M A L W A R E
A N A L Y S I S

INDICE

TRACCIA 1

Con riferimento al file eseguibile Malware_Build_Week_U3, rispondere ai seguenti quesiti utilizzando i tool e le tecniche apprese nelle lezioni teoriche:
1.Quanti parametri sono passati alla funzione Main()?
2.Quante variabili sono dichiarate all'interno della funzione Main()?
3.Quali sezioni sono presenti all'interno del file eseguibile? Descrivete brevemente almeno 2 di quelle identificate
4.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.

TRACCIA 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 costrutto C.
6.Valutate ora la chiamata alla locazione 00401047, qual è il valore del parametro «ValueName»?

TRACCIA 1

Per ricavare informazioni su variabili locali e parametri della funzione main (), ida pro (interactive disassembler professional) è lo strumento avanzato di reverse engineering software che offre capacità di disassemblaggio, debugging e analisi statica.

Variabili locali: le variabili locali in una funzione assembly sono tipicamente allocate nello stack. Questo è spesso fatto attraverso istruzioni di tipo push all'i-nizio di una funzione o con un'istruzione sub che aumenta il puntatore dello stack (sp) per creare spazio.

Parametri: solitamente i parametri sono passati ai regi-stri o attraverso l'uso dello stack prima della chiama-ta della funzione. I commenti nel codice possono forni-re indicazioni su dove e quanti parametri sono passati. Nell'assembly, le etichette che iniziano con var_ tendono a indicare variabili locali, mentre quelle che iniziano con arg_ indicano argomenti o parametri passati alla funzione.

Valori offset: gli offset (come var_54h) sono utilizzati per accedere a dati specifici sullo stack. Gli offset negativi rispetto all'indirizzo base del frame della funzione (ad esempio ebp su x86) di solito indicano variabili locali, mentre gli offset positivi indicano parametri.

1. Parametri nella funzione Main()

All'interno della funzione Main() sono presenti tre parametri identificati dall'offset positivo rispetto ad EBP:
argc
Variabile utilizzata per rappresentare il numero di argomenti passati a un programma da riga di comando quando viene eseguito.
argv
Array di puntatori a stringhe utilizzato per contenere gli argomenti passati al programma da riga di comando quando viene eseguito.
envp
Array di puntatori a stringhe che contiene le variabili d'ambiente.

```
.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
```

2. Variabili nella funzione Main()

All'interno della funzione Main() sono presenti quattro variabili identificate dall'offset negativo rispetto ad EBP:
hModule
Variabile globale di tipo HMODULE che contiene l'handle del modulo di esecuzione corrente.
Data
Variabile utilizzata come buffer di dati
var_8, var_4
Variabili locali

```
.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
```

3. Sezioni presenti all’interno dell’eseguibile

All’interno dell’eseguibile sono presenti quattro sezioni:

.text

Contiene le istruzioni che la CPU eseguirà una volta che il software sarà avviato.

.rdata

Include le informazioni circa le librerie e le funzioni importate ed esportate dall'eseguibile.

