Malware Analysis Project

Con riferimento alla cartella Malware_Build_Week_U3 presente sui file inviati da EPICODE

- *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.*

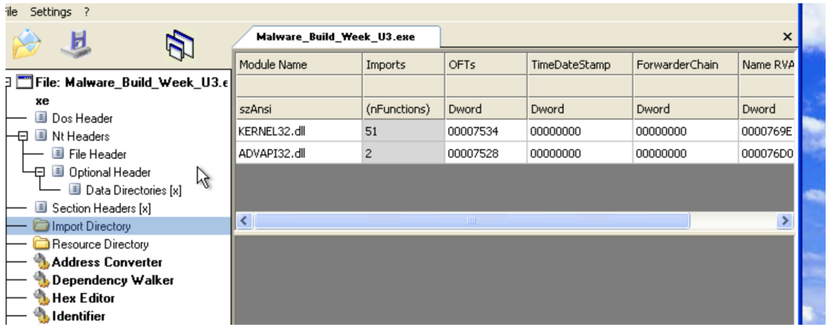
1) Per rispondere al primo e secondo punto ho cercato con il tool IDA Pro la funzione main tra le categorie functions che mi ha portato al indirizzo desiderato, di conseguenza sono visibili in figura le variabili dichiarate all interno della funzione e i parametri passati: rispettivamente 4 variabili e 3 parametri. Ricordando inoltre che le variabili sono riconoscibili dal valore negativo in esadecimale come conseguenza della creazione dello stack

```
.text:004011D0
.text:004011D0 ; int __cdecl main(int argc,const char **argv,const char *envp)
.text:004011D0 _main           proc near               ; CODE XREF: start+AF↓p
.text:004011D0
.text:004011D0 hModule         = dword ptr -11Ch
.text:004011D0 Data            = byte ptr -118h
.text:004011D0 var_8           = dword ptr -8
.text:004011D0 var_4           = dword ptr -4
.text:004011D0 argc            = dword ptr  8
.text:004011D0 argv            = dword ptr  0Ch
.text:004011D0 envp            = dword ptr  10h
.text:004011D0
```

2) Per rispondere al terzo punto ho riportato lo screenshot qua a fianco delle sezioni contenuti all'interno del malware. La sezione .text contiene le righe di codice (istruzioni) che la CPU eseguirà una volta avviato il software.
La sezione .rdata include informazioni circa le librerie e le funzioni importate ed esportate dall' eseguibile.
La sezione .data contiene i dati e le variabili globali del programma.
La sezione .rsrc include le risorse utilizzate dall'eseguibile come icone, imagini, menu o stringhe con non sono parte dell'eseguibile stesso. In alcuni casi, quando si ha a che fare con un dropper per esempio vi è contenuto un altro malware in questa sezione.

J3.e

Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword
.text	00005646	00001000	00006000	00001000	00000000	00000000
.rdata	000009AE	00007000	00001000	00007000	00000000	00000000
.data	00003EA8	00008000	00003000	00008000	00000000	00000000
.rsrc	00001A70	0000C000	00002000	0000B000	00000000	00000000



1)

word	word	Word	szAnsi
0007632	00007632	0295	SizeofResource
0007644	00007644	01D5	LockResource
0007654	00007654	01C7	LoadResource
0007622	00007622	02BB	VirtualAlloc
0007674	00007674	0124	GetModuleFileNameA

2)

Dword	Dword	Word	szAnsi
00007674	00007674	0124	GetModuleFileNameA
0000768A	0000768A	0126	GetModuleHandleA
00007612	00007612	00B6	FreeResource
00007664	00007664	00A3	FindResourceA
00007604	00007604	001B	CloseHandle

3)

	Dword	Word	szAnsi
	00007604	001B	CloseHandle
	000076DE	00CA	GetCommandLineA
	000076F0	0174	GetVersion
	000076FE	007D	ExitProcess
	0000770C	019F	HeapFree

4)

word	Word	szAnsi
007816	0115	GetFileType
007824	0150	GetStartupInfoA
007836	0109	GetEnvironmentVariableA
007850	0175	GetVersionExA
007860	019D	HeapDestroy

5)

Dword	Word	szAnsi
00007916	013E	GetProcAddress
00007928	01C2	LoadLibraryA
00007938	0261	SetEndOfFile
00007948	0218	ReadFile
00007954	01E4	MultiByteToWideChar

6)

Ts (IAT)	Hint	Name
Dword	Word	szAnsi
000076AC	0186	RegSetValueExA
000076BE	015F	RegCreateKeyExA

