



DATA SHIELDS

MALWARE ANALISYS

MARCO MALIZIA
DATASHIELDS

INCARICO

CON RIFERIMENTO AL FILE MALWARE_U3_W2_L5 PRESENTE ALL'INTERNO DELLA CARTELLA «ESERCIZIO_PRATICO_U3_W2_L5 » SUL DESKTOP DELLA MACCHINA VIRTUALE DEDICATA PER L'ANALISI DEI MALWARE, RISPONDERE AI SEGUENTI QUESITI:

1. QUALI LIBRERIE VENGONO IMPORTATE DAL FILE ESEGUIBILE?

2. QUALI SONO LE SEZIONI DI CUI SI COMPONE IL FILE ESEGUIBILE DEL MALWARE?

CON RIFERIMENTO ALLA FIGURA 1, RISPONDERE AI SEGUENTI QUESITI:

3. IDENTIFICARE I COSTRUTTI NOTI (CREAZIONE DELLO STACK, EVENTUALI CICLI) 4. IPOTIZZARE IL COMPORTAMENTO DELLA FUNZIONALITÀ

5. BONUS FARE TABELLA CON SIGNIFICATO DELLE SINGOLE RIGHE DI CODICE ESERCIZIO TRACCIA E REQUISITI ALTRI COSTRUTTI) IMPLEMENTATA ASSEMBLY.



INTRO

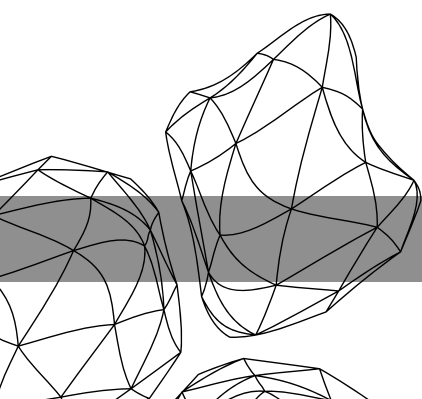
QUESTO REPORT È UN'ANALISI APPROFONDATA DI UN'ESEGUIBILE SOSPETTO CLASSIFICATO COME MALWARE. LE INDAGINI SONO STATE CONDOTTE IN UN AMBIENTE CONTROLLATO E ISOLATO PER PREVENIRE RISCHI ALLA SICUREZZA DURANTE L'ANALISI.

METODO

PER ASSICURARE UN'ANALISI COMPLETA, SONO STATI UTILIZZATI QUATTRO STRUMENTI PRINCIPALI, CIASCUNO MIRATO A ESAMINARE ASPETTI DIVERSI DELL'ESEGUIBILE MALWARE:

1 CFF EXPLORER: UTILIZZATO PER ESPLORARE LE STRUTTURE INTERNE DELL'ESEGUIBILE, COMPRESSE LE LIBRERIE IMPORTATE E LE SEZIONI DI MEMORIA.

2 VIRUS TOTAL: QUESTO SERVIZIO ONLINE AGGREGA I RISULTATI DI SCANSIONE DI NUMEROSI MOTORI ANTIVIRUS E STRUMENTI DI ANALISI DEL MALWARE PER FORNIRE UNA PANORAMICA COMPRENSIVA SULLA SICUREZZA DI UN FILE.





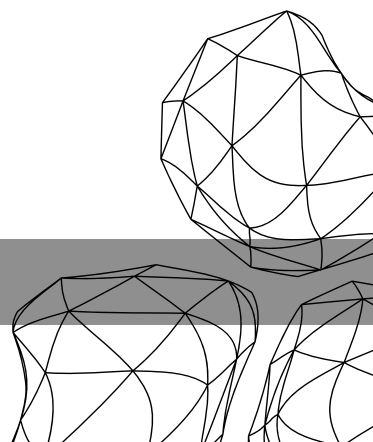
CONFIGURAZIONI PRELIMINARI

CONFIGURAZIONE SCHEDE DI RETE: DURANTE IL TEST, LA MACCHINA NON DEVE AVERE ACCESSO DIRETTO A INTERNET O AD ALTRE MACCHINE, COSÌ DA LIMITARE L'INFEZIONE GENERATA DAL MALWARE.

- CARTELLE CONDIVISE: EVITARE DI CONDIVIDERE CARTELLE TRA LA MACCHINA E LE VIRTUAL MACHINE POICHÈ IL MALWARE POTREBBE PROPAGARSI E CAUSARE DANNI ALLA MACCHINA E ALLA RETE DOMESTICA.

- DISPOSITIVI USB: È CONSIGLIABILE DISABILITARE IL CONTROLLER USB, POICHÉ IL MALWARE POTREBBE USARLO PER DIFFONDERSI SULLA MACCHINA FISICA.

- CREAZIONE DI ISTANTANEE: ANALIZZANDO MALWARE, L'AMBIENTE DI TEST PUÒ ESSERE DANNEGGIATO O COMPROMESSO. È UTILE CREARE ISTANTANEE DELLA MACCHINA VIRTUALE PRIMA DI INIZIARE, COSÌ DA POTERLA RIPRISTINARE FACILMENTE.

