

TRACCIA PARTE 1

Progetto di fine modulo 6 Malware analysis

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.

Per rispondere a questi quesiti andremo ad utilizzare il tool IDA PRO che è un potente strumento di disassemblaggio e reverse engineering utilizzato principalmente per analizzare il codice binario e ci permette di cominciare a fare una prima analisi statica.

```
; 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
```

Come possiamo notare dalla figura sopra alla funzione `Main()` passano 3 parametri con offset positivo rispetto al registro `EBP`:

1. `argc` che rappresenta il numero totale di argomenti passati alla funzione `main`, inclusa l'esecuzione del programma stesso.
2. `argv` che è un array di stringhe che contiene gli argomenti passati al programma. Ogni elemento dell'array è una stringa che rappresenta un argomento.
3. `envp`

Possiamo notare inoltre che nella funzione `Main()` sono dichiarate 5 variabili: `hModule`, `Data`, `var_117`, `var_8`, `var_4` con offset negativo rispetto al registro `EBP`.

Con il tool CFF Explorer si possono notare le sezioni di cui è composto il malware nella voce “section headers”:

Malware_Build_Week_U3.exe									
Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	Characteristics
000001D8	000001E0	000001E4	000001E8	000001EC	000001F0	000001F4	000001F8	000001FA	000001FC
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
.text	00005646	00001000	00006000	00001000	00000000	00000000	0000	0000	60000020
.rdata	000009AE	00007000	00001000	00007000	00000000	00000000	0000	0000	40000040
.data	00003EA8	00008000	00003000	00008000	00000000	00000000	0000	0000	C0000040
.rsrc	00001A70	0000C000	00002000	0000B000	00000000	00000000	0000	0000	40000040

La sessione .text è quella dove si trovano le istruzioni vere e proprie che la cpu esegue quando viene avviato l’elegibile;

La sessione .rdata contiene le funzioni e le librerie importate ed esportate;

La sessione .data contiene dati e variabili globali del programma;

La sessione .rsrc contiene le risorse usate dall’elegibile;

Quali librerie importa il Malware?

Per verificare ciò mi vado a servire del tool CFF Explorer:

Malware_Build_Week_U3.exe						
Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
0000769E	N/A	000074EC	000074F0	000074F4	000074F8	000074FC
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	51	00007534	00000000	00000000	0000769E	0000700C
ADVAPI32.dll	2	00007528	00000000	00000000	000076D0	00007000

Nella figura sopra possiamo quindi notare le 2 principali librerie importate dal malware KERNEL32.dll e ADVAPI32.dll e le rispettive funzioni.

La libreria KERNEL32.dll è piuttosto comune e contiene le funzioni principali per interagire con il sistema operativo, ad esempio:manipolazione dei file o la gestione di memoria.

Analizzando le molteplici funzioni all’ interno di questa libreria ho notato che potrebbe trattarsi di un malware di tipo dropper in quanto vengono utilizzate le APIs

FindResource() LoadResource() LockResource() SizeOfResource();

Queste APIs permettono di localizzare all’interno della sezione «risorse» il malware da estrarre, e successivamente da caricare in memoria per l’esecuzione o da salvare sul disco per esecuzione futura.

La libreria ADVAPI32.dll contiene funzioni per interagire con i servizi ed i registri del sistema operativo Microsoft.

Analizzando le funzioni all'interno di questa libreria che sono RegCreateKeyExA e RegSetValueExA possiamo ipotizzare che il malware cerchi di ottenere la persistenza modificando una chiave di registro.

Un malware che cerca la persistenza, sta cercando di rimanere attivo e nascosto nel sistema a lungo termine, ad esempio aggiungendo voci al registro di sistema, creando attività pianificate o utilizzando altre tecniche per avviarsi automaticamente e evitare la rimozione.

