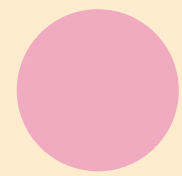


Cyber Security & Ethical Hacking

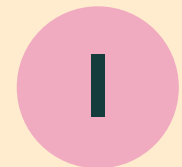
PROGETTO - U3S10L5

Vittorio Braccia - Netrebels

Traccia



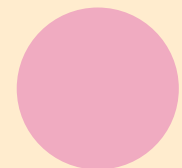
Con riferimento al file Malware_U3_W2_L5 presente all'interno della cartella «Esercizio_Pratico_U3_W2_L5 » sul desktop della macchina virtuale dedicata per l'analisi dei malware, rispondere ai seguenti quesiti:



1 Quali librerie vengono importate dal file eseguibile ? Fare anche una descrizione



2 Quali sono le sezioni di cui si compone il file eseguibile del malware? Fare anche una descrizione



Con riferimento alla figura nella prossima slide , risponde ai seguenti quesiti:



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



4 Ipotizzare il comportamento della funzionalità implementata



5 Fare una tabella per spiegare il significato delle singole righe di codice

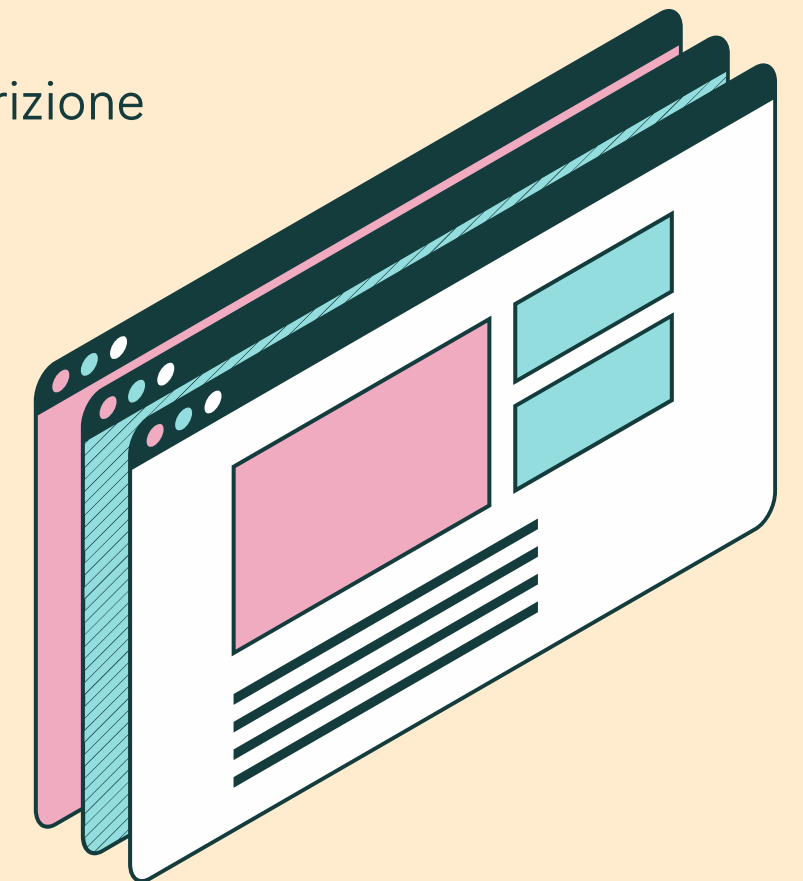
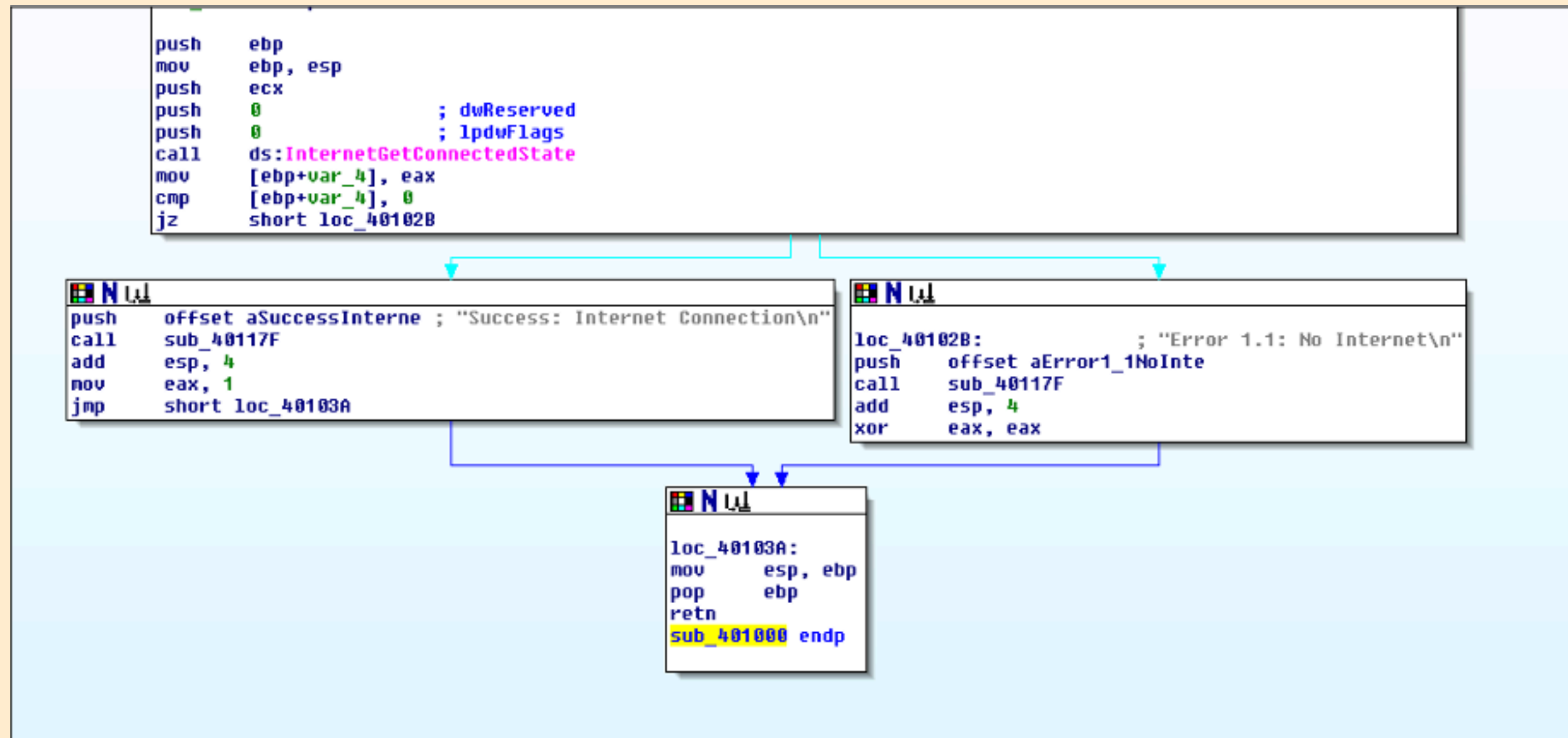