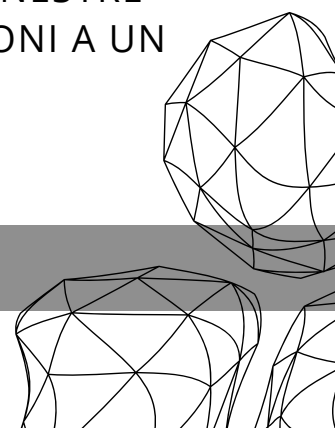




MALWARE

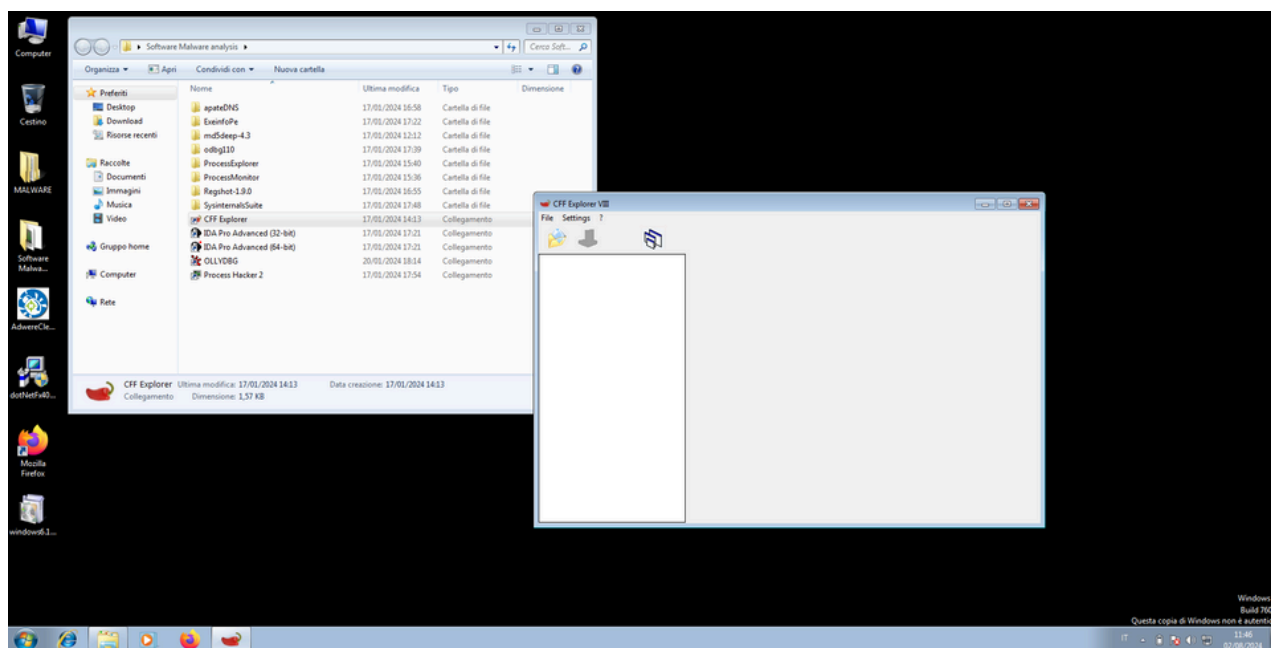
IL TERMINE MALWARE INDICA QUALSIASI SOFTWARE CREATO PER DANNEGGIARE, COMPROMETTERE O ALTERARE UN SISTEMA INFORMATICO, UN DISPOSITIVO O UNA RETE, SENZA IL CONSENSO DELL'UTENTE. PUÒ AVERE DIVERSE FORME E SVOLGERE VARIE ATTIVITÀ DANNOSE.

DOPO UNA RICERCA IN RETE POSSIAMO CITARE I TIPI PIÙ COMUNI DI MALWARE:

- **VIRUS:** SI DIFFONDE TRA COMPUTER SENZA AUTORIZZAZIONE, COPIANDOSI NEL FILE SYSTEM E CERCANDO DI NASCONDERSI DAGLI ANTIVIRUS.
 - **TROJAN:** SI NASCONDE IN FILE APPARENTEMENTE INNOCUI, COME DOCUMENTI OFFICE O PDF, E SI ATTIVA QUANDO VENGONO APERTI, SPESSO FORNENDO AGLI ATTACCANTI ACCESSO REMOTO.
 - **ROOTKIT:** SI NASCONDE DAGLI UTENTI E DAGLI ANTIVIRUS, PERMETTENDO IL CONTROLLO COMPLETO DEL SISTEMA OPERATIVO SENZA ESSERE RILEVATO.
 - **BOOTKIT:** UN TIPO DI ROOTKIT CHE SI ATTIVA PRIMA DELL'AVVIO DEL SISTEMA OPERATIVO, AGGIRANDO LE PROTEZIONI DI SICUREZZA.
 - **ADWARE:** MOSTRA PUBBLICITÀ INDESIDERATE AGLI UTENTI.
 - **SPYWARE:** RACCOGLIE INFORMAZIONI SULLE ATTIVITÀ DEGLI UTENTI, COME SITI VISITATI E PASSWORD, INVIANDOLE A UN SERVER CONTROLLATO DALL'ATTACCANTE.
 - **DIALER:** CHIAMA NUMERI TELEFONICI A PAGAMENTO PER GUADAGNARE SOLDI.
 - **KEYLOGGER:** REGISTRA OGNI TASTO PREMUTO E LE FINESTRE APERTE DALL'UTENTE, INVIANDO QUESTE INFORMAZIONI A UN SERVER CONTROLLATO DALL'ATTACCANTE.
- 

