Analisi avanzate: Un approccio pratico

Parte 1:

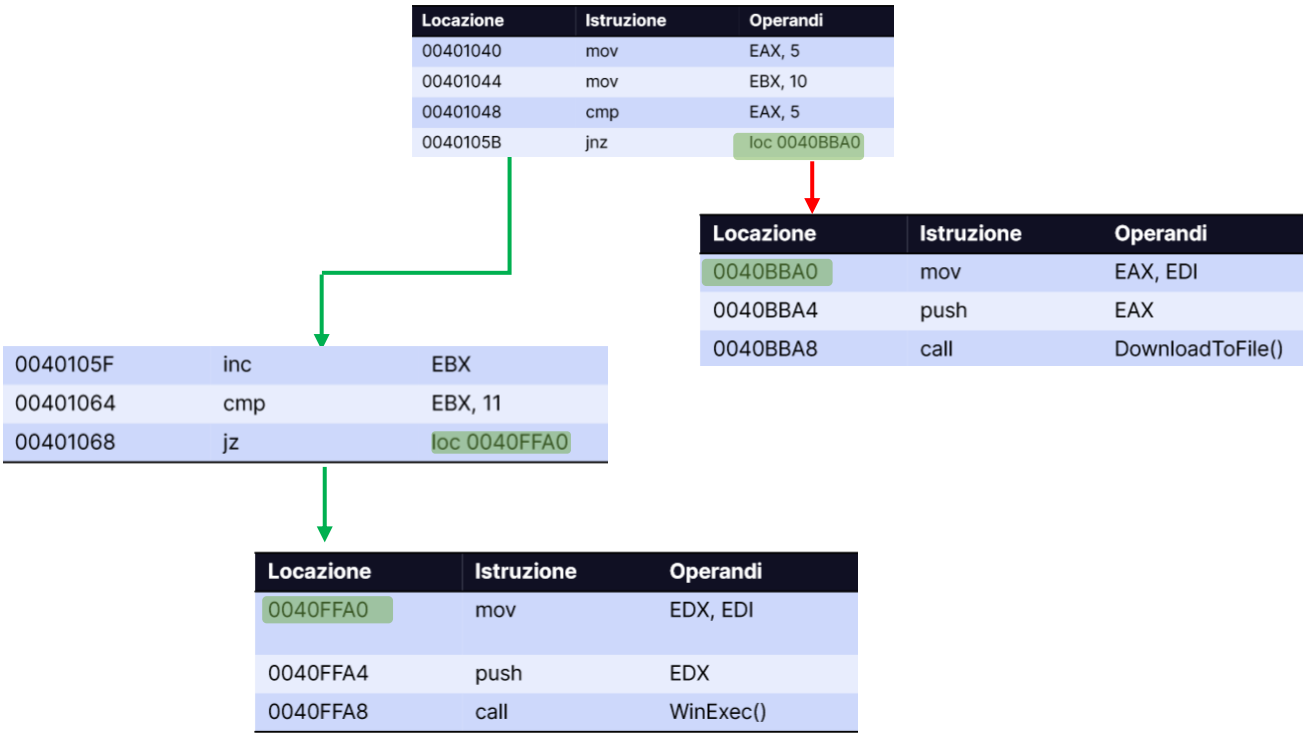
- *Spiegare, motivando, quale salto condizionale effettua il Malware*

Un salto condizionale è un'istruzione con cui il flusso del programma salta a un indirizzo del codice indicato, in base ad una determinata condizione.

Il malware in questo caso compie un salto **JNZ** (**Jump if Not Zero**) se la comparazione *CMP EAX, 5* riporta come risultato un **ZF=0** (Zero Flag = 0 indica che i due operatori erano diversi), altrimenti continua l'esecuzione, può effettuare un altro salto **JZ** (**Jump if Zero**) se la comparazione *CMP EBX, 11* riporta come risultato **ZF=1** (Zero Flag = 1 indica che i due operatori erano uguali)

- *Disegnare un diagramma di flusso (prendere come esempio la visualizzazione grafica di IDA) identificando i salti condizionali (sia quelli effettuati che quelli non effettuati). Indicare con una linea **verde** i salti effettuati, mentre con una linea **rossa** i salti non effettuati.*

Il primo salto non avviene in quanto EAX gli viene assegnato il valore 5 con mov quindi la comparazione cmp EAX, 5 riporta un risultato di 5=5 quindi la ZF=1, per far si che avvenga il salto ZF deve essere 0.



