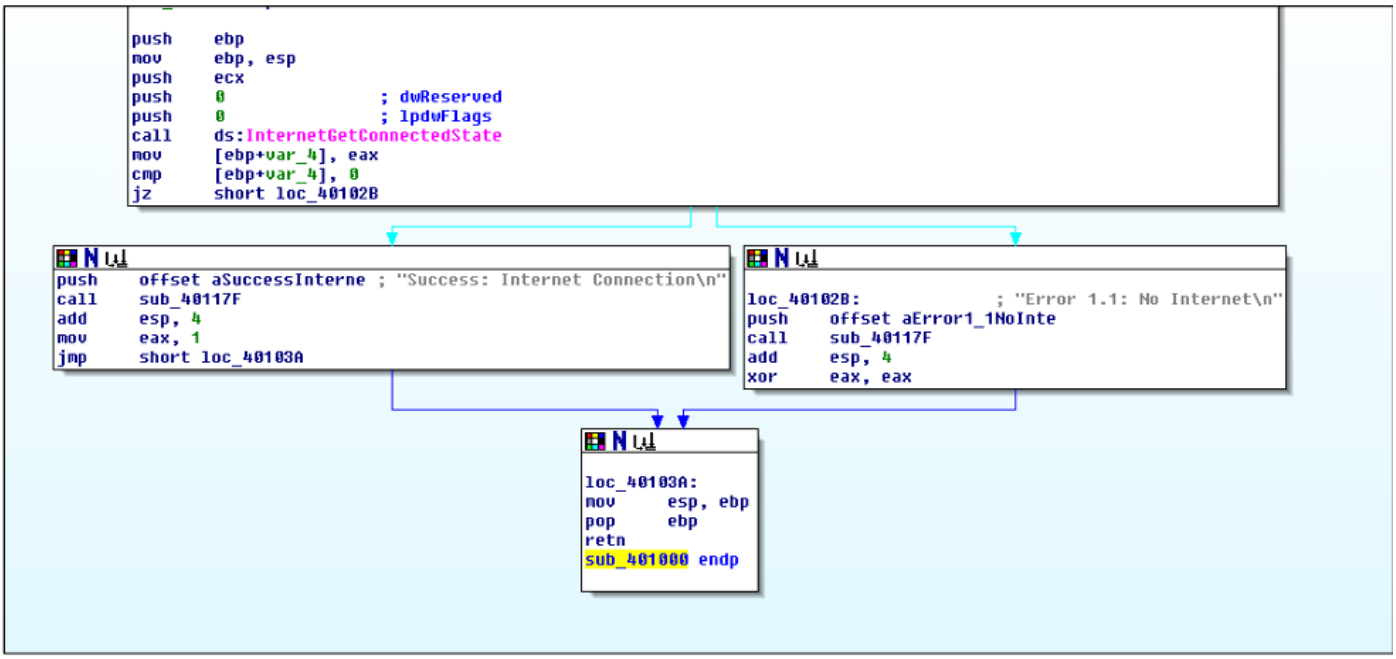
Pratica S10/L5

Traccia:
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, risponde ai seguenti quesiti:
3. Identificare i costrutti noti (creazione dello stack, eventuali cicli, altri costrutti)
4. Ipotizzare il comportamento della funzionalità implementata
5. BONUS fare tabella con significato delle singole righe di codice assembly