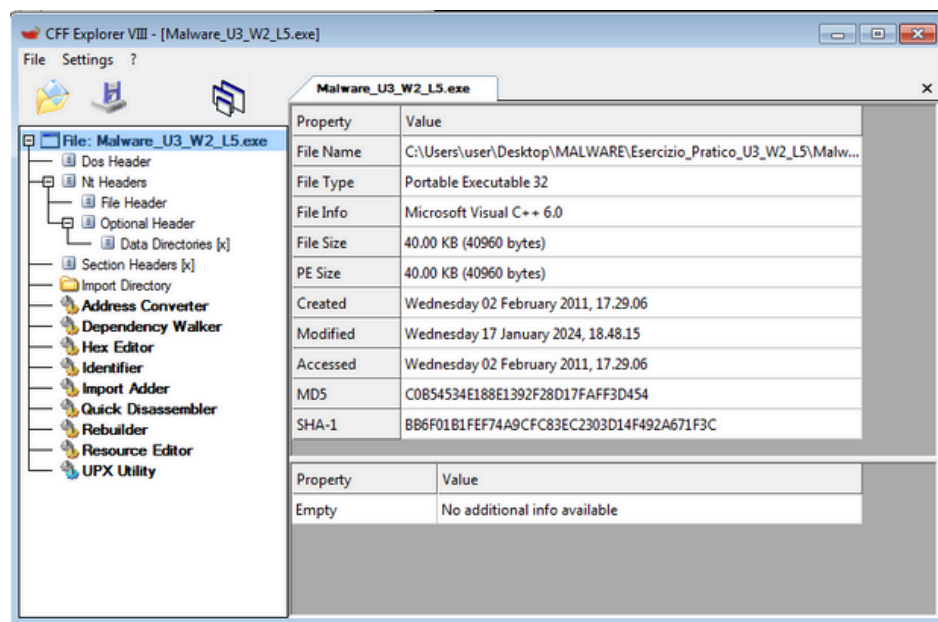
CFF EXPLORER

PER CONTROLLARE LE FUNZIONI IMPORTATE ED ESPORTATE DA UN MALWARE, POSSIAMO USARE IL TOOL **CFF EXPLORER**, INSTALLATO SULLE MACCHINE VIRTUALI PER L'ANALISI DEI MALWARE. CFF EXPLORER PERMETTE DI VISUALIZZARE LE SEZIONI DEL FILE, ESPLORE LE DIRECTORY DI IMPORTAZIONE ED ESPORTAZIONE, ANALIZZARE LE RISORSE E SUPPORTA LO SCRIPTING. È UTILE PER SVILUPPATORI E ANALISTI DI SICUREZZA. BASTA APRIRE IL PROGRAMMA E SCEGLIERE UN FILE ESEGUIBILE DA ANALIZZARE.



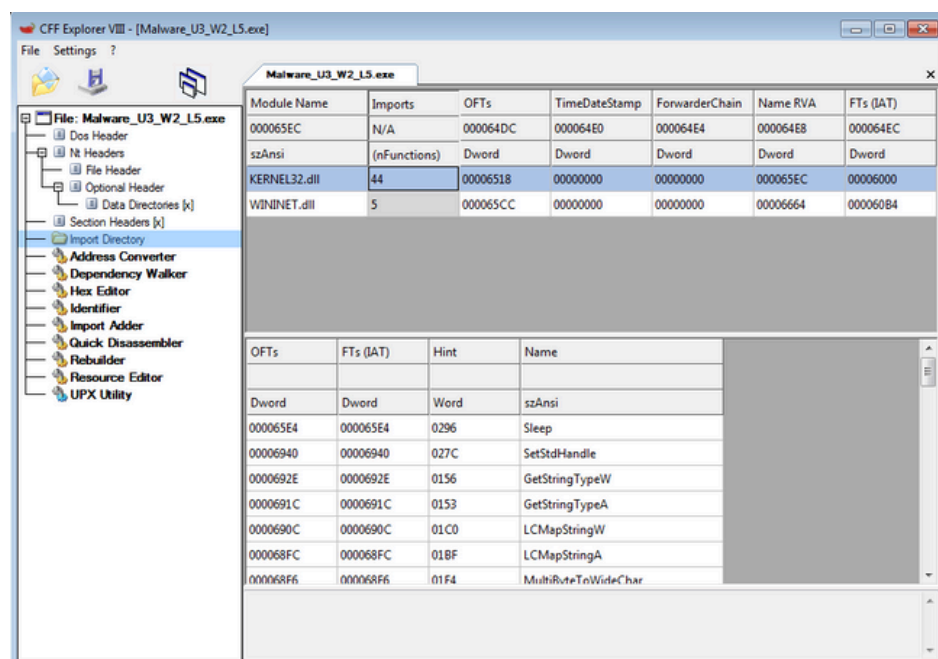
UNA VOLTA APERTO IL PROGRAMMA POSSIAMO ANDARE AD IMPORTARE IL FILE MALEVOLO DA ANALIZZARE