Figura di riferimento

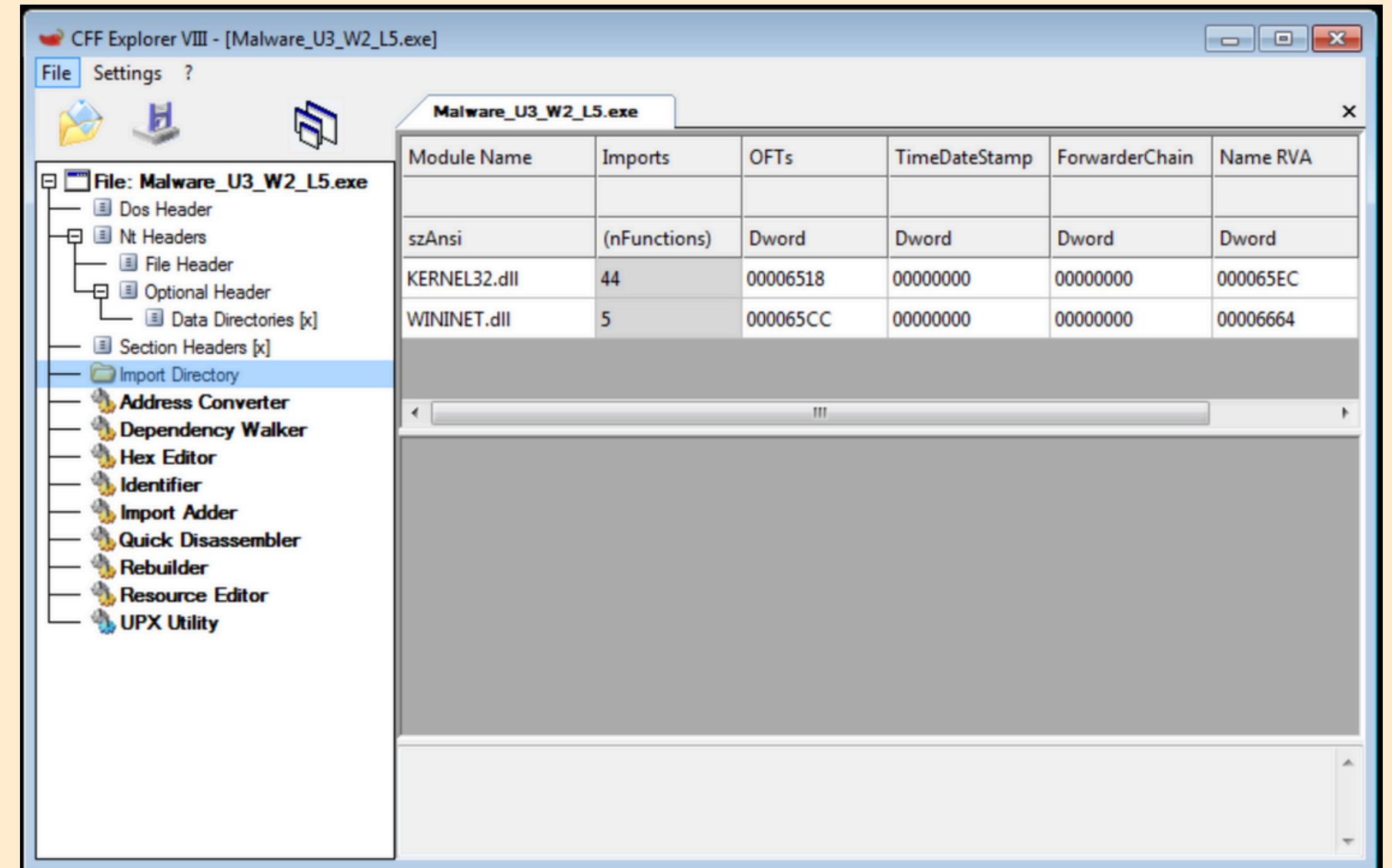


Esercizio I

Per eseguire questo punto dobbiamo aprire il nostro malware utilizzando CFF Explorer. Osserviamo, dalla sezione "import directory", che il Malware_U3_W2_L5 importa due librerie :

Kernel32.dll: Contiene le funzioni principali del sistema operativo, come la gestione della memoria, l'accesso ai file e la gestione dei processi e thread. È una libreria fondamentale per l'esecuzione di molte operazioni a livello di sistema

Wininet.dll: Include funzioni per la gestione delle comunicazioni di rete attraverso protocolli come HTTP, FTP e NTP. È utilizzata per implementare servizi di rete, inclusi download e upload di file, navigazione web e sincronizzazione di orologi di rete



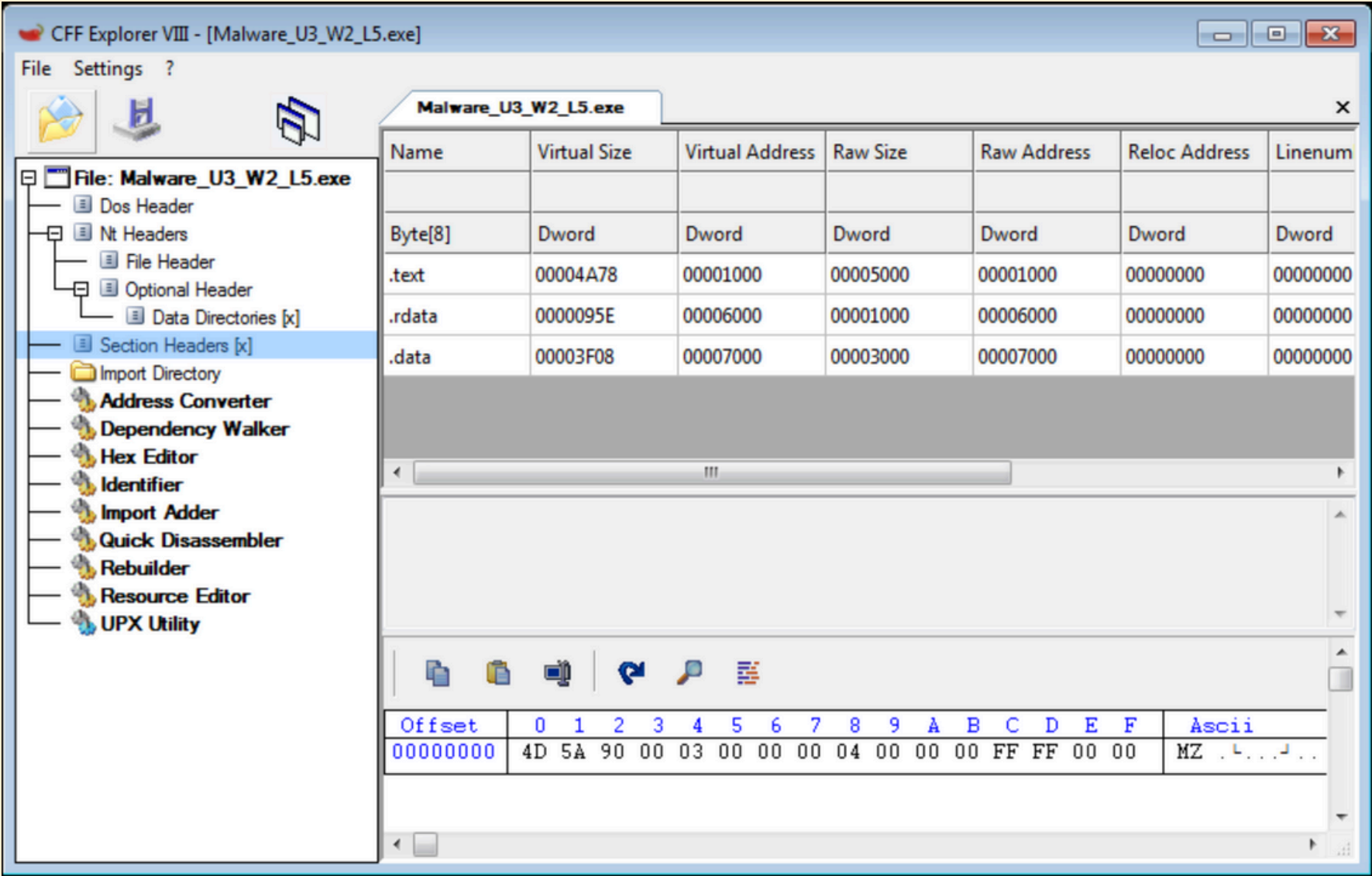
Esercizio 2

Continuando su CFF, nella sezione "section header", vediamo che l'eseguibile è composto da tre sezioni.