TRACCIA PARTE 2

Con riferimento al Malware in analisi, spiegare:

1. Lo scopo della funzione chiamata alla locazione di memoria 00401021
2. Come vengono passati i parametri alla funzione alla locazione 00401021;
3. Che oggetto rappresenta il parametro alla locazione 00401017
4. Il significato delle istruzioni comprese tra gli indirizzi 00401027 e 00401029.
5. Con riferimento all'ultimo quesito, tradurre il codice Assembly nel corrispondente costruito C.
6. Valutate ora la chiamata alla locazione 00401047, qual è il valore del parametro «ValueName»?

RISPOSTE

1. Alla locazione di memoria 00401021 troviamo la call alla funzione RegCreateKeyA. E' una chiamata a una funzione dell'API di Windows utilizzata per creare o aprire una chiave del Registro di sistema. Questa funzione può essere utilizzata da programmi per accedere e modificare le informazioni nel Registro di sistema, come le impostazioni dell'applicazione, le preferenze dell'utente o le configurazioni del sistema.

2. I parametri vengono passati tramite delle push alla funzione alla locazione 00401021

```
.text:00401009      push     eax                ; phkResult
.text:0040100A      push     0                 ; lpSecurityAttributes
.text:0040100C      push     0F003Fh           ; samDesired
.text:00401011      push     0                 ; dwOptions
.text:00401013      push     0                 ; lpClass
.text:00401015      push     0                 ; Reserved
.text:00401017      push     offset SubKey      ; "SOFTWARE\\Microsoft\\Windows NT\\CurrentVe"...
.text:0040101C      push     80000002h          ; hKey
.text:00401021      call    ds:RegCreateKeyExA
```

3. Alla locazione 00401017 viene passato il seguente oggetto

```
.text:00401027      test     eax, eax          SubKey      db 'SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Winlogon',0
.text:00401029      jz       short loc_401032  ; DATA XREF: sub_401000+17To
```

- 4.

```
* .text:00401027      test     eax, eax
* .text:00401029      jz       short loc_401032
```

Queste istruzioni stanno verificando se il valore contenuto nel registro EAX è uguale a zero. Se è zero (Z), il salto condizionale (jz) verrà eseguito, portando il flusso di esecuzione del programma all'indirizzo loc_401032. Altrimenti, se il valore in EAX non è zero, il flusso di esecuzione proseguirà normalmente senza saltare. In breve, queste istruzioni stanno controllando se EAX è zero e agiscono di conseguenza.

5. Costrutto C riferito al quesito precedente:

```
if (eax == 0) {
    // Salto a loc_401032
} else {
    // Continua l'esecuzione delle istruzioni successive
}
```

6. Questa stringa di codice chiama la funzione RegSetValueExA nel segmento di dati specificato (indicato da ds). Questa funzione è utilizzata per impostare il valore di una voce di registro in Windows. Il prefisso ds: indica il segmento di dati in cui si trova la funzione, che è un modo per specificare l'area della memoria in cui è allocata la funzione o i dati ad essa correlati. Inoltre, call è un'istruzione di controllo di flusso che trasferisce l'esecuzione al punto di ingresso della funzione specificata. Il valore del parametro ValueName è GinaDLL.

```
.text:00401032 loc_401032:                                ; CODE XREF: sub_401000+29↑j
.text:00401032      mov     ecx, [ebp+cbData]
.text:00401035      push    ecx                ; cbData
.text:00401036      mov     edx, [ebp+lpData]
.text:00401039      push    edx                ; lpData
.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
```

TRACCIA PARTE 3 ANALISI DINAMICA

Utilizzando il tool Process Monitor, md5deep e virus total andiamo a rispondere alle seguenti domande.

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.