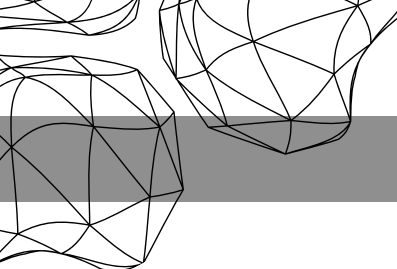
APRIAMO IL SOFTWARE



LIBRERIE

SPOSTANDOSI NELLA DIRECTORY "IMPORT DIRECTORY" POSSIAMO VISIONARE LE LIBRERIE IMPORTATE DAL FILE MALEVOLO, MENTRE PER OGNUNA DELLE LIBRERIE, IL PANNELLO INFERIORE CI MOSTRERÀ LA LISTA DELLE FUNZIONI RICHIESTE ALL'INTERNO DELLA LIBRERIA SELEZIONATA.

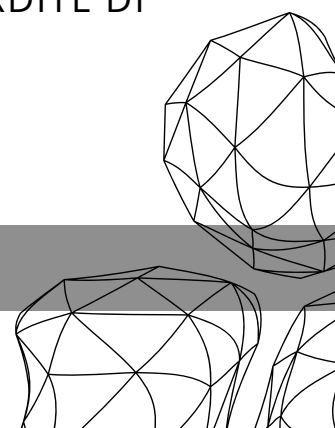




LE LIBRERIE SONO INSIEMI DI FUNZIONI CHE UN PROGRAMMA PUÒ RICHIAMARE QUANDO NECESSARIO. LE LIBRERIE PRESENTI NELLA SITUAZIONE ATTUALE SONOI:

- **KERNEL32.DLL**: LIBRERIA PIUTTOSTO COMUNE CHE CONTIENE LE FUNZIONI PRINCIPALI PER INTERAGIRE CON IL SISTEMA OPERATIVO, AD ESEMPIO MANIPOLAZIONE DEI FILE E LA GESTIONE DELLA MEMORIA.
- **WININET.DLL**: LIBRERIA CHE CONTIENE LE FUNZIONI PER L'IMPLEMENTAZIONE DI ALCUNI PROTOCOLLI DI RETE COME HTTP, FTP E NTP.

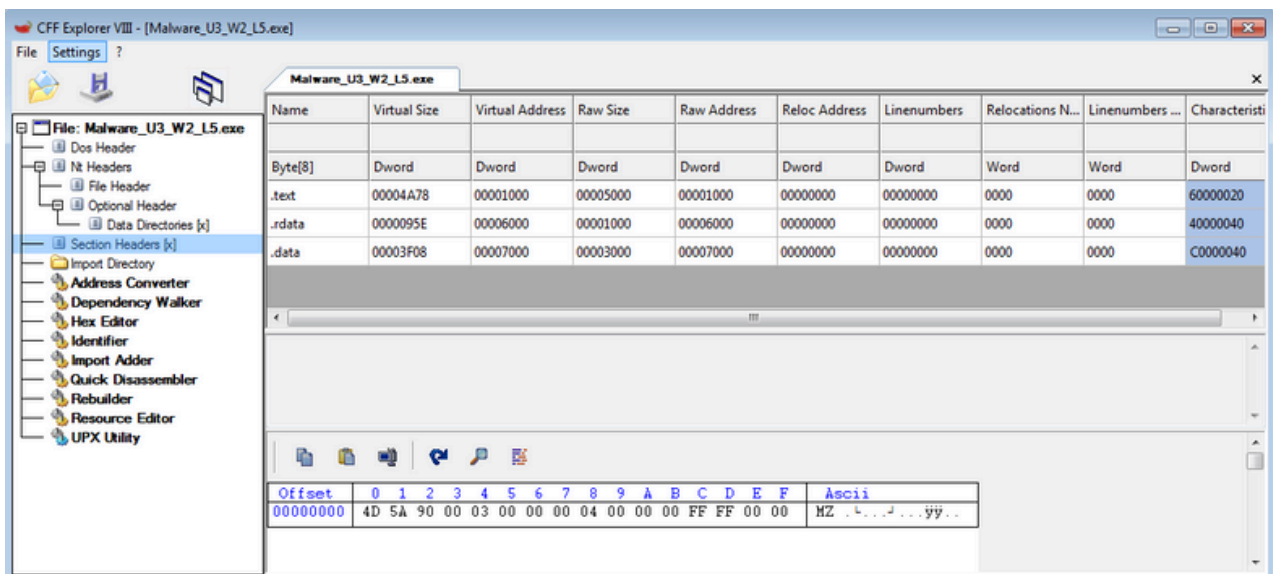
TRA LE 44 FUNZIONI CHE IMPORTA LA LIBRERIA **KERNEL32.DLL**, POSSIAMO NOMINARE LE PIÙ SIGNIFICATIVE:

- **SLEEP**: SOSPENDE TEMPORANEAMENTE L'ESECUZIONE DI UN THREAD PER UN PERIODO SPECIFICATO, PERMETTENDO AD ALTRI THREAD DI UTILIZZARE LA CPU E MIGLIORANDO L'EFFICIENZA DELLE RISORSE.
 - **VIRTUALFREE**: LIBERA LA MEMORIA ALLOCATA CON VIRTUALALLOC, RESTITUENDO MEMORIA NON PIÙ NECESSARIA AL SISTEMA, MIGLIORANDO L'EFFICIENZA E RIDUCENDO IL RISCHIO DI FRAMMENTAZIONE.
 - **VIRTUALALLOC**: RISERVA O ALLOCA MEMORIA VIRTUALE PER UN PROCESSO, CONSENTENDO UNA GESTIONE PIÙ FLESSIBILE DELLE RISORSE SENZA ALLOCARE SUBITO MEMORIA FISICA.
 - **GETPROCADDRESS**: OTTIENE UN PUNTATORE A UNA FUNZIONE IN UNA DLL CARICATA IN MEMORIA, PERMETTENDO DI CHIAMARE FUNZIONI ESPORTATE DURANTE L'ESECUZIONE DEL PROGRAMMA. UTILE PER PLUGIN E MODULI OPZIONALI SENZA DOVER RICOMPILARE L'INTERA APPLICAZIONE.
 - **CLOSEHANDLE**: CHIUDE UN HANDLE, LIBERANDO LE RISORSE ALLOCATE AD ESSO. QUESTA FUNZIONE PREVIENE PERDITE DI MEMORIA E MIGLIORA LA STABILITÀ DEL SISTEMA.
- 

KERNEL32.dll	44	00006518	00000000	00000000
OFTs	FTs (IAT)	Hint	Name	
00006518	00006000	000065E4	000065E6	
Dword	Dword	Word	szAnsi	
000065E4	000065E4	0296	Sleep	
00006810	00006810	02BF	VirtualFree	
0000681E	0000681E	019F	HeapFree	
0000682A	0000682A	022F	RtlUnwind	
00006836	00006836	02DF	WriteFile	
00006842	00006842	0199	HeapAlloc	
0000684E	0000684E	00BF	GetCPInfo	
0000685A	0000685A	00B9	GetACP	
00006864	00006864	0131	GetOEMCP	
00006870	00006870	02BB	VirtualAlloc	
00006880	00006880	01A2	HeapReAlloc	
0000688E	0000688E	013E	GetProcAddress	
000068A0	000068A0	01C2	LoadLibraryA	
000068B0	000068B0	011A	GetLastError	
000068C0	000068C0	00AA	FlushFileBuffers	
000068D4	000068D4	026A	SetFilePointer	
00006950	00006950	001B	CloseHandle	

SEZIONI

NELL'ANALISI STATICA DI UN MALWARE, UNA "SEZIONE" È UNA PARTE SPECIFICA DEL FILE ESEGUIBILE O DEL CODICE SORGENTE CHE VIENE ESAMINATA. LE SEZIONI POSSONO CONTENERE CODICE ESEGUIBILE, DATI, RISORSE E TABELLE DI IMPORTAZIONE ED ESPORTAZIONE.