.text: Questa sezione contiene le istruzioni che la CPU eseguirà quando il software sarà avviato. Di solito, è l'unica parte di un file eseguibile eseguita dalla CPU, poiché le altre sezioni contengono dati o informazioni di supporto

.rdata: Questa sezione include generalmente informazioni sulle librerie e sulle funzioni importate ed esportate dall'eseguibile. Come abbiamo visto, queste informazioni possono essere ottenute con CFF Explorer

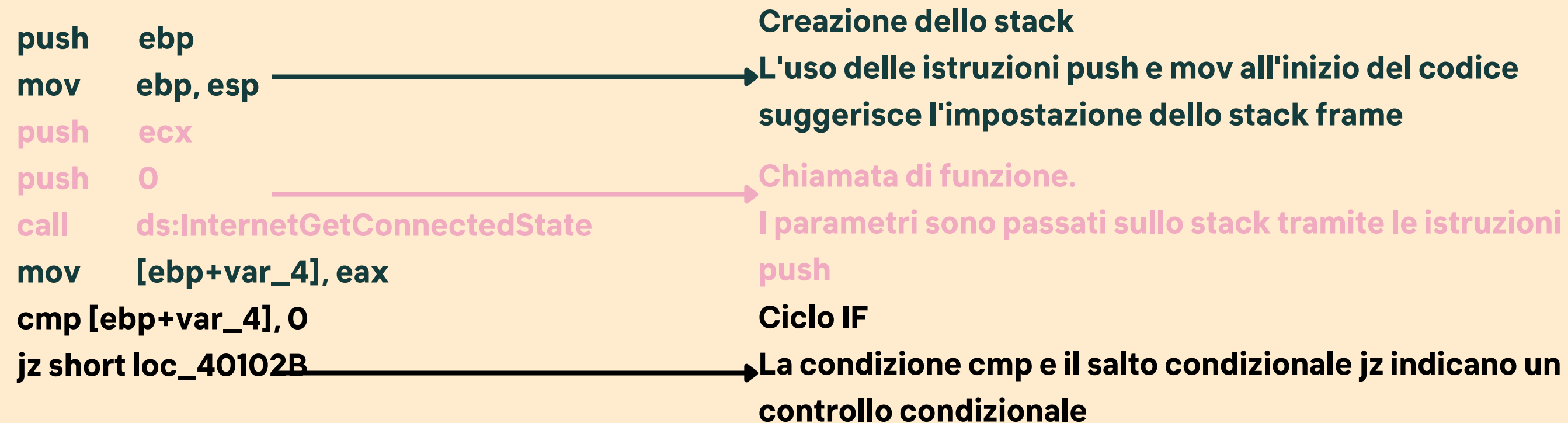
.data: Questa sezione contiene tipicamente i dati o le variabili globali del programma eseguibile, che devono essere accessibili da qualsiasi parte del programma. Una variabile è considerata globale quando non è definita all'interno di una funzione, ma dichiarata globalmente e, quindi, accessibile da qualsiasi funzione all'interno dell'eseguibile



Esercizio 3

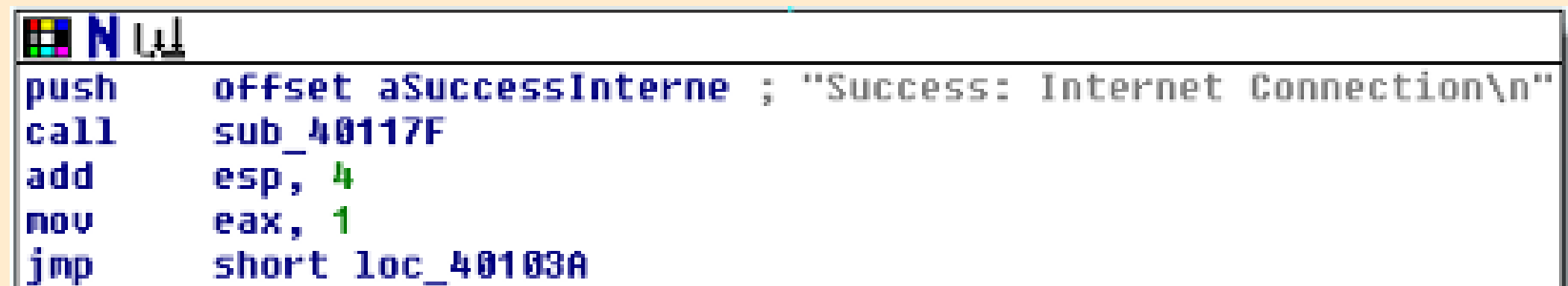
Cerchiamo di identificare i costrutti noti che vediamo in figura

```
push    ebp
mov     ebp, esp
push    ecx
push    0          ; dwReserved
push    0          ; lpdwFlags
call    ds:InternetGetConnectedState
mov     [ebp+var_4], eax
cmp     [ebp+var_4], 0
jz      short loc_40102B
```



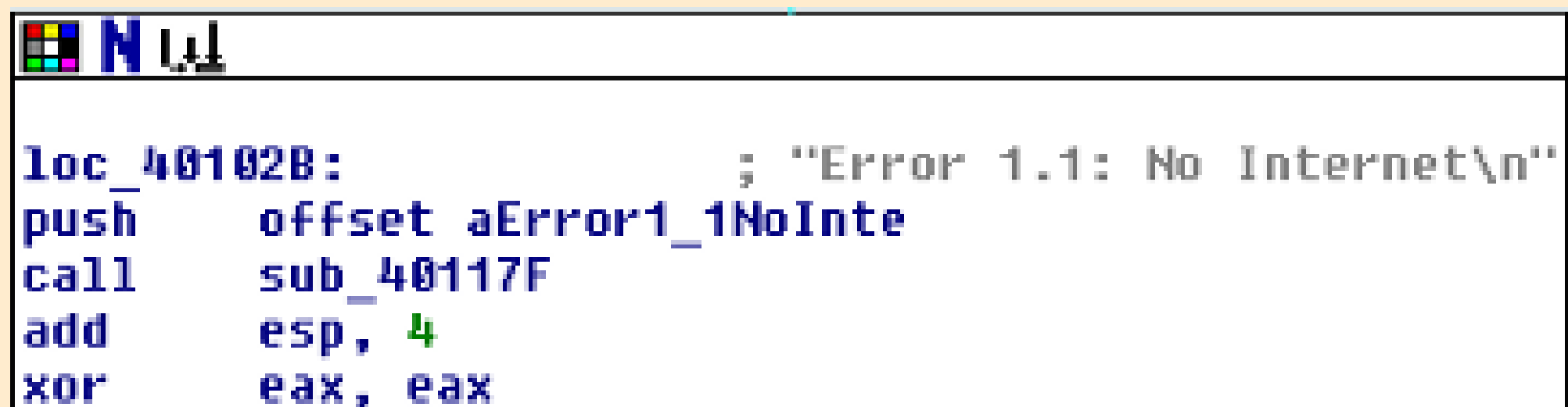
Esercizio 3

Cerchiamo di identificare i costrutti noti che vediamo in figura



```
push    offset aSuccessInterne ; "Success: Internet Connection\n"
call    sub_40117F
add     esp, 4
mov     eax, 1
jmp     short loc_40103A
```