3) Per rispondere all'ultimo punto mi dilungherò un po:
Come si evince dal primo screenshot qua a fianco le uniche due librerie importate sono kernel32.dll e advapi32.dll che rispettivamente contengono, la prima, le funzioni principali per interagire con il sistema operativo (es: manipolazione file, gestione memoria) e la seconda, le funzioni per interagire con i servizi ed i registri del sistema operativo Microsoft.
Analizzando nel dettaglio le funzione importate dalle librerie possiamo trarre già delle ipotesi relative al tipo di malware che stiamo analizzando.
Le funzioni numerate dal 1 al 5 sono relative alla libreria kernel32 mentre la 6 sono le funzioni importate dalla libreria advapi32.
Possiamo notare che il malware utilizza 4 funzioni interessanti della libreria kernel32, ovvero -findresource, -loadresource, -lockresource, -sizeofresource che identificano la ricerca di una risorsa all'interno della sezione .rsrc, il caricamento della stessa nell'eseguibile, il blocco e il check della grandezza. Queste chiamate di funzioni possono identificare il malware come un dropper ovvero un tipo di malware che contiene al suo interno il vero malware, un po' come se all'inizio volesse passare in incognito senza farsi riconoscere.
Abbiamo altre chiamate di funzioni interessanti come -getstartupinfo, -getversionex e -loadlibrary che indicano richiesta di informazioni circa il sistema operativo la seconda, e informazioni riguardo all'avvio automatico la prima per ottenere la persistenza, mentre la terza è una funzione che richiama una libreria non elencata in fase di analisi perchè chiamata appunto a tempo di esecuzione (runtime)

- *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.

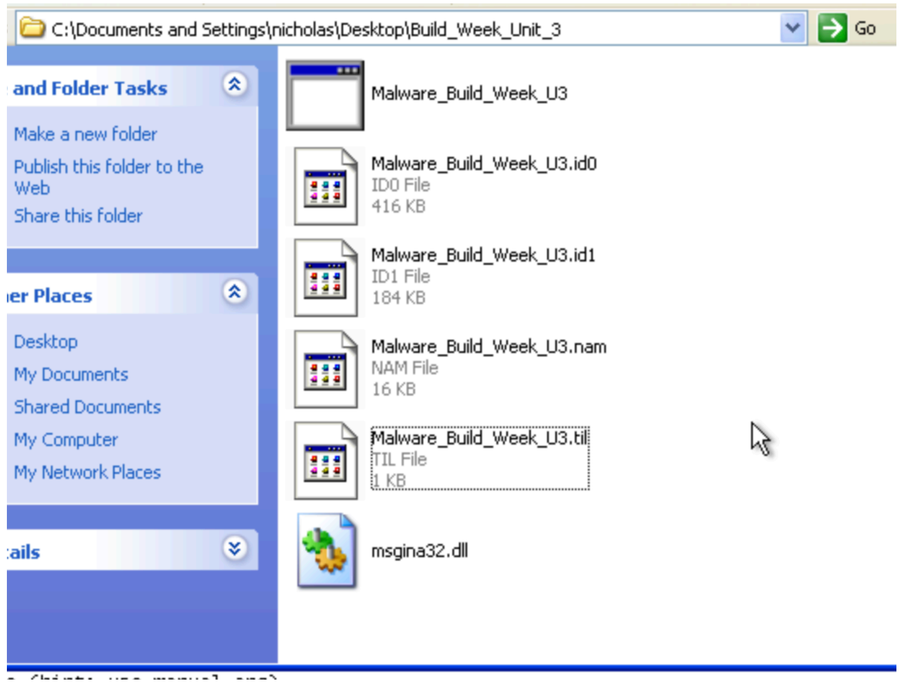
- *Valutate ora la chiamata alla locazione 00401047, qual è il valore del parametro «ValueName»?*

1) In risposta al primo punto, la funzione chiamata alla locazione di memoria 00401021 crea una nuova chiave di registro
2) I parametri vengono passati sullo stack tramite il 'push' che serviranno poi per essere utilizzati dalla funzione chiamata
3) Il tipo di oggetto che rappresenta (subkey) è il nome di una sottochiave aperta o creata da questa funzione. La sottochiave specificata deve essere una sottochiave della chiave identificata dal parametro *hKey.*
4) La prima istruzione, ovvero test eax,eax serve a valutare se il valore del registro eax è 0, in caso affermativo come si evince dalla seconda istruzione, se lo zero flag è 1 allora effettuerà il salto a quel determinato indirizzo.
   Il corrispondente costrutto C potrebbe essere, nel contesto di una chiamata di funzione: ' if eax=0 goto loc_401032 '
5) Il valore del parametro ValueName è GinaDLL quindi può farci sospettare che abbia creato la chiave di registro e impostata con il valore GinaDLL per interagire con il processo di autenticazione tramite Winlogon

```
text:00401015    push    0               ; Reserved
text:00401017    push    offset SubKey   ; "SOFTWARE\\Microsoft\\Window
text:0040101C    push    80000002h       ; hKey
text:00401021    call    ds:RegCreateKeyExA
text:00401027    test    eax, eax
text:00401029    jz      short loc_401032
text:0040102B    mov     eax. 1
```

```
.text:0040103A    push    1               ; dwType
.text:0040103C    push    0               ; Reserved
.text:0040103E    push    offset ValueName ; "GinaDLL"
.text:00401043    mov     eax, [ebp+hObject]
.text:00401046    push    eax             ; hKey
.text:00401047    call    ds:RegSetValueExA
.text:0040104D    test    eax, eax
.text:0040104F    jz      short loc_401062
```

