o dannosi nel programma.

* Un malware potrebbe utilizzare SizeOfResource per determinare ladimensione di una risorsa. Successivamente, potrebbe allocare una grande quantità di memoria per risorse, esaurire le risorse

Del sistema e causare un DoS.

**Con riferimento al Malware in analisi, spiegare:**

* Lo scopo della funzione chiamata alla locazione di memoria 00401021:

La funzione RegCreateKeyExA crea tutte le chiavi mancanti nel percorso specificato, oppure le apre nel caso esistano già.Come specificato prima.

* Come vengono passati i parametri alla funzione alla locazione 00401021:
* pushebp:Questa istruzione mette il valore corrente del registro ebp nello stack.
* movebp,esp: Questa istruzione copia il valore dello stack pointer (esp) nel registro base del frame (ebp). In altre parole, ebp ora punta alla posizione corrente dello stack.
* pushecx: Mette il valore del registro ecx nello stack.Questo

potrebbe essereutilizzato per salvare temporaneamente il valore di ecx prima di modificarlo all’interno della funzione.

* push0:Mette il valore zero nello stack,corrispondente al parametro lpdwDisposition.
* leaeax,[ebp+hObject]:Calcola l’indirizzo della variabile hObject all’interno del frame dello stack e lo mette nel registro eax.
* pusheax:Mette il valore di eax(cheèl’indirizzodihObject)nello stack, corrispondente al parametro phkResult.
* push0:Mette il valore zero nello stack,corrispondent eal parametro lpSecurityAttributes.
* push0F003Fh:Mette il valore esadecimale 0F003Fh nello stack, corrispondente al parametro samDesired.
* push0:Mette il valore zero nello stack,corrispondente al parametro dwOptions.
* push0:Ancora un altro valore zero nello stack,corrispondente al parametro lpClass.
* pushoffsetSubKey:Mettel’indirizzo della stringa

“SOFTWARE\Microsoft\WindowsNT\CurrentVersion”nellostack.

* push80000002h:Mette il valore esadecimale 80000002h nello stack, corrispondente al parametro hkey.
* callds:RegCreateKeyExA: Chiama la funzione RegCreateKeyExA. parametri passati alla funzione sono quelli che abbiamo messo nello stack in precedenza.
* Che oggetto rappresenta il parametro alla locazione 00401017,Il significato delle istruzioni comprese tra gli indirizzi 00401027 e 00401029.

Il parametro alla locazione 00401017,ovvero

"SOFTWARE\Microsoft\Windows NT\CurrentVersion"si riferisce al registro di sistema di Windows. Questo percorso contiene

Informazioni sulla versione attuale di WindowsNT installata sul

sistema,tra cui informazioni di configurazione e altre impostazioni di sistema rilevanti.

Le istruzioni tra gli indirizzi 00401027 e 00401029sono le seguenti:

testeax,eax,questa istruzione effettua l’AND logico tra il registro eax e se stesso. Se il risultato è zero, imposta lo flag di zero a 1 (ZF),

altrimenti lo azzererà.

jz shortloc\_401032, questai struzione effettua un salto alla

locazione 401032 se lo zeroflag corrisponde ad una, altrimenti si procede con le istruzione successiva.

* Con riferimento all’ultimo quesito, tradurre il codice Assembly nel corrispondente costrutto C.Valutate ora lachiamata alla locazione 00401047, qual è il valore del parametro «ValueName»?
* Traduzione in C

# if(eax==0){

**gotoloc\_401032;**//permette di saltare ad un etichetta specifica all’interno della funzione

**}**

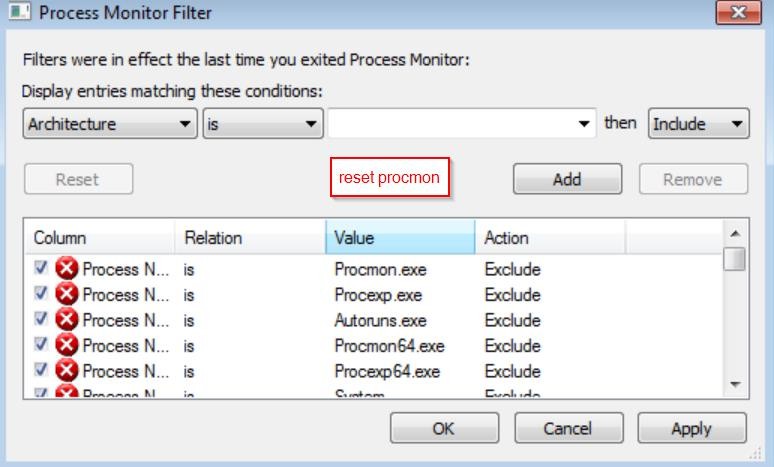
* Il valore del parametro ValueName,che verrà passato alla funzione RegSetValueExA è “GinaDLL”.

# Analisi dinamica

Preparate l’ambiente e di tool per l’esecuzione del Malware

(suggerimento: avviate principalmente ProcessMonitor 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**

****

Nella cartella dove è situato l'eseguibile del malware dopo l'esecuzione, notiamo la presenza di un file chiamato "msgina32.dll".