**Chiamata di funzione che avviene se l'if
(visto nella slide precedente) da come
risposta positiva**

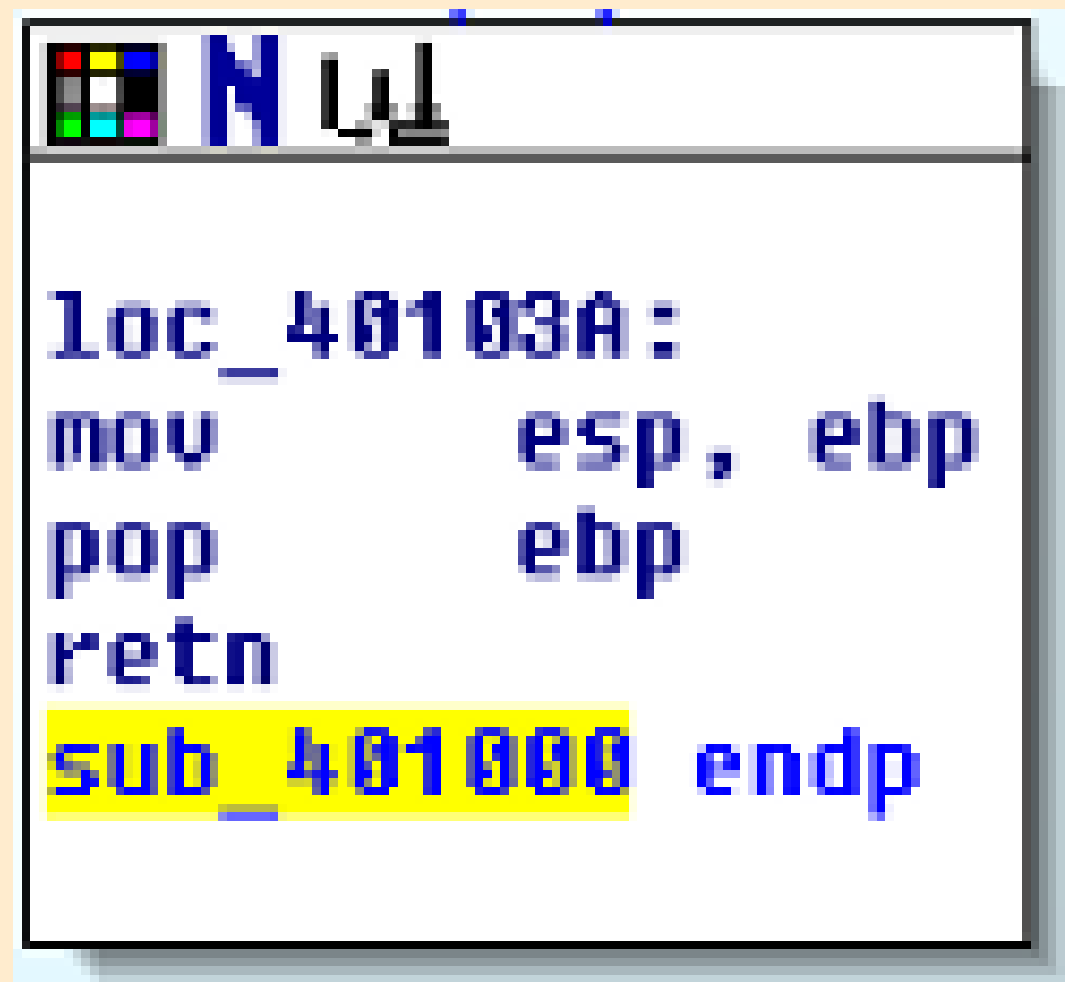


```
loc_40102B:                                ; "Error 1.1: No Internet\n"
push    offset aError1_1NoInte
call    sub_40117F
add     esp, 4
xor     eax, eax
```

**Chiamata di funzione che avviene se l'if
(visto nella slide precedente) da come
risposta negativa**

Esercizio 3

Cerchiamo di identificare i costrutti noti che vediamo in figura

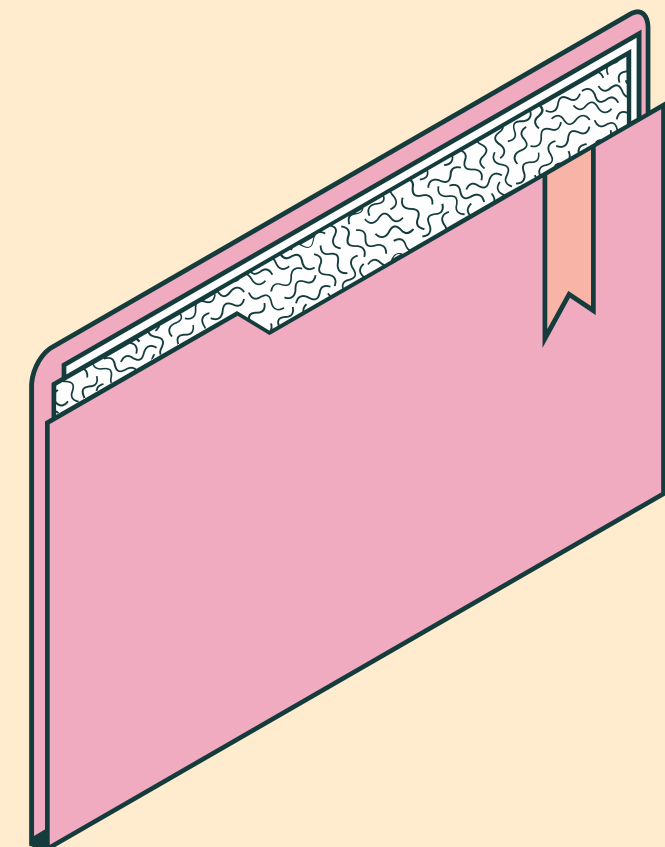


Ripristino del Puntatore dello Stack

Questo passaggio è parte del processo di ripristino dello stack frame alla fine di una funzione

Chiusura della Procedura

Indica la fine del blocco di codice della procedura



Esercizio 4

Ipotizzare il comportamento della funzionalità implementata

Il programma in questione è un semplice controllo della connessione internet con messaggi di stato.

Il programma controlla lo stato della connessione internet e produce un messaggio di successo o errore a seconda del risultato. Utilizza pratiche standard di gestione dello stack per assicurare che il contesto del chiamante sia ripristinato correttamente dopo l'esecuzione della procedura.

In breve

- Imposta lo stack frame.
- Controlla lo stato della connessione internet.
- Mostra messaggi di successo o errore in base al risultato.
- Ripristina lo stack frame e termina la funzione

```
push    ebp
mov     ebp, esp
push    ecx
push    0          ; dwReserved
push    0          ; lpdwFlags
call    ds:InternetGetConnectedState
mov     [ebp+var_4], eax
cmp     [ebp+var_4], 0
jz      short loc_40102B
```

```
push    offset aSuccessInterne ; "Success: Internet Connection\n"
call    sub_40117F
add     esp, 4
mov     eax, 1
jmp     short loc_40103A
```

```
loc_40102B:          ; "Error 1.1: No Internet\n"
push    offset aError1_1NoInte
call    sub_40117F
add     esp, 4
xor     eax, eax
```

```
loc_40103A:
mov     esp, ebp
pop     ebp
retn
sub_401000 endp
```

Esercizio 5

Fare una tabella per spiegare il significato delle singole righe di codice

Istruzione	Descrizione
push ebp	Salva il contenuto del registro ebp sullo stack. Questo è parte della creazione dello stack frame per la funzione
mov ebp, esp	Copia il valore del registro esp nel registro ebp. Questo imposta il registro ebp come il nuovo frame base
push ecx	Salva il contenuto del registro ecx sullo stack.
push 0	Passa il valore 0 come argomento alla funzione InternetGetConnectedState (dwReserved)
push 0	Passa il valore 0 come argomento alla funzione InternetGetConnectedState (lpdwFlags)

Esercizio 5

Fare una tabella per spiegare il significato delle singole righe di codice

Istruzione	Descrizione
call ds:InternetGetConnectedState	Chiama la funzione InternetGetConnectedState che verifica lo stato della connessione internet. I risultati della funzione vengono messi nel registro eax
mov [ebp+var_4], eax	Sposta il valore del registro eax (risultato della funzione chiamata) nella variabile locale var_4 situata nello stack frame
cmp [ebp+var_4], 0	Confronta il valore di var_4 con 0 per verificare se la connessione internet è presente o meno
jz short loc_40102B	Se var_4 è zero (indicando che non c'è connessione internet), salta all'etichetta loc_40102B

Esercizio 5

CONNESSIONE PRESENTE

Istruzione	Descrizione
push offset aSuccessInterne ; "Success: Internet Connection\n"	Stampa la stringa di successo ("Success: Internet Connection") sullo stack
call sub_40117F	Chiama la funzione sub_40117F
add esp, 4	Ripulisce lo stack rimuovendo l'indirizzo della stringa precedente
mov eax, 1	Imposta il registro eax a 1, indicando uno stato di successo
jmp short loc_40103A	Salta all'etichetta loc_40103A, evitando il blocco di errore