.data

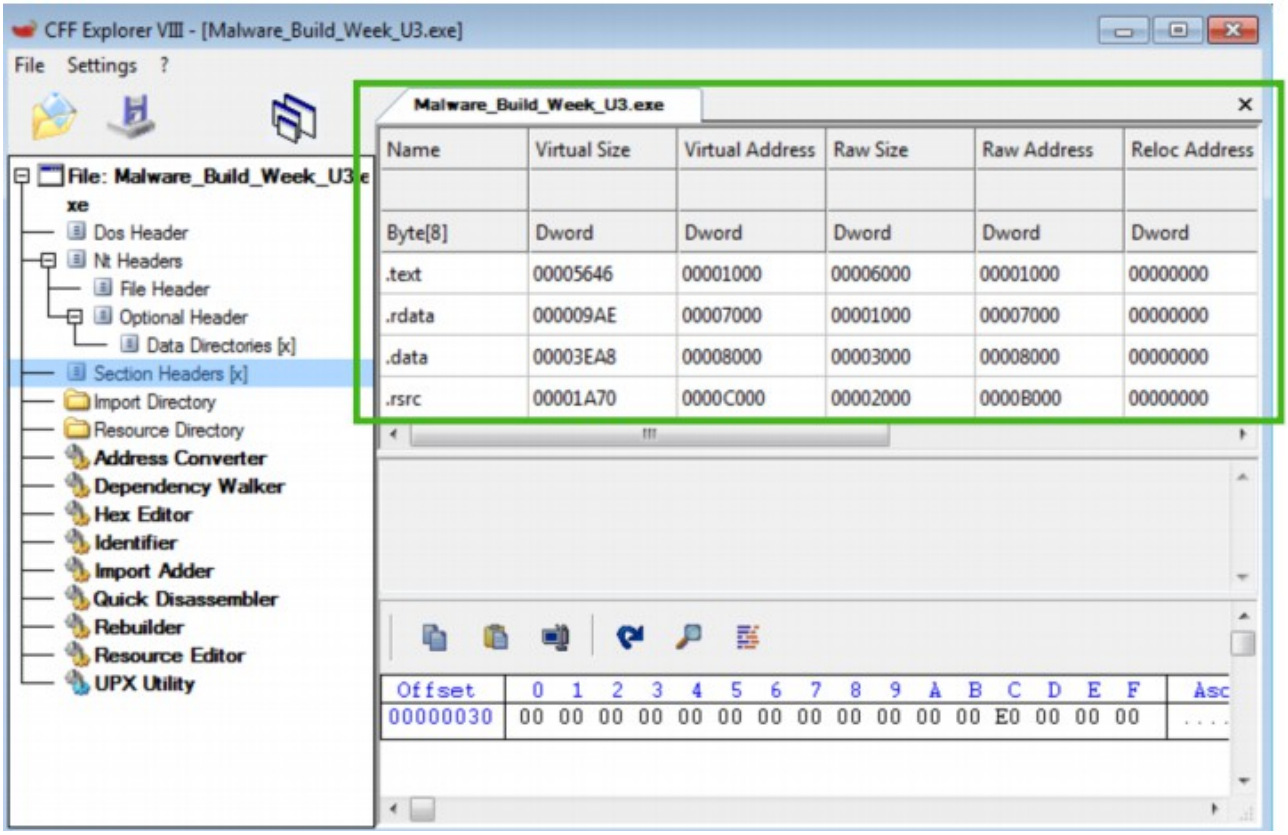
Contiene dati e variabili globali del programma eseguibile, che devono essere disponibili da qualsiasi parte del programma.

.rsrc

Include le risorse utilizzate dall'eseguibile come ad esempio icone, immagini, menu e stringhe che non sono parte dell'eseguibile stesso.

Sections

Name	Virtual Address	Virtual Size	Raw Size	Entropy	MD5	Chi2
.text	4096	22086	24576	6.23	6bb361ab84e6ea32f545b12825db9c07	304301.84
.rdata	28672	2478	4096	3.77	23fde5162e5b17a6e440a468b942c3b8	290048.5
.data	32768	16040	12288	0.6	e433b4c400efc11a593220e77ab72779	2826623.75
.rsrc	49152	6768	8192	4.15	9d561586eeb5ecda6c3214cd6a35d6f3	573983.75



4. Librerie importate dal malware

Il malware importa due librerie:

Kernel32.dll

Libreria che contiene le funzioni principali per interagire col sistema operativo, come per esempio la manipolazione di file e la gestione della memoria.

Advapi32.dll

Libreria che contiene le funzioni per interagire con i registri e i servizi del sistema operativo Microsoft.

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

Applicazione dell'analisi statica basica ed avanzata.