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

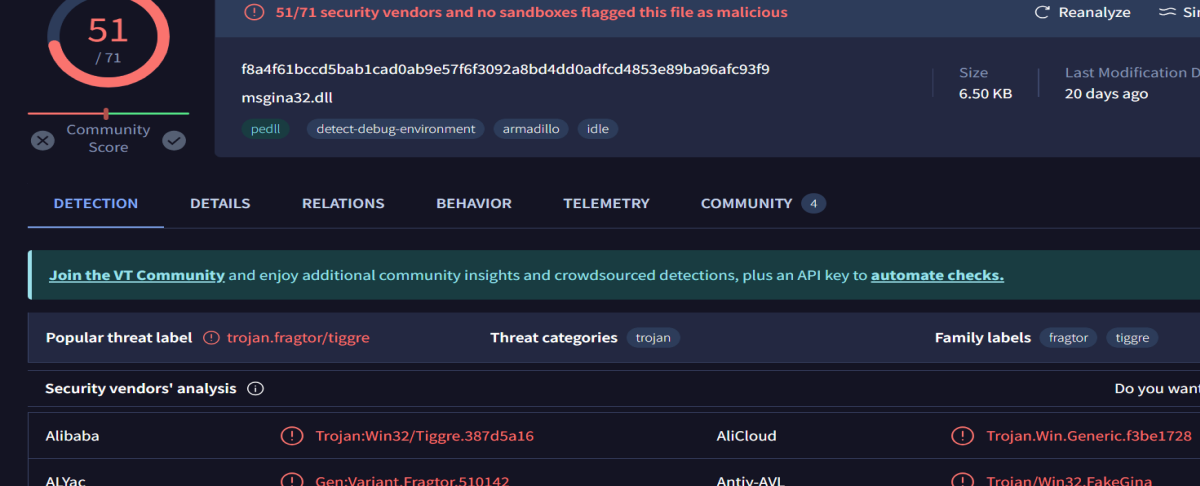
Dopo aver eseguito il malware questo ha creato all' interno della cartella dove è situato nel file system un file eseguibile “msgina32.dll”

Msgina32.dll è una libreria dinamica (DLL) presente nei sistemi operativi Windows. Questa è responsabile della gestione del processo di Identificazione e Autenticazione Grafica in Windows. GINA è un componente che gestisce l'autenticazione dell'utente durante il processo di accesso a Windows. Interagisce con l'interfaccia utente per richiedere le credenziali e verificarle nel database di sicurezza del sistema.

A questo punto andiamo a verificare che il file sia malevolo calcolando l'hash con il tool md5deep utilizzato dal prompt di Windows;

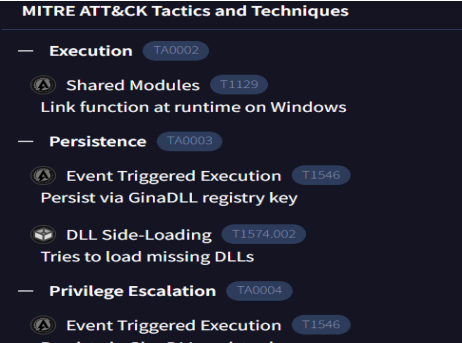
```
C:\Users\user\Desktop\Software Malware analysis\md5deep-4.3>cd md5deep-4.3
C:\Users\user\Desktop\Software Malware analysis\md5deep-4.3\md5deep-4.3>md5deep
"C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll"
md5deep: WARNING: You are running a 32-bit program on a 64-bit system.
md5deep: You probably want to use the 64-bit version of this program.
7ce4f799946f0fa44e5b2b5e6a702f27 C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll
C:\Users\user\Desktop\Software Malware analysis\md5deep-4.3\md5deep-4.3>
```

Andiamo ora ad inserire l'hash del file su Virus Total per un riscontro;



The screenshot shows the VirusTotal analysis interface for the file `msgina32.dll`. At the top, a red circle indicates a score of 51/71. A message states: "51/71 security vendors and no sandboxes flagged this file as malicious". The file's SHA-256 hash is `f8a4f61bccd5bab1cad0ab9e57f6f3092a8bd4dd0adfcc4853e89ba96afc93f9`, with a size of 6.50 KB and a last modification date of 20 days ago. The file is categorized as `trojan.fragtor/tiggre`. The "Security vendors' analysis" section shows detections from Alibaba (Trojan:Win32/Tiggre.387d5a16), AliCloud (Trojan.Win.Generic.f3be1728), ALYac (Gen:Variant.Fragtor.510142), and Antiy-AVL (Trojan/Win32.FakeGina).

