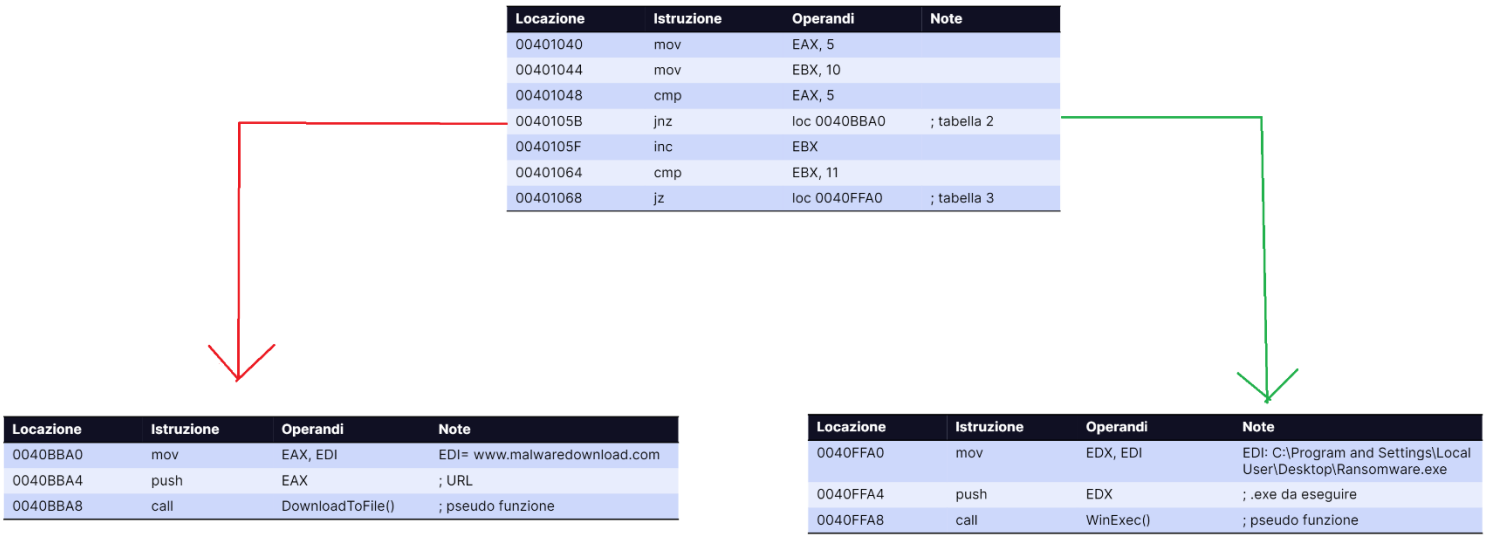
Progetto Week 3 Unit 3

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

Datoci il codice assembly in figura possiamo notare che il malware basa il suo salto condizione (jz) sulla basa del valore del registro EBX=11; EBX viene specificato a 10 e incrementato (inc) di 1 arrivando perciò al valore 11 e saltando all'allocazione di memoria "**loc 0040FFA0**" presente a tabella 3, fosse stato diverso da 11 il salto sarebbe avvenuto all'allocazione di memoria "**loc 0040BBA0**" presente a tabella 2.

Qui sottostante l'immagine del diagramma di flusso del salto condizionale spiegato sopra:



Specificando che la linea verde corrisponde al salto effettuato e quella rossa al salto NON effettuato possiamo notare che il malware si comporta in questo modo:

1. Il malware tramite internet tenta di scaricare un file dal sito www.malwaredownload.com
2. Una volta scaricato il file o se già presente avvia un processo di esecuzione di esso