Esercizio 5

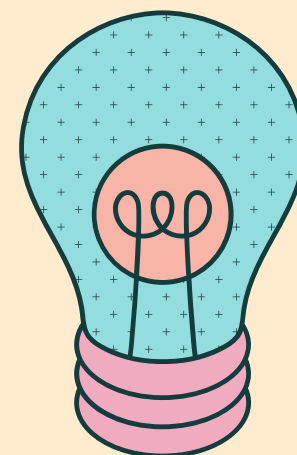
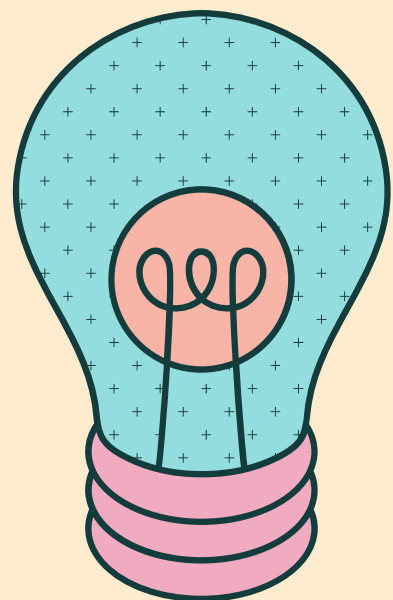
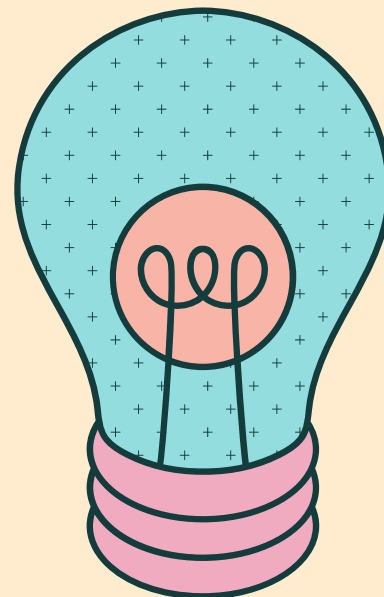
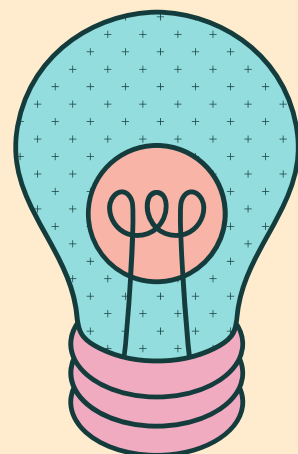
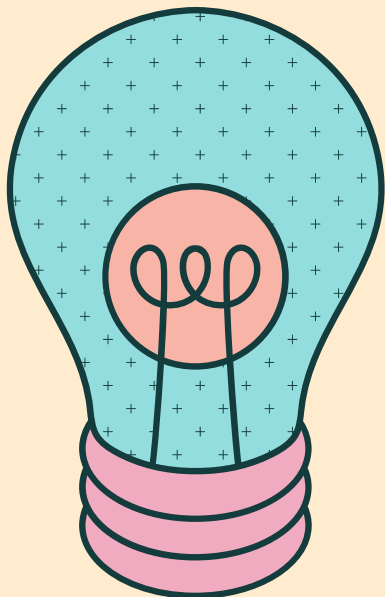
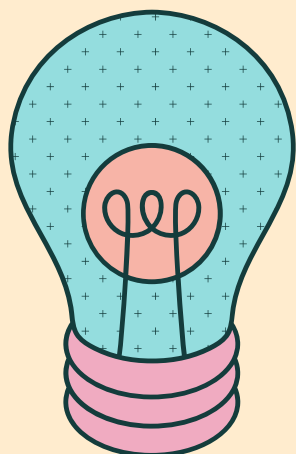
CONNESSIONE ASSENTE

Istruzione	Descrizione
loc_40102B:	Etichetta che indica l'inizio del blocco di codice per gestire l'errore (nessuna connessione internet)
push offset aError1_1NoInte ; "Error 1.1: No Internet\n"	Stampa la stringa di errore ("Error 1.1: No Internet") sullo stack
call sub_40117F	Chiama la funzione sub_40117F
add esp, 4	Ripulisce lo stack rimuovendo l'indirizzo della stringa precedente
xor eax, eax	Esegue un'operazione XOR su eax con se stesso, azzerando eax (indicando uno stato di errore).

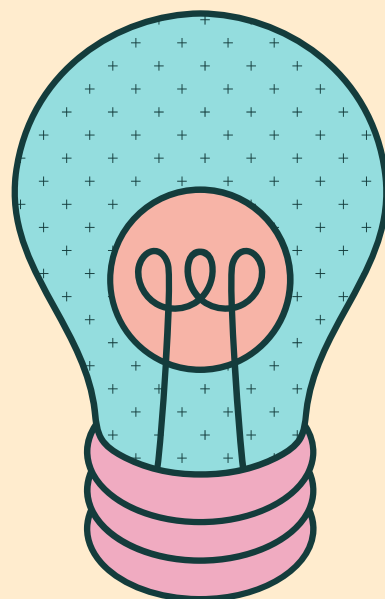
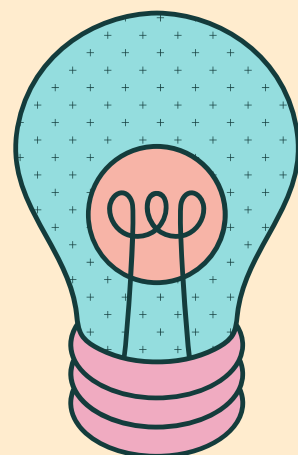
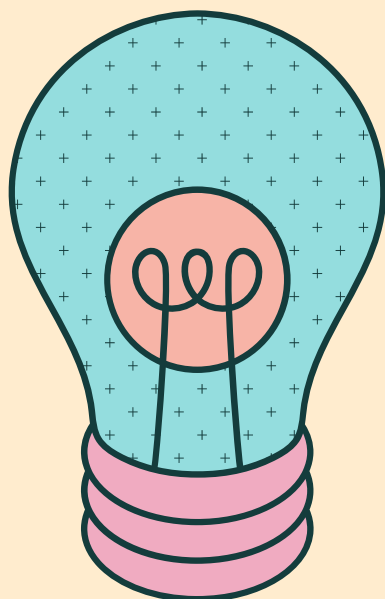
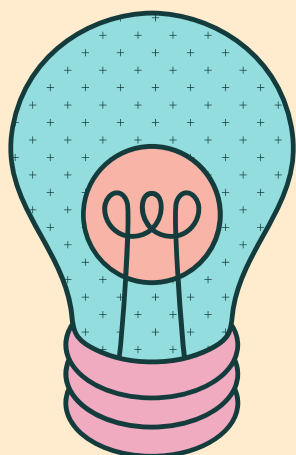
Esercizio 5

Finale

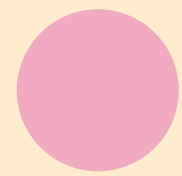
Istruzione	Descrizione
loc_40103A:	Etichetta che indica l'inizio del blocco di codice finale
mov esp, ebp	Ripristina il puntatore dello stack esp al valore del frame base ebp
pop ebp	Ripristina il valore originale di ebp dallo stack.
ret	Ritorna dalla funzione
sub_401000 endp	Indica la fine della procedura sub_401000



BONUS



Traccia



Un giovane dipendente neo assunto segnala al reparto tecnico la presenza di un programma sospetto. Il suo superiore gli dice di stare tranquillo ma lui non è soddisfatto e chiede supporto al SOC.

Il file "sospetto" è iexplore.exe contenuto nella cartella C:\Programmi\Internet Explorer (no, non ridete ragazzi)

Come membro senior del SOC ti è richiesto di convincere il dipendente che il file non è maligno.

Possono essere usati gli strumenti di analisi statica basica e/o analisi dinamica basica visti a lezione.

No disassembly no debug o similari

VirusTotal non basta, ovviamente

Non basta dire iexplorer è Microsoft quindi è buono, punto



Esercizio Bonus

Analisi statica

Perchè statica?

Sebbene l'analisi dinamica offra una visione del comportamento del file durante l'esecuzione, l'analisi statica fornisce un'alternativa sicura e immediata per verificare file come `iexplorer.exe`, specialmente quando si cerca di evitare i rischi associati all'esecuzione di software potenzialmente sospetto.