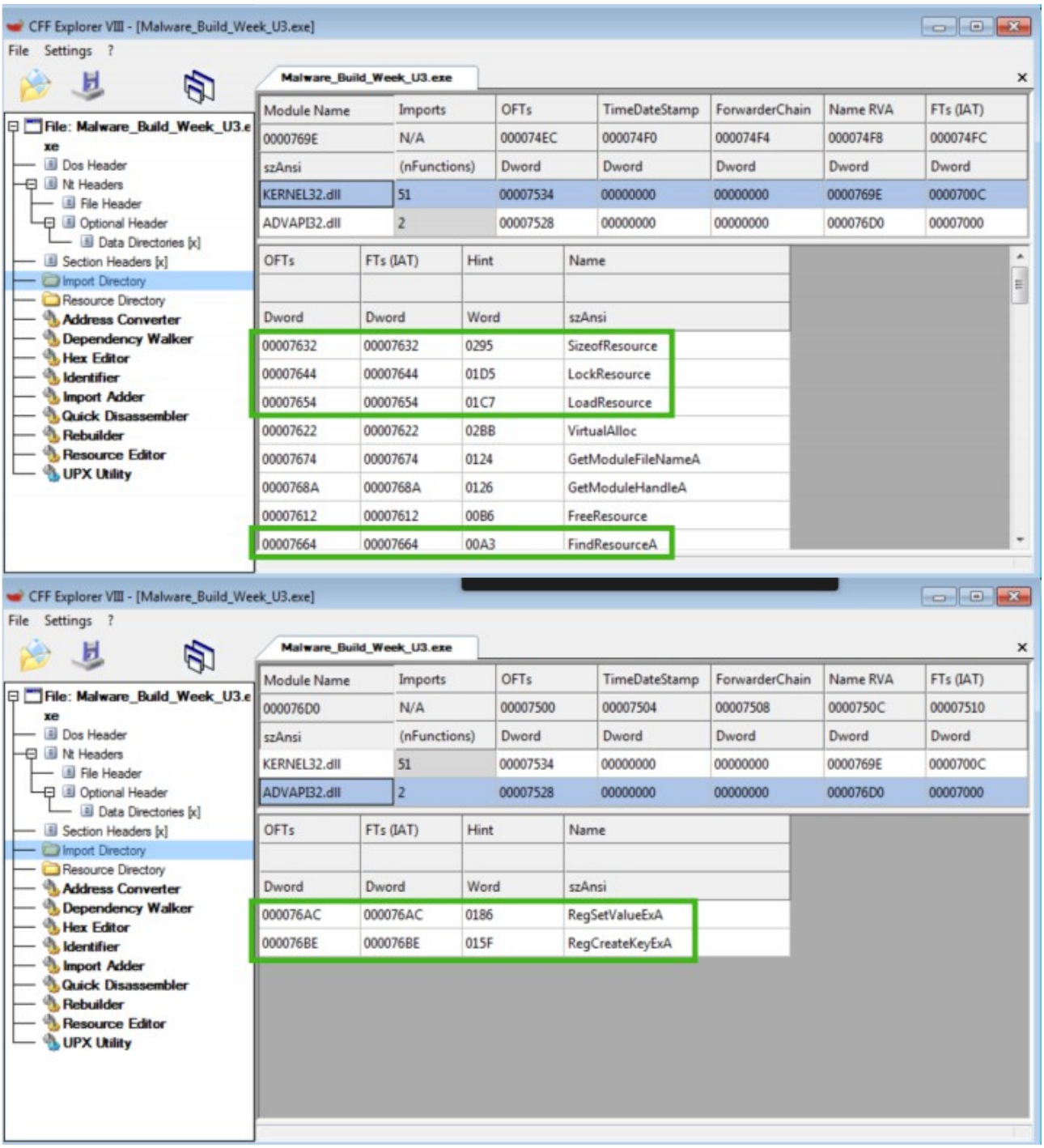
L'analisi statica si riferisce all'ispezione del codice sorgente o del codice binario di un programma (in questo caso, un malware) per identificarne la funzionalità, le caratteristiche e le potenziali minacce senza eseguirlo, questo approccio si contrappone all'analisi dinamica, dove il codice viene eseguito in un ambiente controllato (sandbox) per osservare il suo comportamento.

L’analisi statica basica consiste nell’esaminare un eseguibile senza vedere le istruzioni che lo compongono e la sua funzione è confermare se un dato file è malevolo e fornire informazioni generiche circa le sue funzionalità.

L'analisi statica avanzata presuppone la conoscenza dei fondamenti di «reverse-engineering» al fine di identificare il comportamento di un malware a partire dall’analisi delle istruzioni che lo compongono. Questo passaggio è essenziale per capire esattamente cosa fa il malware a livello di istruzioni della cpu.

Si possono inoltre estrarre stringhe di testo, url, chiavi di cifratura, e altre risorse dal codice del malware, che possono indicare il suo comportamento o intento e se ne può esaminare il codice relativo alla rete per comprendere come il malware comunica.
Nell'estrare il malware con l'hash md5deep e controllando su virustotal si riscontra di fatto essre un virus noto di tipo trojan, progettato per colpire la macchina intel 386 e processori. sucessivi/compatibili.
Ad evidenziare ciò, sono presenti nelle funzioni di libreria del Kernel32.dll troviamo:
FindResourceA()
LoadResource()
SizeofResource()
Generalmente utilizzate per recuperare altri file malevoli dalla sezione delle risorse “.rsrc”.
Si può anche ipotizzare che il malware ottiene persistenza modificando le chiavi di registro con le funzioni della libreria Advapi32.dll:

RegCreateKeyExA()
RegSetValueExA()


CFF Explorer VIII - [Malware_Build_Week_U3.exe]
File Settings ?
Malware_Build_Week_U3.exe
File: Malware_Build_Week_U3.exe
Dos Header
Nt Headers
File Header
Optional Header
Data Directories [x]
Section Headers [x]
Import Directory
Resource Directory
Address Converter
Dependency Walker
Hex Editor
Identifier
Import Adder
Quick Disassembler
Rebuilder
Resource Editor
UPX Utility
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
OFTs | FTs (IAT) | Hint | Name
Dword | Dword | Word | szAnsi
00007632 | 00007632 | 0295 | SizeofResource
00007644 | 00007644 | 01D5 | LockResource
00007654 | 00007654 | 01C7 | LoadResource
00007622 | 00007622 | 02BB | VirtualAlloc
00007674 | 00007674 | 0124 | GetModuleFileNameA
0000768A | 0000768A | 0126 | GetModuleHandleA
00007612 | 00007612 | 00B6 | FreeResource
00007664 | 00007664 | 00A3 | FindResourceA
CFF Explorer VIII - [Malware_Build_Week_U3.exe]
File Settings ?
Malware_Build_Week_U3.exe
File: Malware_Build_Week_U3.exe
Dos Header
Nt Headers
File Header
Optional Header
Data Directories [x]
Section Headers [x]
Import Directory
Resource Directory
Address Converter
Dependency Walker
Hex Editor
Identifier
Import Adder
Quick Disassembler
Rebuilder
Resource Editor
UPX Utility
Module Name | Imports | OFTs | TimeDateStamp | ForwarderChain | Name RVA | FTs (IAT)
000076D0 | N/A | 00007500 | 00007504 | 00007508 | 0000750C | 00007510
szAnsi | (nFunctions) | Dword | Dword | Dword | Dword | Dword
KERNEL32.dll | 51 | 00007534 | 00000000 | 00000000 | 0000769E | 0000700C
ADVAPI32.dll | 2 | 00007528 | 00000000 | 00000000 | 000076D0 | 00007000
OFTs | FTs (IAT) | Hint | Name
Dword | Dword | Word | szAnsi
000076AC | 000076AC | 0186 | RegSetValueExA
000076BE | 000076BE | 015F | RegCreateKeyExA

TRACCIA 2

1.Scopo della funzione chiamata alla locazione di memoria 00401021

All'indirizzo di memoria 401021, viene chiamata la funzione RegCreateKeyExA per creare una chiave di registro.

2.Come vengono passati i parametri alla funzione

I parametri vengono passati attraverso il comando push, che inserisce i dati, uno ad uno, in cima allo stack.

3.Che oggetto rappresenta il parametro alla locazione 00401017

Il parametro alla locazione 00401017 è la sottochiave "SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon", dove risiedono informazioni riguardanti gli utenti e i programmi in avvio al login dell'utente.

```
.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
```

4.Significato delle istruzioni comprese tra gli indirizzi 00401027 e 00401029

Con test eax, eax viene verificato se il valore del registro è 0. Quindi se eax è uguale a 0, viene effettuato un salto alla locazione di memoria 00401032.
Quando eax è uguale a 0, la creazione della chiave è avvenuta con successo

```
.text:00401021    call    ds:RegCreateKeyExA
.text:00401027    test    eax, eax
.text:00401029    jz      short loc_401032
.text:0040102B    mov     eax, 1
.text:00401030    jmp     short loc_40107B
```

5.Traduzione codice da Assembly in C

```c
if(eax==0){
    //Salta alla locazione 401032
    //Chiama RegSetValue()
}
else{
    eax = 1;
    //Salta alla locazione 40107B
}
```

6.Valore del parametro ValueName

Il valore del "ValueName" è GinaDLL. GinaDLL è l'acronimo di Graphical Identification and Authentication DLL. Si tratta di una libreria di collegamento dinamico (DLL) che fornisce l'interfaccia grafica per l'autenticazione degli utenti in Windows.

```
.text:00401032 ; ---------------------------------------------------------------------------
.text:00401032
.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
```

La funzionalità che sta implementando il Malware in questa sezione, ipotizzando che avvenga effetivamente il salto, è la creazione di una nuova chiave di registro con asseganti valore GinaDLL.