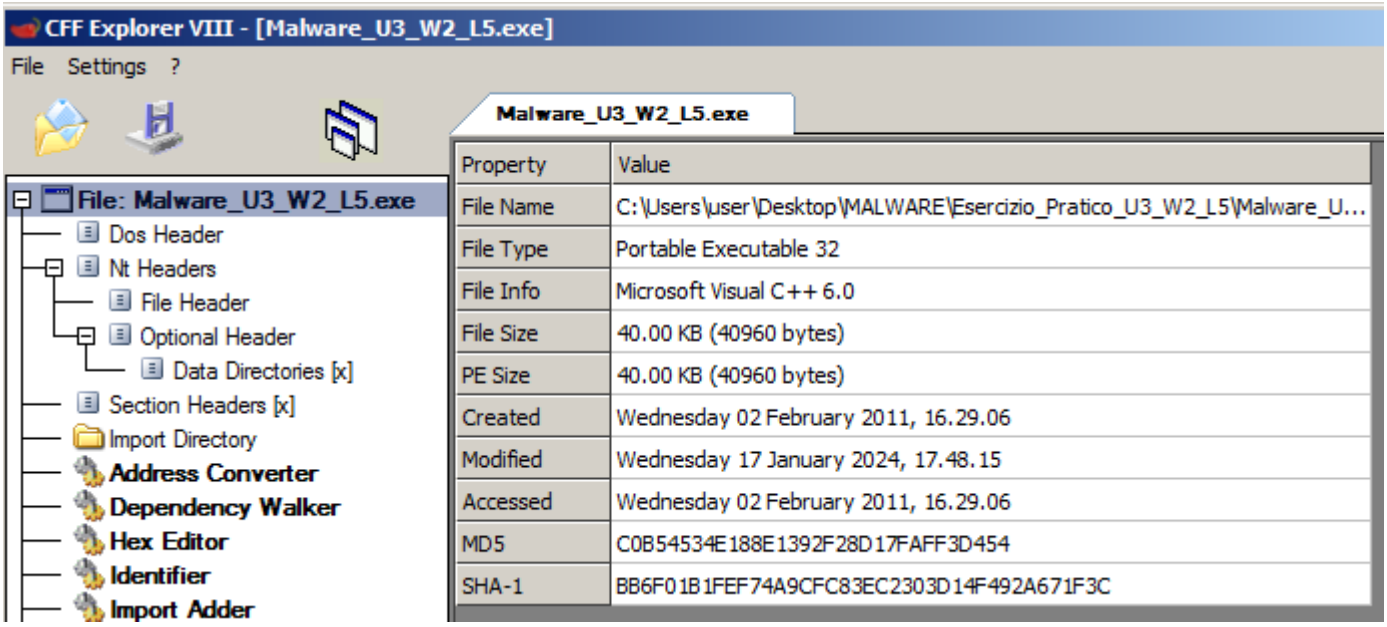
Per la prima parte dell'esercizio viene eseguita un'**Analisi Statica Basica** sul comportamento del Malware.
L'Analisi Statica Basica consiste nell'esaminare un **PE (Portable Executable)**, senza essere eseguito tramite dei tools specifici.
È sicuramente l'analisi più intuitiva, ma inefficace contro malware sofisticati.

I **Malware** (Malicius Software) sono programmi scritti per arrecare danno a sistemi informativi, spesso a scopo di lucro.

Il programma utilizzato per l'analisi statica basica è **CFF Explorer**, un software, Open Source, che permette di controllare le funzioni importate /esportate di un file potenzialmente malevolo.

Per scovare il tipo di virus è stato utilizzato **Virus Total**, un sito web che permette l'analisi gratuita di files e/o URLs e/o Hash per scovare malware all'interno di un programma

Avvio analisi del file «**Malware_U3_W2_L5.exe**» con CFF Explorer



In questa prima immagine ci viene indicato:

- **File Type**; il file eseguibile è a 32 bit;
- **File Info**: Linguaggio di programmazione utilizzato (C++);
- **File Size** a **PE Size**: spazio occupato dal file;
- **MD5** e **SHA-1**: codice HASH (Firma Digitale) identificativo del file.

QUESITI

1. Quali librerie vengono importate dal file eseguibile?

Malware_U3_W2_L5.exe

Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	44	00006518	00000000	00000000	000065EC	00006000
WININET.dll	5	000065CC	00000000	00000000	00006664	000060B4

Le librerie importate dal file sono:

- **KERNEL32.dll**: contiene le informazioni principali per interagire con il sistema operativo;
- **WINIET.dll**: contiene le funzioni per l'implementazione di alcuni protocolli di rete come http, FTP, NTP.