- *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*



Sono stati creati alcuni file che potrebbero voler semplicemente depistare il reale funzionamento del malware, quello che invece interessa maggiormente e la creazione della libreria MSGina32.dll che correlata a ciò abbiamo visto in precedenza ci fornisce informazioni riguardo Winlogon e GINA che sono, insieme ai provider di rete le parti del modello di accesso interattivo. La procedura di accesso interattivo è in genere controllata da Winlogon, MSGina.dll e provider di rete. Per modificare la procedura di accesso interattivo, MSGina.dll può essere sostituito con una DLL GINA personalizzata cambiando i valori delle chiavi predefiniti.

- Winlogon: Parte del sistema operativo Windows che fornisce supporto interattivo per l'accesso. Winlogon è progettato intorno a un modello di accesso interattivo costituito da tre parti: l'eseguibile Winlogon, una libreria di collegamento dinamico di identificazione grafica e autenticazione (DLL) denominata GINA e qualsiasi numero di provider di rete.
- GINA: Libreria dll (Dynamic Link Library) di identificazione grafica e autenticazione. GINA è un componente DLL sostituibile caricato dall'eseguibile *Winlogon*. GINA implementa i criteri di autenticazione del modello di accesso interattivo ed è prevista l'esecuzione di tutte le interazioni utente di identificazione e autenticazione.

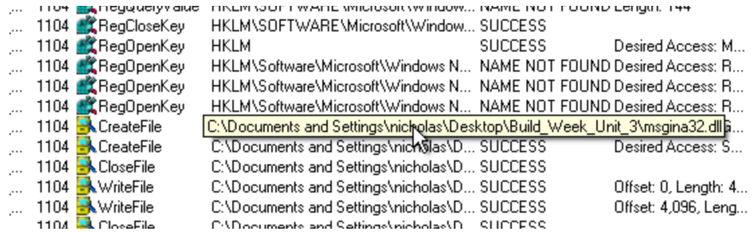
Riassumendo la libreria msgina32.dll è la libreria standard creta per poter interagire con il processo di autenticazione tramite Winlogon che è la subkey creata in precedenza “SOFTWARE\\Microsoft\\WindowsNT\\CurrentVersion\\Winlogon”, ricordando inoltre che I valori delle chiavi del Registro di sistema controllano la disponibilità o il comportamento di molte di queste funzionalità GINA standard.

Utilizzo di Procmon

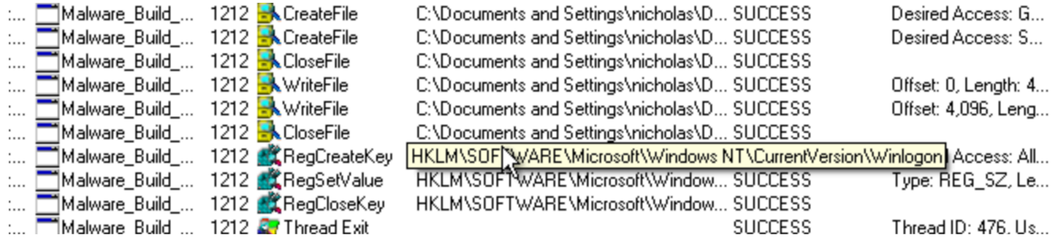
- Quale chiave di registro viene creata?
-Quale valore viene associato alla chiave di registro creata?.
-Quale chiamata di sistema ha modificato il contenuto della cartella dove è presente l'eseguibile del Malware?

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

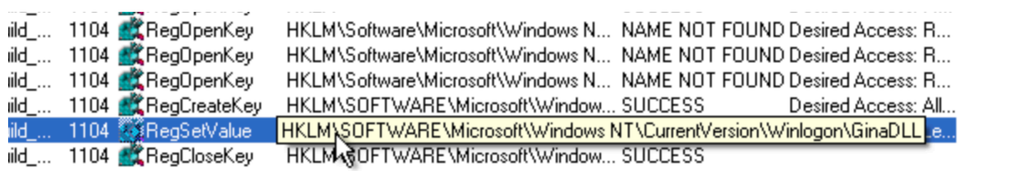
3)



1)



2)



1) La chiave di registro creata SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon è visibile nello screenshot .
2) Il valore che viene associata alla chiave di registro creata è la libreria GinaDLL, visibile nello screenshot 2.
3) Passando alla visualizzazione delle operazioni sul file system tramite la chiamata di sistema create file è stato creato un nuovo file, o in questo caso è stata importata una nuova libreria all'interno della cartella contenente il malware.

Abbiamo stabilito che si tratta di un dropper che attraverso una serie di funzioni importa una libreria che permette di interagire con il sistema di autenticazione di Windows, il suo scopo potrebbe essere quello di impadronirsi del dispositivo cambiando le configurazioni di accesso a proprio favore.