PROGETTO M6

Traccia Malware Analysis

Analisi statica

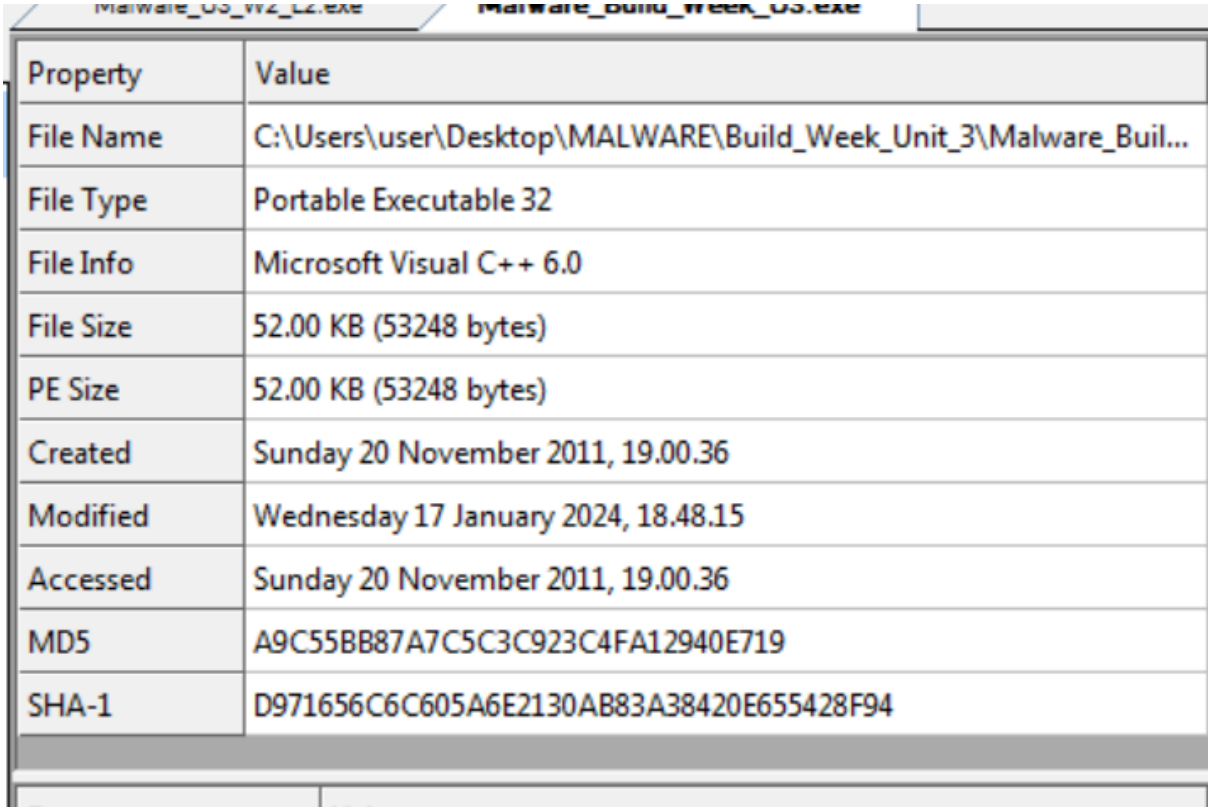
Il Malware da analizzare è nella cartella Build_Week_Unit_3 presente sul desktop della macchina virtuale dedicata. Analisi statica Con riferimento al file eseguibile **Malware_Build_Week_U3**, rispondere ai seguenti quesiti utilizzando i tool e le tecniche apprese nelle lezioni teoriche:

- 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:

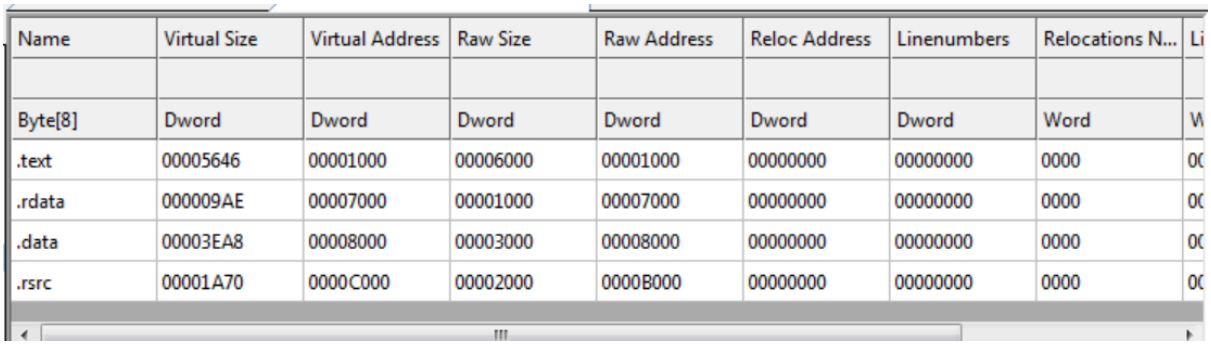
- Lo scopo della funzione chiamata alla locazione di memoria 00401021
- Come vengono passati i parametri alla funzione alla locazione 00401021;
- Che oggetto rappresenta il parametro alla locazione 00401017
- Il significato delle istruzioni comprese tra gli indirizzi 00401027 e 00401029.
- Con riferimento 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»?

Da CFF posso vedere informazioni più dettagliate in merito al malware.



Property	Value
File Name	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\Malware_Buil...
File Type	Portable Executable 32
File Info	Microsoft Visual C++ 6.0
File Size	52.00 KB (53248 bytes)
PE Size	52.00 KB (53248 bytes)
Created	Sunday 20 November 2011, 19.00.36
Modified	Wednesday 17 January 2024, 18.48.15
Accessed	Sunday 20 November 2011, 19.00.36
MD5	A9C55BB87A7C5C3C923C4FA12940E719
SHA-1	D971656C6C605A6E2130AB83A38420E655428F94

Sezioni del malware:



Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Li
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	W
.text	00005646	00001000	00006000	00001000	00000000	00000000	0000	00
.rdata	000009AE	00007000	00001000	00007000	00000000	00000000	0000	00
.data	00003EA8	00008000	00003000	00008000	00000000	00000000	0000	00
.rsrc	00001A70	0000C000	00002000	0000B000	00000000	00000000	0000	00

La seguenti sezioni:

- .text: contiene istruzioni di codice che la cpu esegue per far funzionare il programma
- .rdata: contiene i dati di sola lettura del codice
- .data: variabili accessibili da qualsiasi parte del codice
- .rsrc: contiene le risorse (immagini ecc)..

Le librerie che importa:

Malware_U3_W2_L2.exe | **Malware_Build_Week_U3.exe**

Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	51	00007534	00000000	00000000	0000769E	0000700C
ADVAPI32.dll	2	00007528	00000000	00000000	000076D0	00007000

Le librerie:

- KERNEL 32 permette al malware di creare, aprire e chiudere file, leggere e scrivere nella memoria, e interagire con il registro di sistema (quando si entra nel registro di sistema è una cosa grave, corrisponde a modificare le configurazioni del sistema operativo). La sua finalità è rubare i dati e file sensibili.Inoltre accedendo al registro di sistema, si può caricare del codice dannoso direttamente nella memoria bypassando i meccanismi di sicurezza del sistema.
- ADVAPI32 è una funzione che il malware va ad attivare in maniera tale da poter agire come amministratore di sistema. Può disattivare l'antivirus, il firewall (per far sì di non venire rilevato), può scaricare altri malware e controlla completamente il sistema, grazie ai privilegi di amministratore.

Utilizzando Ida continuiamo con l'analisi statica del codice del malware:

I parametri identificati sono 3 mentre le variabili sono 5.