2. Quali sono le sezioni di cui si compone il file eseguibile del malware?

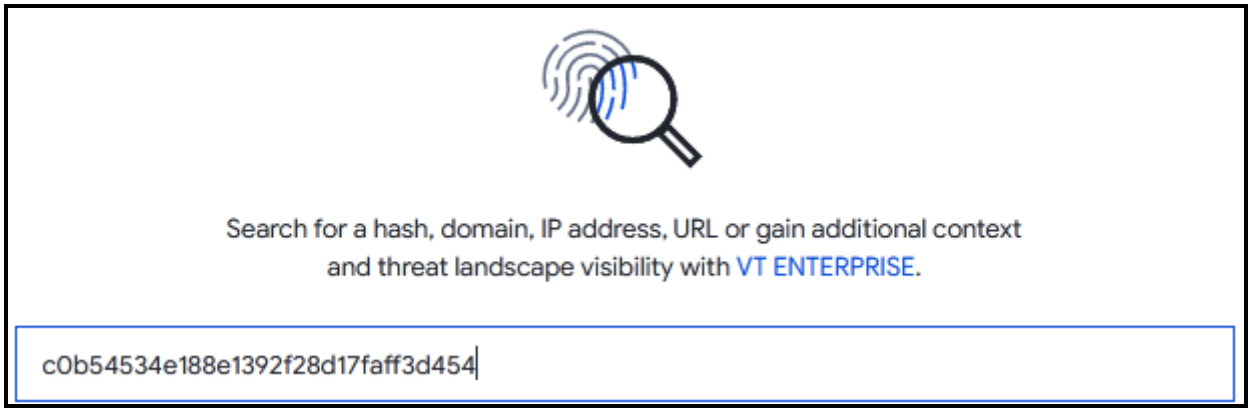
Malware_U3_W2_L5.exe

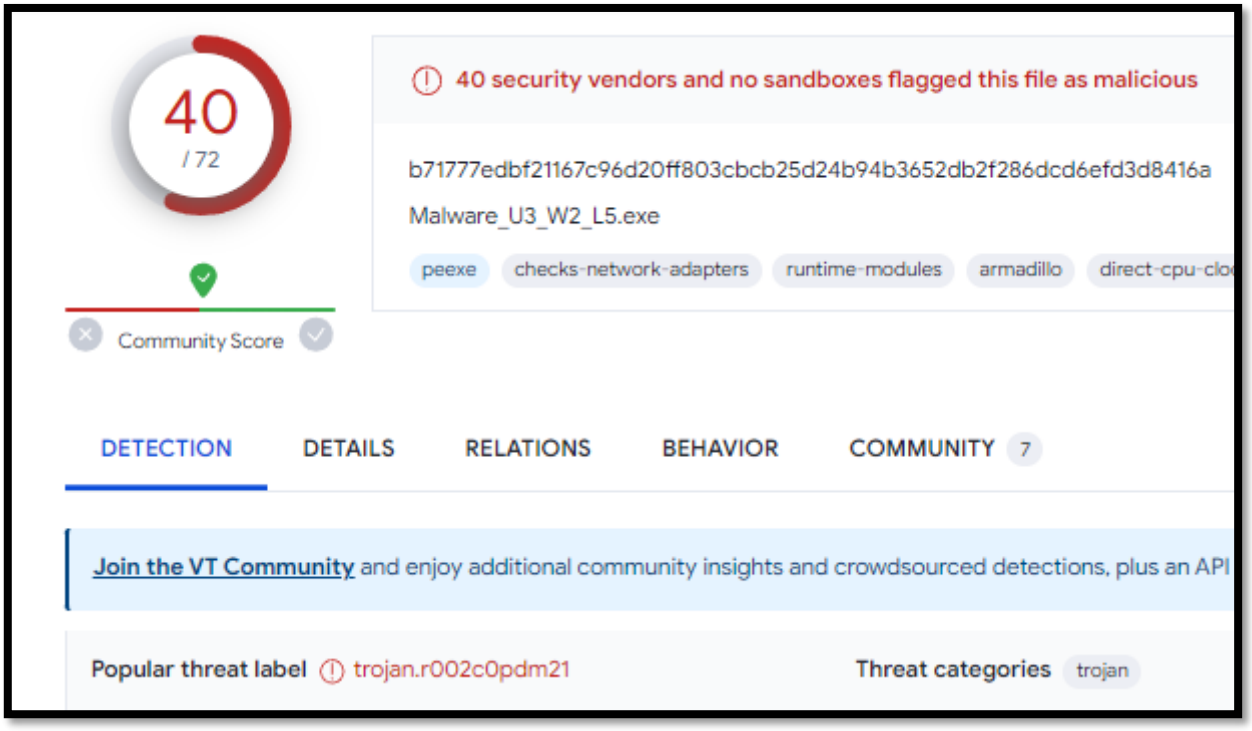
Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations ...	Linenumber...	Characteristics
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
.text	00004A78	00001000	00005000	00001000	00000000	00000000	0000	0000	60000020
.rdata	0000095E	00006000	00001000	00006000	00000000	00000000	0000	0000	40000040
.data	00003F08	00007000	00003000	00007000	00000000	00000000	0000	0000	C0000040