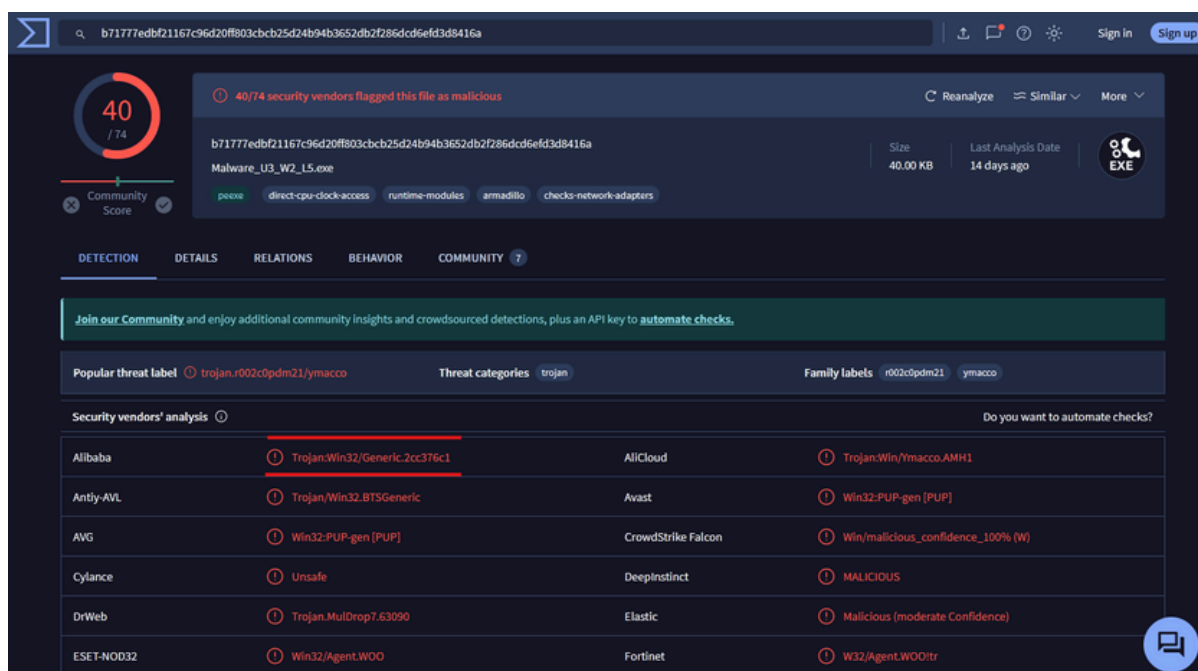
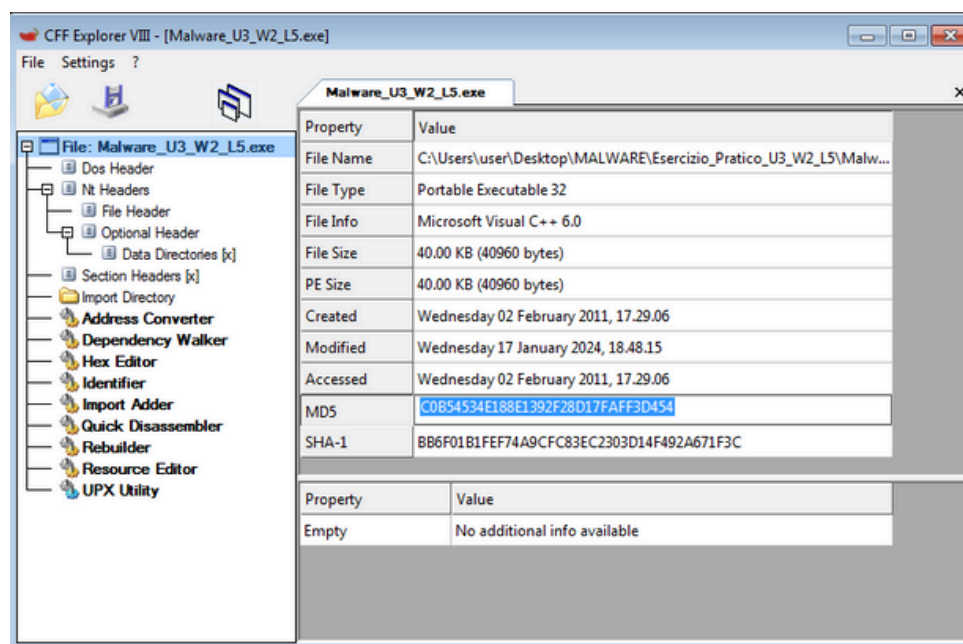


LE SEZIONI PRESENTI SONO:

- **.TEXT:** CONTIENE LE ISTRUZIONI ESEGUIBILI DALLA CPU QUANDO IL SOFTWARE VIENE AVVIATO. È L'UNICA SEZIONE ESEGUITA DALLA CPU, MENTRE LE ALTRE CONTENGONO DATI DI SUPPORTO.
- **.RDATA:** INCLUDE INFORMAZIONI SULLE LIBRERIE E FUNZIONI IMPORTATE ED ESPORTATE DALL'ESEGUIBILE, DATI CHE POSSIAMO OTTENERE CON CFF EXPLORER.
- **.DATA:** CONTIENE I DATI E LE VARIABILI GLOBALI DEL PROGRAMMA, ACCESSIBILI DA QUALSIASI PARTE DEL CODICE. LE VARIABILI GLOBALI SONO DICHIARATE FUORI DALLE FUNZIONI, RENDENDOLE ACCESSIBILI OVUNQUE NEL PROGRAMMA.

ANALISI VIRUSTOTAL

POSSIAMO FARE UN'ANALISI SFRUTTANDO IL SERVIZIO DI **VIRUSTOTAL**, IN CUI, INSERENDO IL CODICE HASH APPARTENENTE AL FILE MALEVOLO OGGETTO, AVREMO COME RISULTATO LA TIPOLOGIA DI **MALWARE** ED IL PUNTEGGIO ASSOCIATO ALLA TIPOLOGIA DI VIRUS DA PARTE DEI VENDOR.





