

Gelosa Matteo Progetto M6

L'analisi statica del Malware Build Week U3 è stata effettuata utilizzando IDA, esso converte il codice binario in codice assembly leggibile dall'uomo, facilitando l'analisi del funzionamento interno del malware.

Quanti parametri sono passati alla funzione main?

Nel codice è possibile vedere che i parametri passati alla funzione main sono **ARGC**, **ARGV**, e **ENVP**.

Sono parametri perchè hanno un offset positivo rispetto ad EPB e dichiarati nella funzione **MAIN**.

```
; int __cdecl main(int argc, const char **argv, const char **envp)
_main proc near

hModule= dword ptr -11Ch
Data= byte ptr -118h
var_117= byte ptr -117h
var_8= dword ptr -8
var_4= dword ptr -4
argc= dword ptr 8
argv= dword ptr 0Ch
envp= dword ptr 10h
```

Quante variabili sono dichiarate all'interno della funzione main?

Le variabili dichiarate nella funzione main risultano essere 5 , e sono:

hModule, **Data**, **Var_117**, **Var_8** e **Var_4**.

```
hModule= dword ptr -11Ch
Data= byte ptr -118h
var_117= byte ptr -117h
var_8= dword ptr -8
var_4= dword ptr -4
```





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

Le sezioni presenti sono:
























.text (contiene il codice eseguibile)

.data(contiene variabili)

.rdata(contiene informazioni delle funzioni utilizzate e delle librerie a cui il malware fa riferimento)

Name	Start	End	R	W	X	D	L	Align	Base	Type	Class
 .text	00401000	00407000	R	.	X	.	L	para	0001	public	CODE
 .idata	00407000	004070DC	R	.	.	.	L	para	0002	public	DATA
 .rdata	004070DC	00408000	R	.	.	.	L	para	0002	public	DATA
 .data	00408000	0040C000	R	W	.	.	L	para	0003	public	DATA

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.

x IDA View-A x Hex View-A x Structures x Enums x Imports x Exports				
Address	Ordinal	Name	Library	
 00407000		RegSetValueExA	ADVAPI32	
 00407004		RegCreateKeyExA	ADVAPI32	
 0040700C		SizeofResource	KERNEL32	
 00407010		LockResource	KERNEL32	
 00407014		LoadResource	KERNEL32	
 00407018		VirtualAlloc	KERNEL32	
 0040701C		GetModuleFileNameA	KERNEL32	
 00407020		GetModuleHandleA	KERNEL32	
 00407024		FreeResource	KERNEL32	
 00407028		FindResourceA	KERNEL32	
 0040702C		CloseHandle	KERNEL32	
 00407030		GetCommandLineA	KERNEL32	
 00407034		GetVersion	KERNEL32	
 00407038		ExitProcess	KERNEL32	
 0040703C		HeapFree	KERNEL32	
 00407040		GetLastError	KERNEL32	
 00407044		WriteFile	KERNEL32	
 00407048		TerminateProcess	KERNEL32	
 0040704C		GetCurrentProcess	KERNEL32	
 00407050		UnhandledExceptionFilter	KERNEL32	
 00407054		FreeEnvironmentStringsA	KERNEL32	
 00407058		FreeEnvironmentStringsW	KERNEL32	
 0040705C		WideCharToMultiByte	KERNEL32	

Le librerie importate dal malware sono **ADVAPI32** e **KERNEL32**.

Libreria ADVAPI32

ADVAPI32.dll è una libreria essenziale per la gestione dei registri in Windows è anche una libreria che i malware possono sfruttare per compiere azioni dannose come installare servizi nascosti, modificare le impostazioni di sicurezza del sistema . In questo caso come si può notare il malware utilizza la funzione **regcreatekeyexa**

per creare una nuova chiave di registro e **regsetvalueexa** per impostare il valore di una chiave di registro.

Libreria Kernel.32

Kernel32.dll fornisce una serie di funzioni per la gestione della memoria virtuale, comprese l'allocazione e la deallocazione di spazio di memoria, nonché la creazione e il controllo di processi, thread e sezioni di memoria.

Infatti in questo caso possiamo notare come il malware richiami la funzione di **Readfile** e **Writefile** .

Che oggetto rappresenta il parametro alla locazione 00401017?