Le sezioni di cui si compone il file sono:

- **.text**: contiene le informazioni (righe di codice) chela CPU eseguirà una volta che il software sarà avviato;
- **.rdata**: include le informazioni delle librerie e delle funzioni importate/esportate;
- **.data**: contiene i dati/variabili globali che devono essere disponibili da qualsiasi parte del programma.

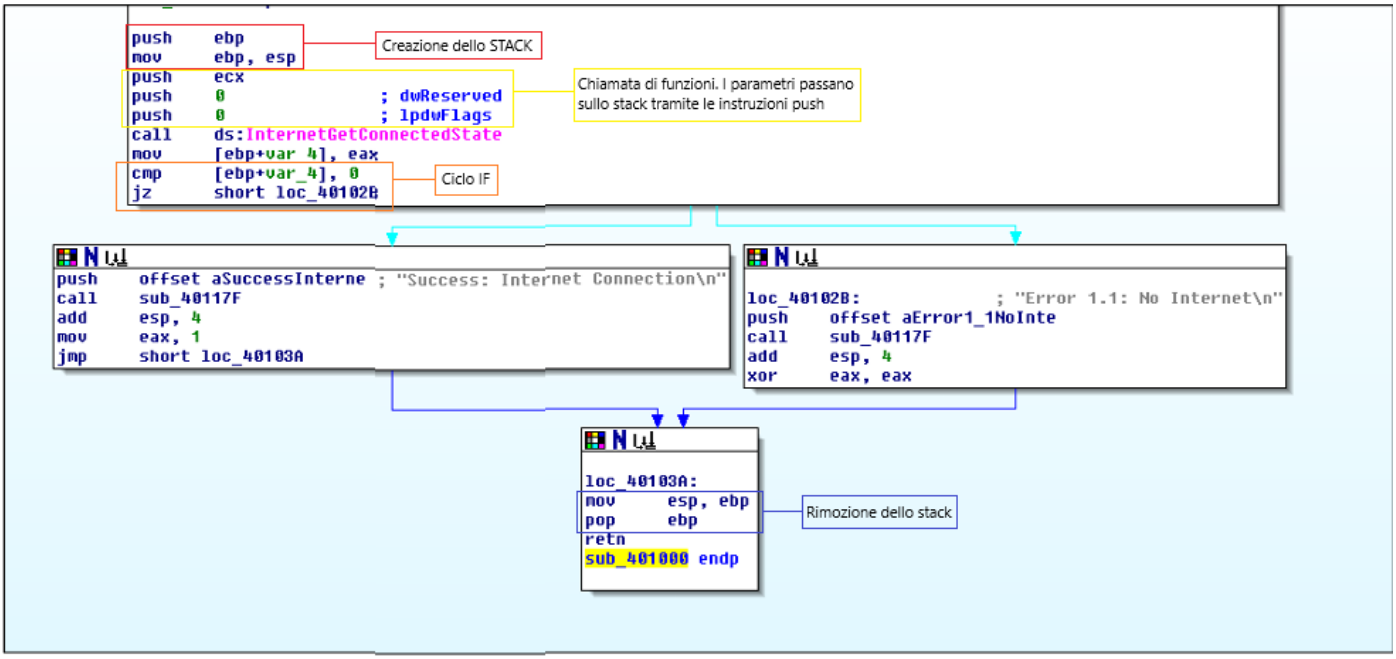
Il codice Hash è stato controllato su di un database online (VirusTotal) per verificare la tipologia ed il comportamento del malware.