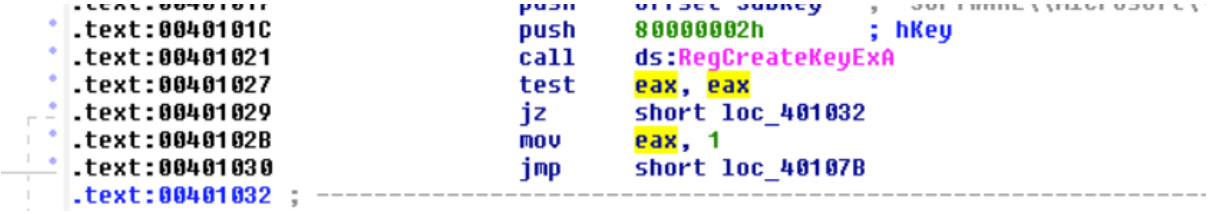
```
; Attributes: bp-based frame

; 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
```

Alla locazione di memoria 00401021 viene chiamata una funzione che crea una chiave di registro.

```
.text:00401011                push    0               ; dwOptions
.text:00401013                push    0               ; lpClass
.text:00401015                push    0               ; Reserved
.text:00401017                push    offset SubKey   ; "SOFTWARE\\Microsoft\\Windows NT\\Cur
.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
```

Possiamo vedere che viene creato uno stack per la funzione chiamata e i parametri vengono passati sullo stack con delle push.

Funzioni dalla locazione 00401027 a 00401029.
La test serve per confrontare i valori di registro se sono 0, se lo sono lo ZF viene settato a 1 e poi viene eseguita l'istruzione successiva che sarebbe la jz.



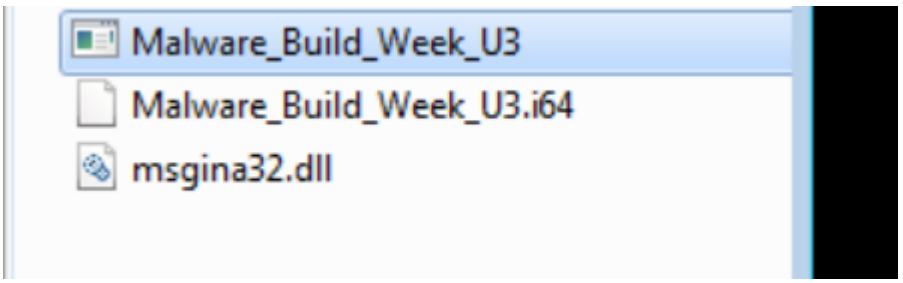
Traduzione delle istruzioni in linguaggio C:
Test eax, eax à if (var == 0)
Jz short loc_401032 à esegui codice contenuto tra le parentesi graffe
If (var == 0){ Istruzioni
}

Analisi dinamica

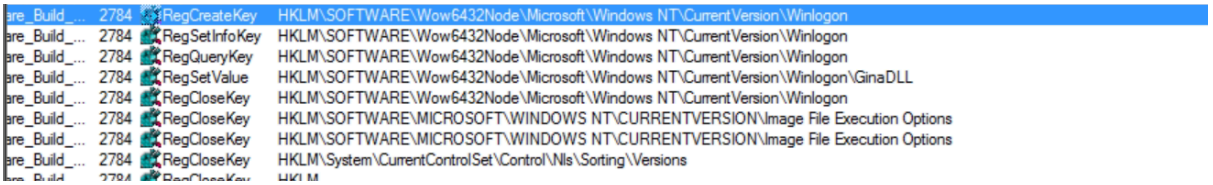
Preparate l'ambiente ed i tool per l'esecuzione del Malware (suggerimento: avviate principalmente Process Monitor ed assicurate di eliminare ogni filtro cliccando sul tasto «reset» quando richiesto in fase di avvio). Eseguite il Malware, facendo doppio click sull'icona dell'eseguibile.

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.

All'interno della cartella dell'eseguibile noto che c'è la libreria msgina32.dll (libreria che permette agli utenti di effettuare il login).



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.



Viene creata la chiave di registro Software\Wow6432Node\Microsoft\WindowsNT\CurrentVersion\Winlogon alla quale si va ad associare un valore col parametro Regsetvalue.
NB: Winlogon è responsabile dei login e logout degli utenti, il fatto che venga creata questa chiave di registro in questo malware ci permette di dire che alla login o logout di un utente parte il malware.

In merito alle attività del file system possiamo vedere che viene fatta una chiamata di sistema che crea un file all'interno della directory in cui ci è presente il malware che stiamo analizzando. Ecco perchè appena facevamo partire il malware vedevamo questalibreria.