Il parametro situato all'indirizzo 00401017 rappresenta il percorso nel registro di sistema che sarà utilizzato successivamente dalla funzione RegCreateKeyExA per creare una nuova chiave di registro.

Il significato delle istruzioni comprese tra gli indirizzi 00401027 e 00401029?

L'istruzione "test" esegue un'operazione logica AND tra il registro "eax" e se stesso. Questa operazione non altera il valore di "eax", ma serve a impostare il flag "ZF" (Zero Flag). Se "eax" ha un valore di zero, il flag "ZF" viene impostato a 1; altrimenti, viene impostato a 0. Una volta che il flag "ZF" è stato settato, l'istruzione successiva "jz" (jump if zero) potrebbe condurre a un salto condizionale all'indirizzo 401032 nel caso in cui il flag "ZF" sia stato precedentemente impostato a 1 dopo l'istruzione "test".

Con riferimento all'ultimo quesito, tradurre il codice Assembly nel corrispondente costruito C?

Può essere paragonato ad un IF in C.

```
main() {  
    if() ;  
}
```

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

```
.text:0040103E
.text:00401043
.text:00401046
.text:00401047

push    offset ValueName ; "GinaDLL"
mov     eax, [ebp+hObject]
push    eax               ; hKey
call    ds:RegSetValueExA
```

Il parametro "ValueName" contiene la stringa "GinaDLL", come indicato dall'istruzione "push offset ValueName". La chiamata alla posizione 00401047 alla funzione RegSetValueExA suggerisce che il malware potrebbe essere progettato per impostare un nuovo valore denominato "GinaDLL".

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 directory in cui si trova l'eseguibile, è stato generato un nuovo file denominato "msgina32.dll", che, presumibilmente, è correlato al parametro "ValueName" associato alla funzione RegSetValueExA rilevata durante l'analisi statica.

Quale chiave di registro viene creata e quale valore viene associato?

20:15:50,4077265	Malware_Build_Week_U3.exe	1948	RegSetInfoKey	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon
20:15:50,4077365	Malware_Build_Week_U3.exe	1948	RegQueryKey	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon
20:15:50,4077455	Malware_Build_Week_U3.exe	1948	RegSetValue	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL

Nel registro di sistema, è stato individuato un percorso associato al malware:

HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows

NT\CurrentVersion\Winlogon. Questa chiave è stata creata dal malware o aperta

nel caso fosse già presente. Il malware ha tentato di aggiungervi un nuovo valore

denominato GinaDLL. Attraverso l'analisi dei dettagli dell'operazione, si è scoperto

che questo valore fa riferimento al file msgina32.dll, creato dal malware al momento

del suo avvio.

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

0,4068505	Malware_Build_Week_U3.exe	1948	CreateFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll	SUCCESS
0,4073563	Malware_Build_Week_U3.exe	1948	WriteFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll	SUCCESS
0,4074072	Malware_Build_Week_U3.exe	1948	ReadFile	C:	SUCCESS
0,4074444	Malware_Build_Week_U3.exe	1948	WriteFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll	SUCCESS
0,4074649	Malware_Build_Week_U3.exe	1948	CloseFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll	SUCCESS

Dopo aver esaminato le attività del sistema di file, è stato rilevato che la chiamata CreateFile

è stata eseguita per creare il file msgina32.dll all'interno della stessa directory del malware.

Considerazione finale.

Le informazioni combinate dall'analisi statica e dinamica del Malware_Build_Week_U3 delineano un quadro complesso del suo comportamento. Questo malware, classificato come persistente, crea una sottochiave di registro nella HKEY locale, garantendo la sua presenza al riavvio del PC, senza dipendere da protocolli di rete. Nonostante non agisca come un backdoor, è considerato un rootkit, poiché potenzialmente modifica le chiavi di sistema per agevolare l'accesso a funzionalità sensibili. Inoltre, avvia numerosi processi per raccogliere informazioni dettagliate sulla macchina. Allo stesso tempo, esiste il rischio che possa danneggiare o infettare file presenti nella memoria, compromettendo così il funzionamento del sistema. Una caratteristica preoccupante è l'ipotesi che il malware miri a sostituire la DLL di sistema msgina.dll con il suo payload msgina32.dll durante il processo di login di Windows, il che potrebbe consentire di intercettare credenziali o eseguire operazioni dannose durante l'accesso