Esercizio Bonus

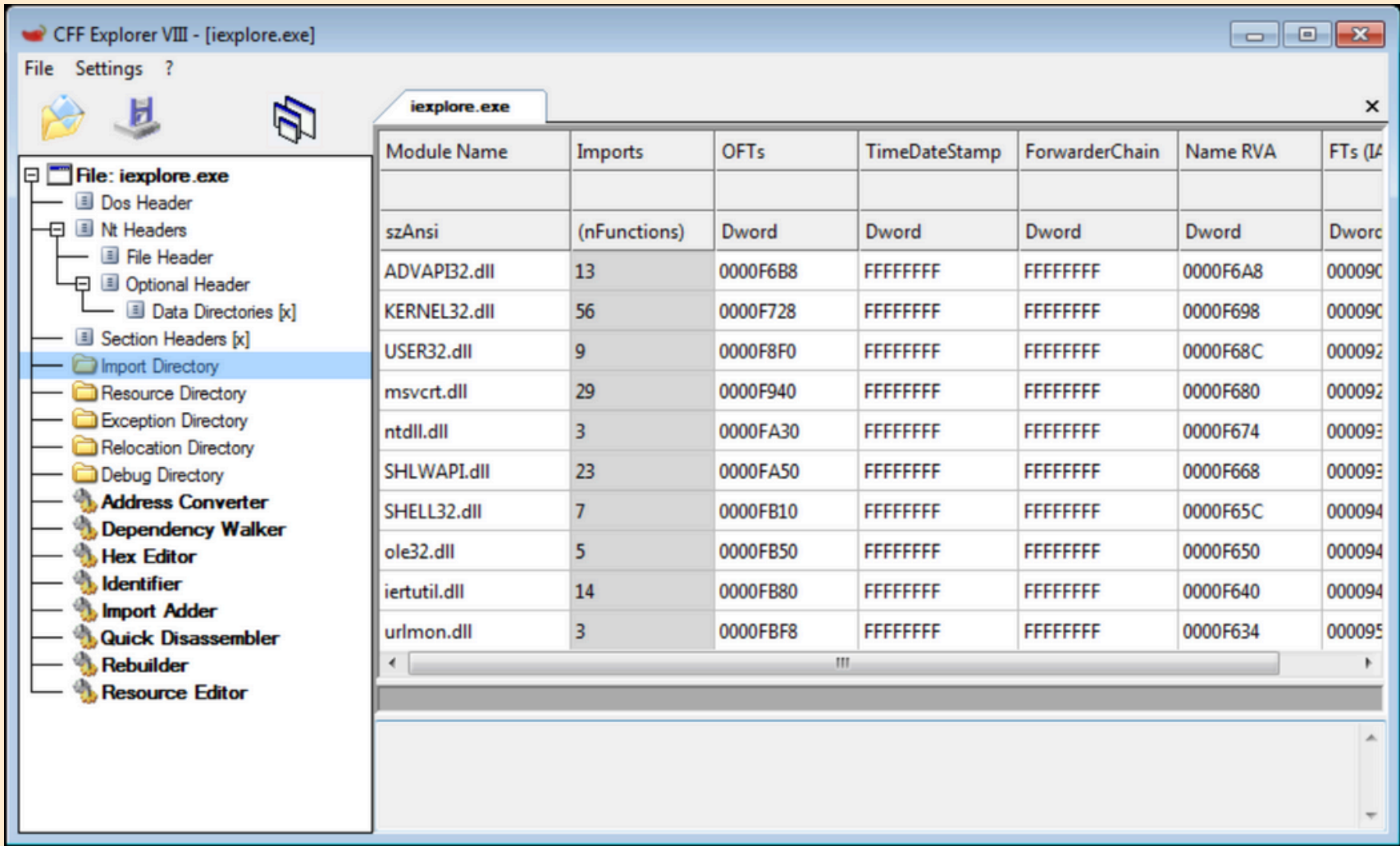
Analisi statica

Metodologia

CFF Explorer è stato utilizzato per analizzare il file iexplorer.exe sotto il profilo strutturale e per individuare eventuali anomalie che potrebbero indicare attività malevole.

Sono state esaminate le sezioni del file per verificare la presenza di anomalie. In un file legittimo, le sezioni come .text, .data, e .rsrc devono essere presenti e strutturate correttamente. In oltre, spesso in un malware queste sezioni sono difficilmente leggibili o se ne trovano alcune sospette. In questo caso erano leggibili ed in chiaro.

Sono state anche controllate le librerie che iexplorer.exe usa per controllare che non ci fosse qualche attività sospetta.



Module Name	Imports	OFs	TimeDateStamp	ForwarderChain	Name RVA	FTs (I)
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
ADVAPI32.dll	13	0000F6B8	FFFFFFFF	FFFFFFFF	0000F6A8	00009C
KERNEL32.dll	56	0000F728	FFFFFFFF	FFFFFFFF	0000F698	00009C
USER32.dll	9	0000F8F0	FFFFFFFF	FFFFFFFF	0000F68C	000092
msvcrt.dll	29	0000F940	FFFFFFFF	FFFFFFFF	0000F680	000092
ntdll.dll	3	0000FA30	FFFFFFFF	FFFFFFFF	0000F674	000093
SHLWAPI.dll	23	0000FA50	FFFFFFFF	FFFFFFFF	0000F668	000093
SHELL32.dll	7	0000FB10	FFFFFFFF	FFFFFFFF	0000F65C	000094
ole32.dll	5	0000FB50	FFFFFFFF	FFFFFFFF	0000F650	000094
iertutil.dll	14	0000FB80	FFFFFFFF	FFFFFFFF	0000F640	000094
urlmon.dll	3	0000FBF8	FFFFFFFF	FFFFFFFF	0000F634	000095

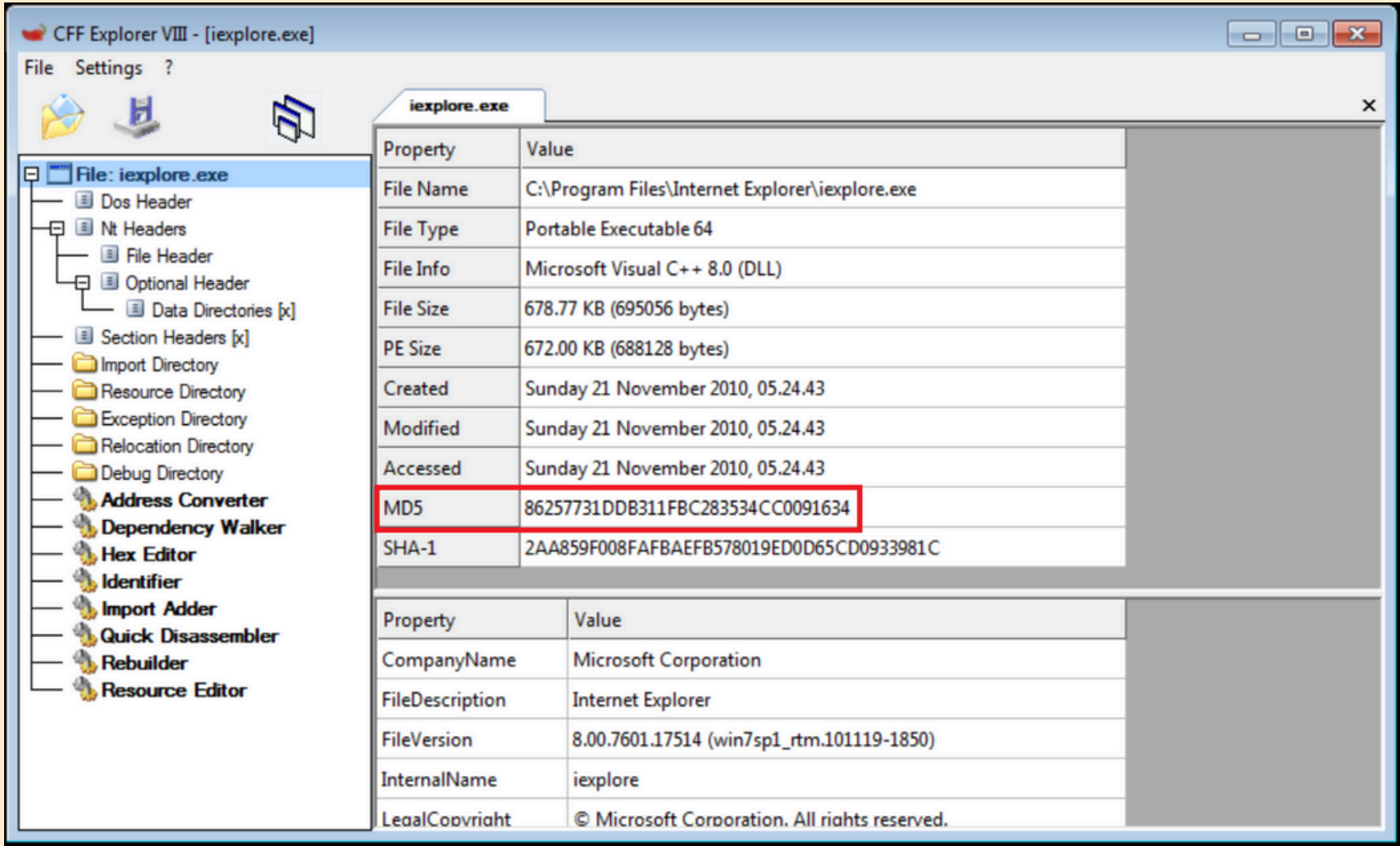
Esercizio Bonus

Analisi statica

Continuando su CFF, si può notare , che , tra le varie caratteristiche dell'eseguibile, c'è anche il suo hash. Quest'ultimo può essere utilizzato:

- Come se fosse un etichetta per l'identificazione univoca del file
- Condiviso con analisti della sicurezza per aiutarli nell'identificazione del malware
- Cercare l'hash online per confermare che sia un malware e magari per trovare delle informazioni circa il suo comportamento

Quest'ultimo punto è il nostro prossimo passo

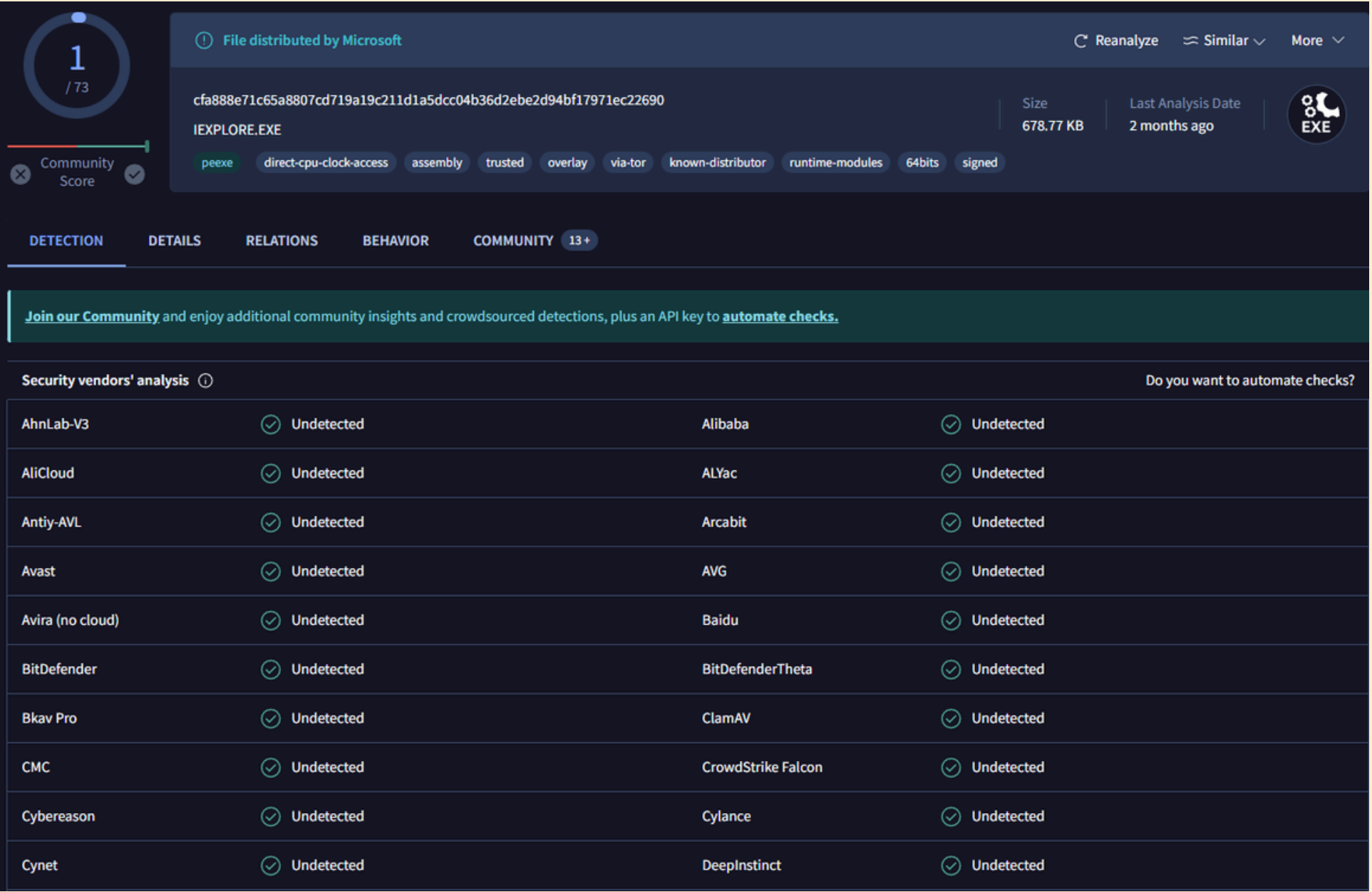
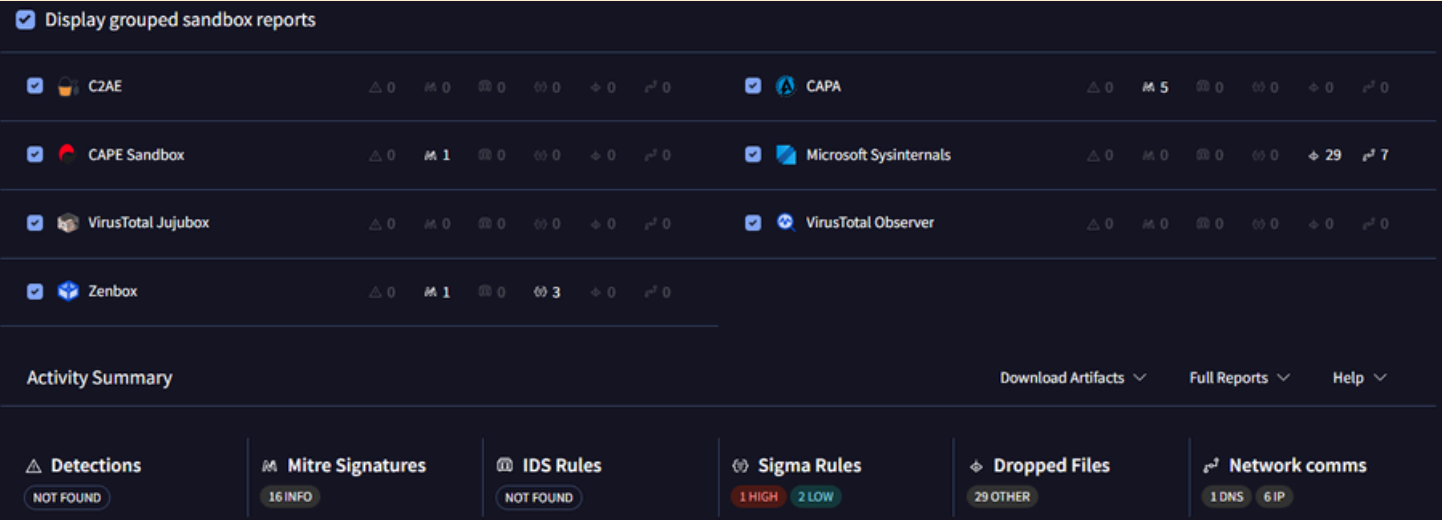


Esercizio Bonus

Analisi statica

Grazie all’hash trovato possiamo caricarlo sulla piattaforma VirusTotal che ci mostra che l’eseguibile è innocuo.

VirusTotal utilizza una vasta gamma di motori antivirus e strumenti di analisi per esaminare il file. Questi motori sono aggiornati per rilevare le minacce più recenti e forniscono rapporti su eventuali rilevamenti di malware.