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

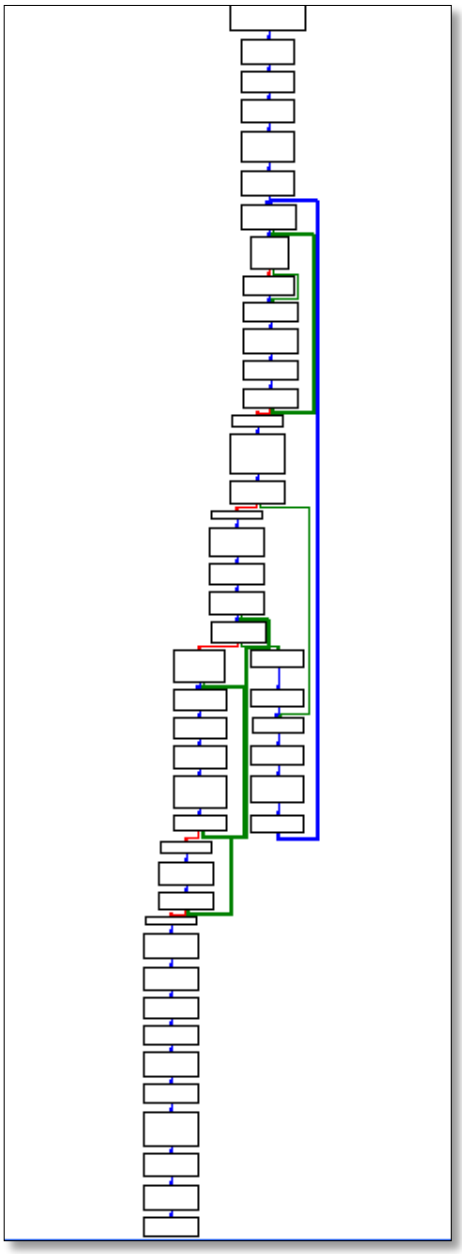
MOV assegna i valori 5 a EAX e 10 a EBX,
CMP compara il valore di EAX con 5
JNZ salta all'indirizzo di destinazione LOC 0040BBA0 se i valori precedentemente confrontati non sono uguali, quindi ZF=0.
INC incrementa il valore di EBX di 1.
CMP confronta il valore di EBX con 11.
JZ salta all'indirizzo di destinazione LOC 0040FFAO se i valori precedentemente confrontati sono uguali, quindi ZF=1.
JNZ su LOC 0040BBA0:

MOV assegna il valore di EDI a EAX (EDI è www.malwaredownload.com).
PUSH mette il valore di EAX nello stack come argomento per la successiva chiamata alla funzione
CALL chiamata della pseudo funzione **DOWNLOADTOFILE**()
JZ su LOC 0040FFAO:
MOV assegna il valore di EDI a EDX (EDI è C:\Program and Settings\LocalUser\Desktop\Ransomware.exe).
PUSH mette il valore di EDX nello stack come argomento per la successiva chiamata alla funzione
CALL chiamata della funzione **WINEXEC**().
Le funzioni implementate sono DONWLOADTOFILE e WINEXEC.

DownloadToFile(): data l'assegnazione con MOV di un URL come valore a EAX (www.malwaredownload.com) possiamo ipotizzare che il malware voglia scaricare un file da quell'URL e salvarlo sulla macchina in cui si trova per poi avviarlo.
WinExec(): esegue il file specificato dal percorso (C:\Program and Settings\LocalUser\Desktop\Ransomware.exe) che apparentemente è un file eseguibile (.exe).

Parte 2:
Il dipendente riceve una mail losca e chiama il SOC. Siamo certi che è un malware, anche se innocuo.

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



Dagli import possiamo vedere le librerie importate che sono:

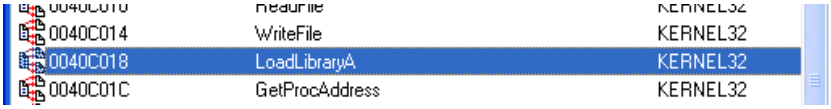
KERNEL32.dll: è una libreria di sistema fondamentale per i sistemi operativi Windows. Contiene un insieme di funzioni che consentono l'interazione con il sistema operativo e la gestione delle risorse di base. Ad esempio la gestione dei file, dei processi e dei thread, della memoria, sincronizzazione e segnalazione degli eventi.

MSVCRT.dll: contiene implementazioni di funzioni standard del linguaggio C, inclusa la manipolazione delle stringhe, l'allocazione di memoria dinamica e le operazioni di input/output.

WS2_32.dll e **WSOCK32.dll**: contengono le implementazioni di funzioni per la creazione e la gestione dei socket, la comunicazione di rete, l'utilizzo di protocolli di rete come TCP/IP e UDP, la risoluzione dei nomi di dominio e altre operazioni di rete.

Nello specifico riconosciamo alcune funzioni già note per essere usate dai malware:

LOADLIBRARY e GETPROCADDRESS: usate per caricare funzioni addizionali durante l'esecuzione.