Il file risulta essere un **Trojan**, ovvero un file malevolo che può infettare il PC, o qualsiasi dispositivo elettronico, per prenderne il possesso, usando l'inganno e l'ingegneria sociale per convincere utenti ignari a eseguire programmi apparentemente benevoli.
In questo caso permette l'accesso remoto all'attaccante di poter mantenere il controllo remoto del computer, senza che il proprietario ne sia a conoscenza.

3. Identificare i costrutti noti (creazione dello stack, eventuali cicli, altri costrutti)



4. Ipotizzare il comportamento della funzionalità implementata

Verifica la connessione Internet e stampa un messaggio appropriato, Una volta stabilita la connessione il programma tenta di scaricare e leggere un file.

5. BONUS fare tabella con significato delle singole righe di codice assembly

push ebp
Salva il valore ebp (Puntatore alla base dello stack) nello stack.

mov ebp, esp
Copia il valore di esp (Puntatore in cima dello stack) in ebp.

push ecx
Salva il valore di ecx nello stack e lascia spazio per una variabile locale.

push 0 ;dwReserved
Passa 0 come secondo parametro alla funzione InternetGetConnectedState. È un parametro riservato e deve essere sempre 0.

push 0 ;lpdwFlags
Passa 0 come primo parametro alla funzione InternetGetConnectedState. È un puntatore ad una variabile che riceve il tipo di connessione ad internet.

call ds:InternetGetConnectedState
Chiama la funzione InternetGetConnectedState chiamata per determinare se è possibile stabilire una connessione a una destinazione specifica. È di tipo Booleano, ovvero accetta solo 2 valore, vero e falso.

mov [ebp+var_4], eax
Salva il valore restituito dalla funzione InternetGetConnectedState in una variabile lovale.

cmp ebp+var_4], 0
Confronta il valore della variabile locale con 0.

jz short loc_40102B
Salta all'indirizzo 40102B se il valore della variabile locale è 0, cioè se non c'è connessione.

push offset aSuccessInterne ; "Success: Internet Connection\n"
Stampa la stringa "Success: Internet Connection\n" come parametro alla funzione sub_40117F, possibilmente per poter comunicare con un Server remoto.

call sub_40117F
Chiama la funzione sub_40117F definita dall'utente.

add esp, 4
Ripristina il puntatore dello stack dopo aver passato il parametro, sommando 4 byte al valore esp.

mov eax, 1
Assegna 1 al registro eax come valore di ritorno ella funzione corrente.

jmp short loc_40103A
salta incondizionatamente all'indirizzo 40103° per terminare l'esecuzione del programma.

loc_40102B
percorso del salto

push offset aError1_1NoInte ; "Error 1.1: No Internet\n"
Stampa la stringa "Error 1.1: No Internet\n" indicando che non vi è connessione internet, richiamando la funzione all'indirizzo sub_40117F per fare un altro tentativo.

call sub_40117F
Chiama la funzione sub_40117F definita dall'utente.

add esp, 4
Ripristina il puntatore dello stack dopo aver passato il parametro, sommando 4 byte al valore esp.

xor eax, eax
inizializza a 0 il registro eax. Infatti l'operatore logico tra due bit identici restituisce sempre 0. Darà 1 soltanto quando i bit su cui opera sono diversi.

loc_40103A
Percorso del salto

mov esp, ebp
copia il valore ebp in esp

pop ebp
chiude e rimuove il valore di ebp

retn
Organizza il ritorno al programma chiamante al termine della procedura, cioè un sottoprogramma chiamato con call

sub_401000 endp
Chiude il programma

EXTRA

Pesudocodice C++

```
bool checkInternetConnection() {
    DWORD dwFlags;
    BOOL isConnected = InternetGetConnectedState(&dwFlags, 0);
    return isConnected != FALSE;
}

int main() {
    if (checkInternetConnection()) {
        std::cout << "Success: Internet Connection\n";
        return 1;
    } else {
        std::cout << "Error 1.1: No Internet\n";
        return 0;
    }
}
```