ASSEMBLY

L'ANALISI STATICA AVANZATA RICHIEDE LA CONOSCENZA DEL LINGUAGGIO ASSEMBLY, SPECIFICO PER OGNI ARCHITETTURA DI PC. L'ANALISTA DI SICUREZZA UTILIZZA STRUMENTI CHIAMATI "DISASSEMBLER" PER TRADURRE LE ISTRUZIONI BINARIE DELLA CPU IN ASSEMBLY, RENDENDOLE LEGGIBILI.

L'ASSEMBLY DIPENDE DALL'ARCHITETTURA DEL CALCOLATORE (ES. X86, X64, ARM, MIPS, POWERPC) E SERVE PER "LEGGERE" LE ISTRUZIONI DELLA CPU. PER IL NOSTRO PROGETTO, CI CONCENTREREMO SULL'ASSEMBLY PER I PROCESSORI A 32 BIT (X86).

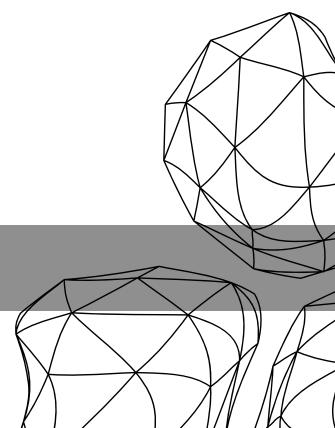
L'ASSEMBLY È COMPOSTO DA ISTRUZIONI CON DUE PARTI:

- **CODICE MNEMONICO:** PAROLA CHE IDENTIFICA L'ISTRUZIONE.
- **OPERANDI:** VARIABILI O MEMORIA OGGETTO DELL'ISTRUZIONE.

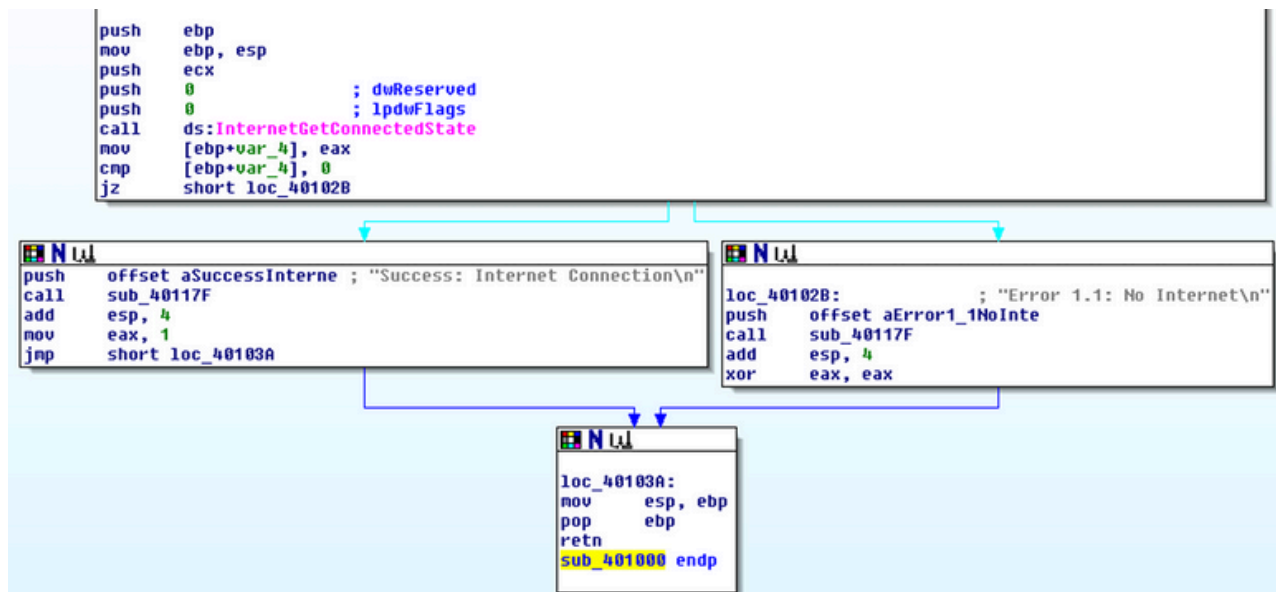
ESISTONO TRE TIPI DI OPERANDI:

- UN VALORE (SOLITAMENTE IN FORMATO ESADECIMALE, ES. 0XYY).
- UN REGISTRO DELLA CPU.
- UN INDIRIZZO DI MEMORIA CONTENENTE UN VALORE.

I REGISTRI SONO MEMORIE A RAPIDO ACCESSO USATE DALLA CPU PER SALVARE TEMPORANEAMENTE VARIABILI.