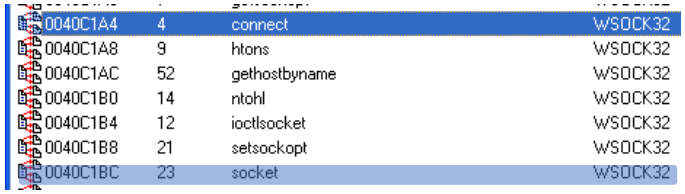
SOCKET: usata per creare un socket

CONNECT: usata lato client per procedere alla connessione verso un socket in ascolto

WSASTARTUP: usata per allocare risorse che verranno poi usate dalle librerie del network

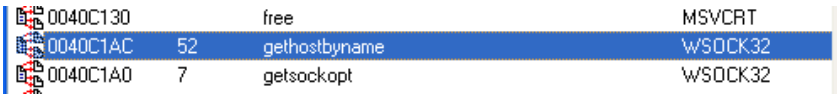
WSARECV: usata per ricevere dati da un socket di comunicazione di rete

WSASEND: usata per inviare dati attraverso un socket di comunicazione di rete

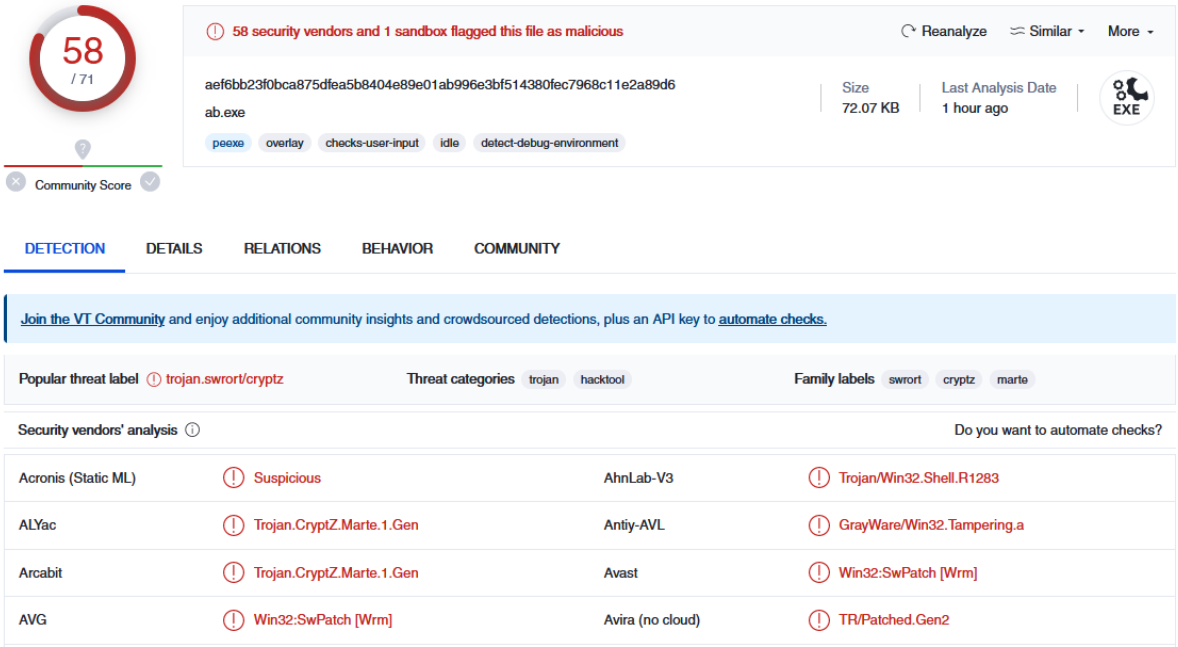


GETCOMMANDLINE: usata per accedere alla riga di comando con cui è stato avviato e ottenere informazioni utili per il suo funzionamento o per nascondere la sua presenza.

GETHOSTBYNAME: usata per ottenere informazioni sul nome host, come l'indirizzo IP associato a un determinato nome di dominio.



Analizzandolo tramite Virus Total riscontriamo un risultato di 58/71 vendor che lo segnalano principalmente come trojan.



2. *Indicare il tipo di malware e il comportamento.*
Questa analisi ci indica il file come Trojan, un tipo di malware che si presenta come un software legittimo o inoffensivo ma in realtà nasconde funzionalità malevole. Le funzionalità di un trojan possono essere molte, tra cui la creazione di backdoor, spyware e keylogger. Nel nostro caso specifico possiamo dire che si tratta di una backdoor in quanto abbiamo delle funzioni usate principalmente da questo tipo di malware.