Vendor	Detection
Alibaba	Trojan:Win32/Tiggre.387d5a16
AliCloud	Trojan.Win.Generic.f3be1728
ALYac	Gen:Variant.Fragtor.510142
Antiy-AVL	Trojan/Win32.FakeGina



The screenshot displays the MITRE ATT&CK Tactics and Techniques for the file `msgina32.dll`. The tactics and techniques listed are:

- Execution** (TA0002)
 - Shared Modules (T1129): Link function at runtime on Windows
- Persistence** (TA0003)
 - Event Triggered Execution (T1546): Persist via GinaDLL registry key
 - DLL Side-Loading (T1574.002): Tries to load missing DLLs
- Privilege Escalation** (TA0004)
 - Event Triggered Execution (T1546): Persist via GinaDLL registry key

Da questo riscontro possiamo dire dunque che si tratta di file malevolo che riguarda la categoria dei Trojan e che ottiene la persistenza tramite il file creato al suo avvio “msgina.dll”

La chiamata di sistema che ha modificato il contenuto dov'è presente il malware è descritta nella figura seguente.

Malware_Build_Week_U3.exe	1144	CreateFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll
Malware_Build_Week_U3.exe	1144	WriteFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll
Malware_Build_Week_U3.exe	1144	WriteFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll
Malware_Build_Week_U3.exe	1144	CloseFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll

Il malware ha creato il file ed ha scritto all'interno del file.

Aprendolo dal file system possiamo vedere tramite CFF le librerie e le rispettive funzioni che sono state importate al suo interno.

msgina32.dll						
Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
00000E24	N/A	00000C7C	00000C80	00000C84	00000C88	00000C8C
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	12	000020F0	00000000	00000000	00002224	00002010
MSVCRT.dll	11	00002124	00000000	00000000	00002288	00002044
ADVAPI32.dll	3	000020E0	00000000	00000000	000022F2	00002000
USER32.dll	1	00002154	00000000	00000000	0000230C	00002074

TRACCIA PARTE 4 ANALISI DINAMICA

Filtrate includendo solamente l'attività sul registro di Windows.

-Quale chiave di registro viene creata?

-Quale valore viene associato alla chiave di registro creata?

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

Sempre utilizzando Procmon possiamo notare che il Malware prova a creare la chiave di registro con RegCreateKey. L'operazione è eseguita con successo e veniamo a conoscenza che la chiave di registro esiste già, quindi è stata aperta anziché creata nuovamente.

Malware_Build_Week_U3.exe	1144	RegCreateKey	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	Desired Access: All Access, Disposition: REG_OPENED_EXISTING_KEY
Malware_Build_Week_U3.exe	1144	RegSetInfoKey	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	KeySetInformationClass: KeySetHandleTagsInformation, Length: 0
Malware_Build_Week_U3.exe	1144	RegQueryValue	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	Query: HandleTags, HandleTags: 0x400
Malware_Build_Week_U3.exe	1144	RegSetValue	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL	ACCESS DENIED	Type: REG_SZ, Length: 520, Data: C:\Users\user\Desktop\MALWARE\Bui

Dall'analisi statica e dinamica eseguite si può concludere di avere a che fare con un malware della famiglia dei dropper.

Il programma in questione manipola le chiavi di registro di Windows per aggiungere una nuova chiave e impostare un valore che punta al file "msgina32.dll", un possibile trojan. Questo file viene estratto dal dropper e copiato nella directory del file eseguibile. Il trojan si maschera come un componente legittimo di Windows, ma una volta eseguito, può avviare azioni dannose, inclusa la comunicazione con domini esterni.

LEONARDO MARGHERI