COSTRUTTI NOTI



```
push    ebp
mov     ebp, esp
push    ecx
```

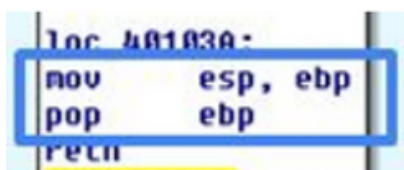
LE ISTRUZIONI "**PUSH EBP**" E "**MOV EBP, ESP**", NEL LINGUAGGIO ASSEMBLY X86 SONO USATE ALL'INIZIO DI UNA FUNZIONE PER CREARE UNO STACK FRAME.

"**PUSH EBP**" SALVA IL VALORE CORRENTE DEL REGISTRO BASE NELLO STACK, MENTRE "**MOV EBP, ESP**" IMPOSTA IL REGISTRO BASE AL VALORE DEL PUNTATORE DELLO STACK (ESP). QUESTO CONSENTE DI ACCEDERE FACILMENTE ALLE VARIABILI LOCALI E AI PARAMETRI DELLA FUNZIONE UTILIZZANDO IL REGISTRO BASE COME RIFERIMENTO.

QUESTO CODICE ASSEMBLY X86 CONFRONTA (CMP) IL VALORE ALL'INDIRIZZO [EBP+VAR_4] CON 0.

```
mov     [ebp+var_4], eax
cmp     [ebp+var_4], 0
jz      short loc_40102B
```

SE IL VALORE È ZERO, L'ISTRUZIONE DI SALTO CONDIZIONATO (JZ) ESEGUE UN SALTO ALL'ETICHETTA LOC_40102B; ALTRIMENTI, L'ESECUZIONE CONTINUA NORMALMENTE.

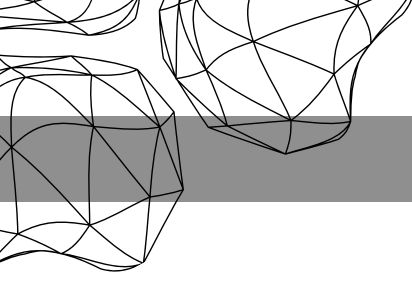


LE ISTRUZIONI "**MOV ESP, EBP**" E "**POP EBP**" NEL LINGUAGGIO ASSEMBLY X86 RIPRISTINANO LO STACK ALLA FINE DI UNA FUNZIONE. "**MOV ESP, EBP**" RIPOSTA IL PUNTATORE DELLO STACK (ESP) AL VALORE SALVATO NEL REGISTRO BASE (EBP). "**POP EBP**" ESTRAE IL VALORE DALLO STACK E LO CARICA IN EBP, RIPRISTINANDO IL REGISTRO BASE ORIGINALE. QUESTE ISTRUZIONI PULISCONO LO STACK FRAME CREATO DALLA FUNZIONE, RIPORTANDOLO ALLO STATO PRECEDENTE.



TABALLA CODICI

PUSH EBP	SALVA IL VALORE DI EBP SULLO STACK
MOV EBP, ESP	COPIA IL VALORE DI ESP IN EBP
PUSH ECX	SALVA IL VALORE DI ECX SULLO STACK
PUSH 0	SPINGE 0 SULLO STACK
CALL DS	CHIAMA LA FUNZIONE INTERNETGETCONNECTEDSTATE
MOV [EBP+VAR_4], EAX	MEMORIZZA IL VALORE DI EAX NELLA VARIABILE VAR_4
CMP [EBP+VAR_4], 0	CONFRONTA IL VALORE DI VAR_4 CON 0
JZ SHORT LOC_40102B	SALTA A LOC_40102B SE IL CONFRONTO È ZERO
PUSH OFFSET ASUCCESSINTERNE ; "SUCCESS: INTERNET CONNECTION"	SPINGE L'INDIRIZZO DEL MESSAGGIO DI SUCCESSO SULLO STACK
CALL SUB_40117F	CHIAMA LA FUNZIONE SUB_40117F
MOV [EBP+VAR_4], EAX	MEMORIZZA IL VALORE DI EAX NELLA VARIABILE VAR_4
ADD ESP, 4	AUMENTA ESP DI 4
MOV EAX, 1	SPOSTA 1 IN EAX
JMP SHORT LOC_40103A	SALTA A LOC_40103A
PUSH OFFSET AERROR1_1NOINTE ; "ERROR 1.1: NO INTERNET"	SPINGE L'INDIRIZZO DEL MESSAGGIO DI ERRORE SULLO STACK
CALL SUB_40117F	CHIAMA LA FUNZIONE SUB_40117F



ADD ESP, 4	AUMENTA ESP DI 4
XOR EAX, EAX	PONE EAX A 0 TRAMITE XOR
MOV ESP, EBP	RIPRISTINA ESP DAL VALORE DI EBP
POP EBP	RIPRISTINA IL VALORE DI EBP
RETN	RITORNA DALLA FUNZIONE
ADD ESP, 4	AUMENTA ESP DI 4