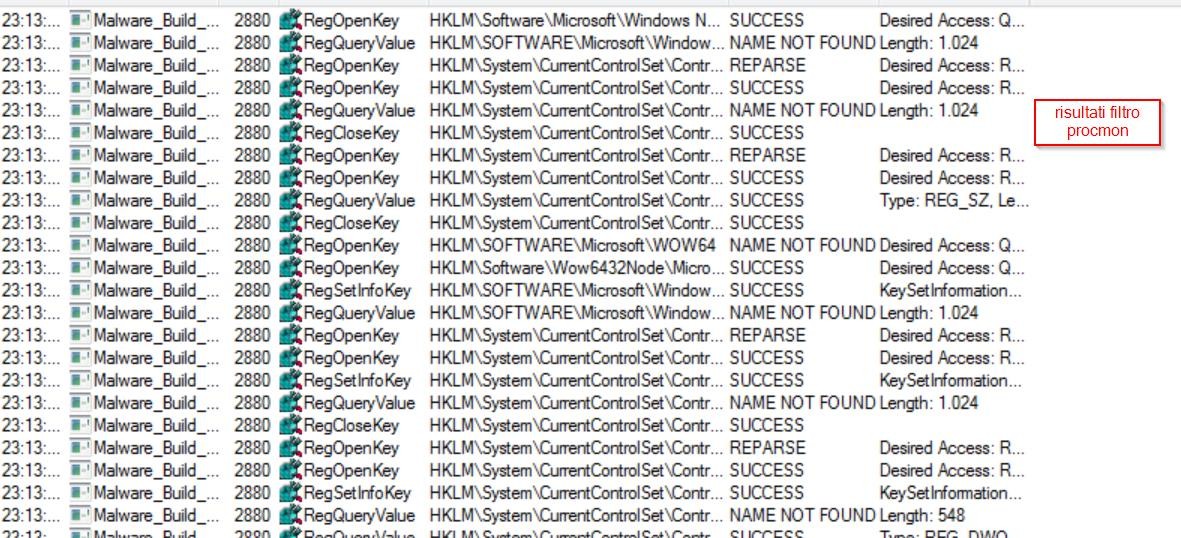
In base all’analisi statica precedente, possiamo dedurre che il

malware ha utilizzato la funzione RegSetValueExA per scrivere nel registro di sistema il valore"GinaDLL", e successivamente ha creato un file DLL chiamato "msgina32.dll".

Questo suggerisce fortemente che il malware stia cercando di sostituire o aggiungere un componente di autenticazione GINA (Graphical Identification and Authentication) nel sistema.

* **Analizzate ora i risultati di ProcessMonitor Filtrate includendo solamente l’attività sul registro di Windows.**

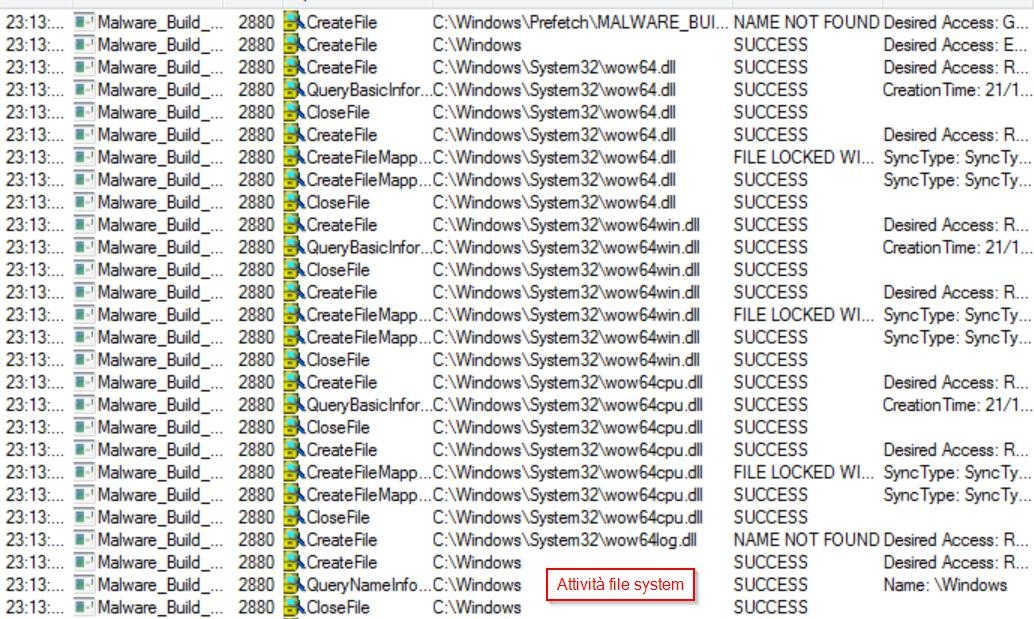




* **Qualechiavediregistrovienecreata?** msgina32.dll
* **Qualevalorevieneassociatoallachiavediregistrocreata?**

GinaDLL

* **Passate ora alla visualizzazione dell’attività sul file system.**



* Qualechiamatadisistemahamodificatoilcontenutodellacartella 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.

Come abbiamo detto prima, il malware ha utilizzato la funzione

RegSetValueExA per scrivere nel registro di sistema il valore"GinaDLL",e successivamente ha creato un file DLL chiamato "msgina32.dll".

Dopodiciò, osservando l’analisi dinamica del malware, si notano diverse operazioni di tipo QueryNameInformationFile, che hanno raccolto

Informazioni riguardo numerose librerie del sistema operativo,ad

esempio kernel32.dll, usato per la manipolazione dei file e la gestione della memoria, advapi32.dll, utile per interagire con i servizi e i registri oppure apisetschema.dll, utilizzato per la gestione dei file di definizione

delle api.