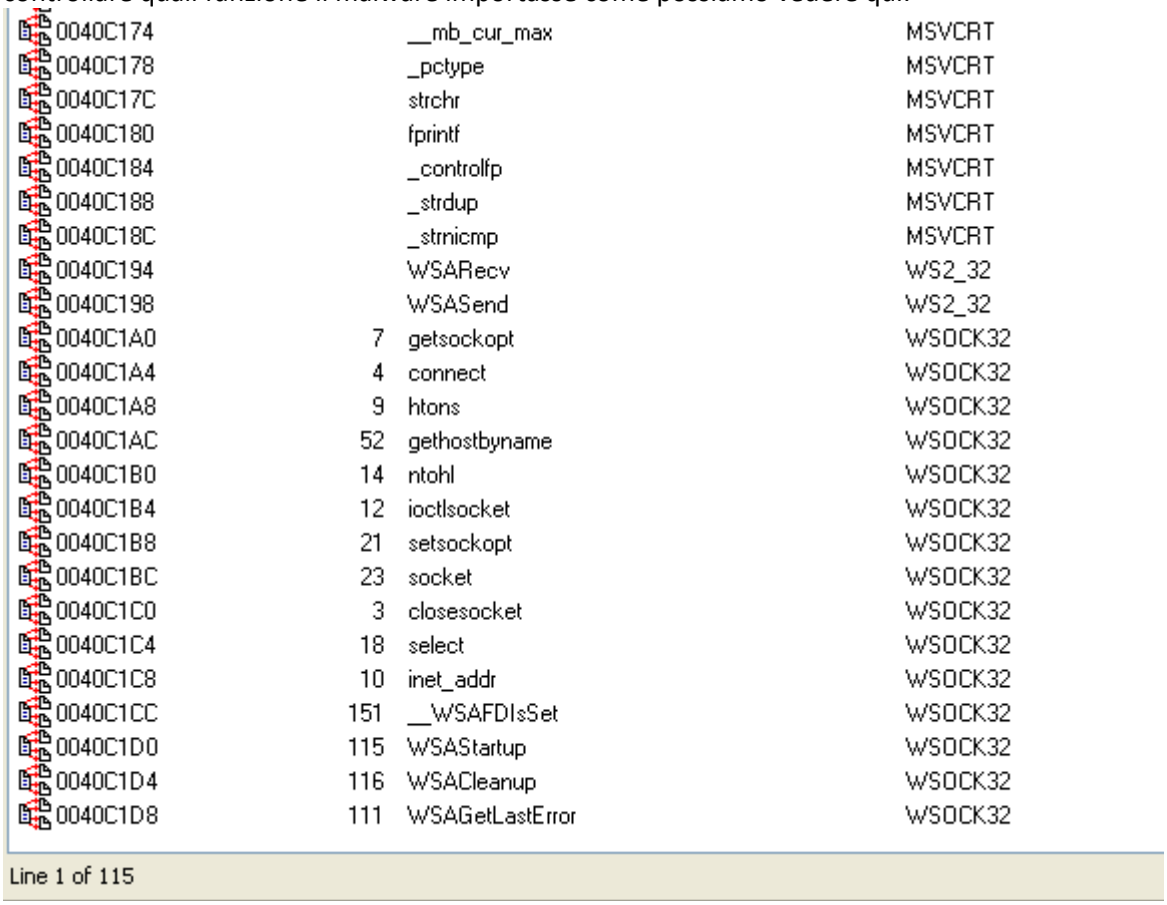
Da queste informazioni possiamo dedurre che il malware sia un **Downloader** in quanto è tipico di questa tipologia di malware scaricare file malevoli via internet per poi eseguirli o salvarli per uso futuro.

Concentrandoci sulle istruzioni “call” presenti sia in tabella 2 che 3 possiamo affermare:

- “**0040BBA8 call DownloadToFile()**” mette il sito www.malwaredownload.com nella funzione DownloadToFile() per essere scaricato.
- “**0040FFA8 call Winexec()**” prende il percorso del file malevolo, scaricato o già presente, e lo esegue con la funzione Winexec.

Parte 2

Aprendo il file di esecuzione datoci con IDA pro la prima cosa che ho controllato è la sezione “import” per controllare quali funzione il malware importasse come possiamo vedere qui:



0040C174		__mb_cur_max	MSVCRT
0040C178		_pctype	MSVCRT
0040C17C		strchr	MSVCRT
0040C180		fprintf	MSVCRT
0040C184		_controlfp	MSVCRT
0040C188		_strdup	MSVCRT
0040C18C		_strnicmp	MSVCRT
0040C194		WSARecv	WS2_32
0040C198		WSASend	WS2_32
0040C1A0	7	getsockopt	WSOCK32
0040C1A4	4	connect	WSOCK32
0040C1A8	9	htons	WSOCK32
0040C1AC	52	gethostbyname	WSOCK32
0040C1B0	14	ntohl	WSOCK32
0040C1B4	12	ioctlsocket	WSOCK32
0040C1B8	21	setsockopt	WSOCK32
0040C1BC	23	socket	WSOCK32
0040C1C0	3	closesocket	WSOCK32
0040C1C4	18	select	WSOCK32
0040C1C8	10	inet_addr	WSOCK32
0040C1CC	151	__WSAFDIsSet	WSOCK32
0040C1D0	115	WSAStartup	WSOCK32
0040C1D4	116	WSACleanup	WSOCK32
0040C1D8	111	WSAGetLastError	WSOCK32

Line 1 of 115

Guardando specialmente questa porzione di funzioni importate possiamo vedere funzioni provenienti dalle librerie “WS2_32” e “WSOCK32” che sono usate nella gestione di connessioni tramite socket, leggendo il nome in particolare di “connect”, “gethostbyname”, “socket” possiamo dedurre che il malware sia una **backdoor** di tipo **client** principalmente data la presenza della funzione “connect” e non è possibile possa essere di tipo server perché mancano le funzioni di “bind”, che associa il socket a un indirizzo IP/porta, e “listen” che permette al socket di mettersi in ascolto in attesa di connessioni esterne.
I malware di tipo backdoor tipicamente usano funzionalità di networking per creare nella macchina infetta un socket che può funzionare sia da server o da client che si connette a un server malevolo, in ogni caso permette all’attaccante di creare una connessione con l’host da remoto che gli da funzionalità da amministratore e/o esecuzione-download.