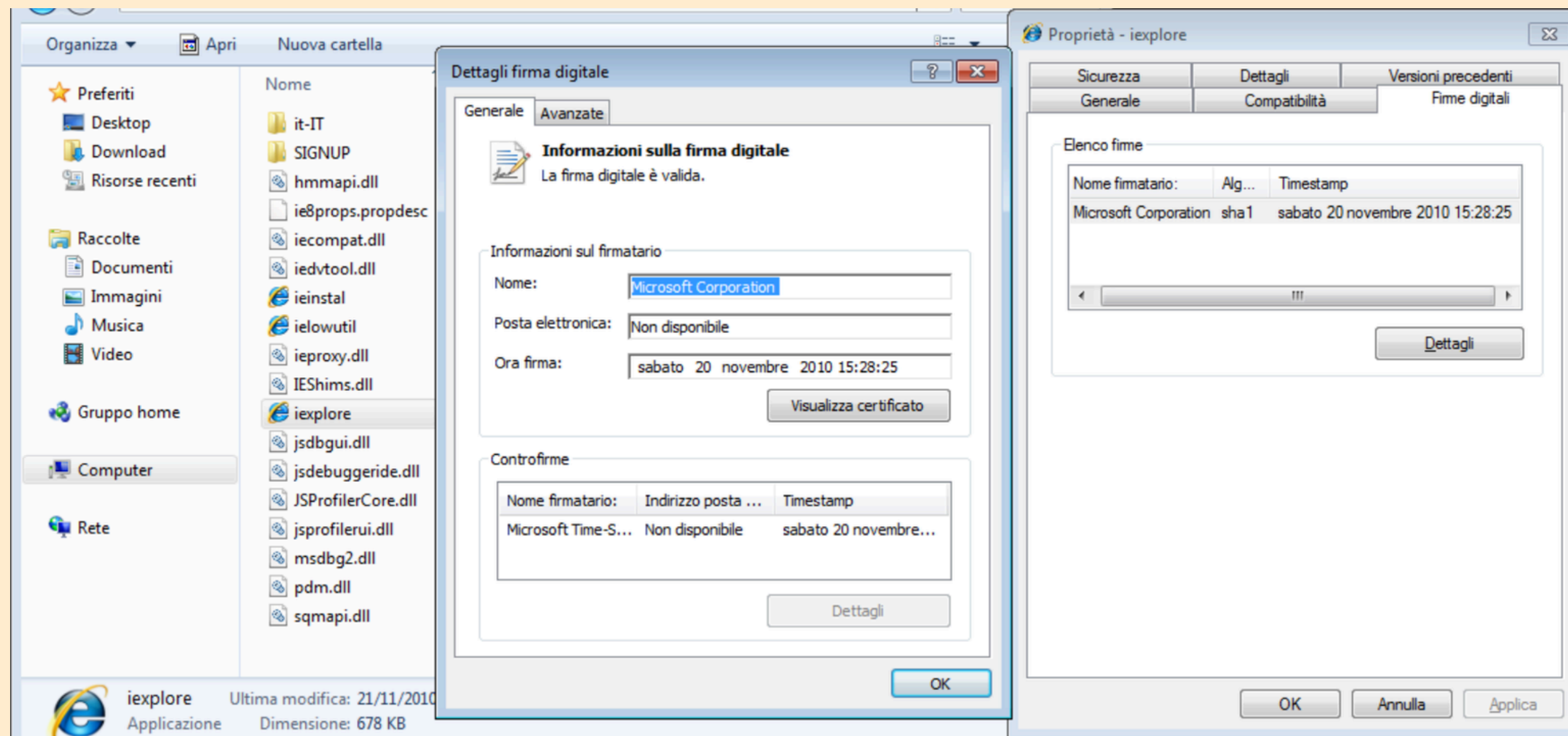


Esercizio Bonus

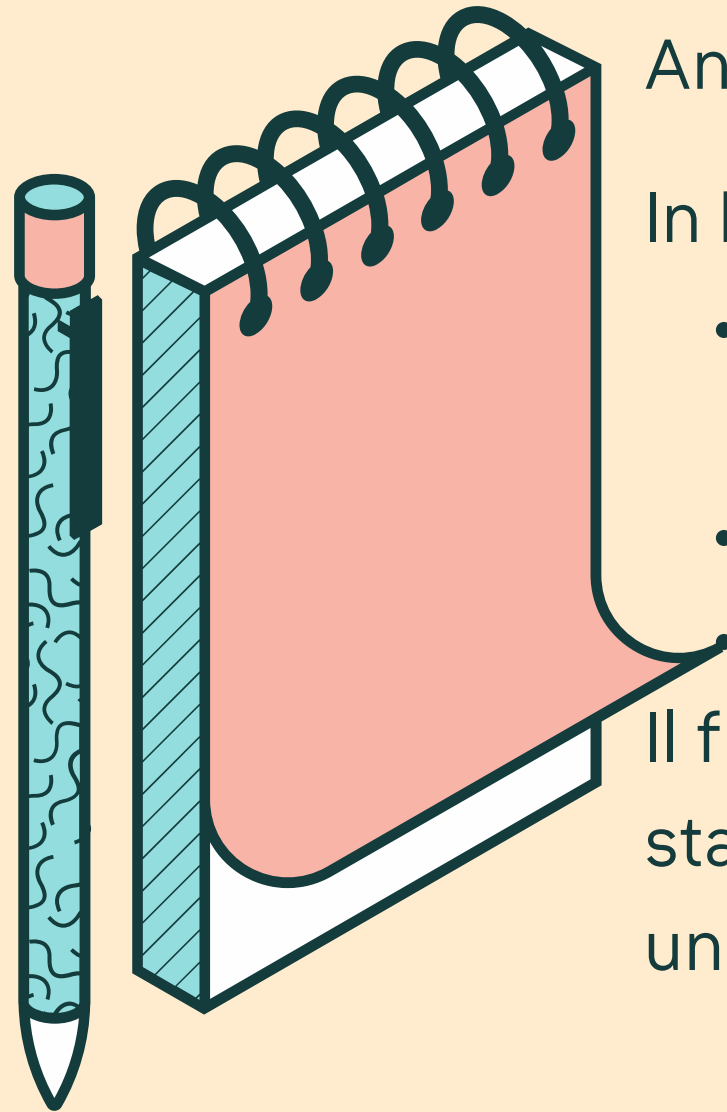
Analisi statica

E' stata eseguita la verifica della firma digitale per confermare l'autenticità del file e garantirne l'integrità.

È stata esaminata la sezione "Dettagli" delle proprietà per confermare la presenza di una firma digitale valida. Questo perchè spesso nei malware la firma digitale è assente



Esercizio Bonus

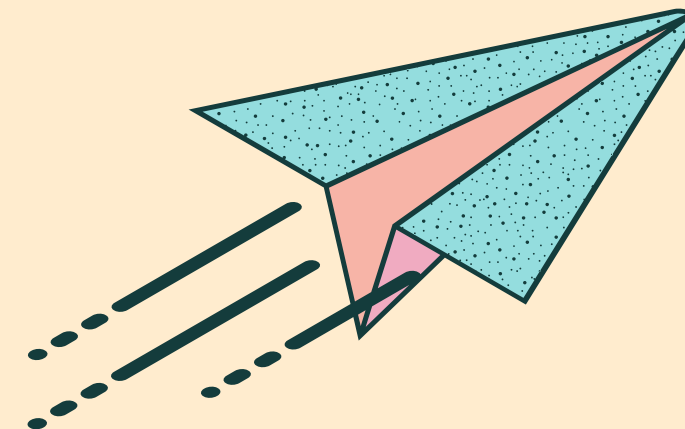
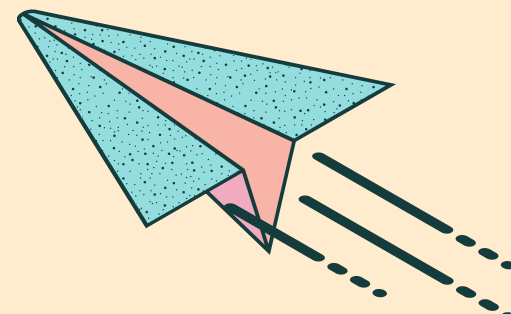
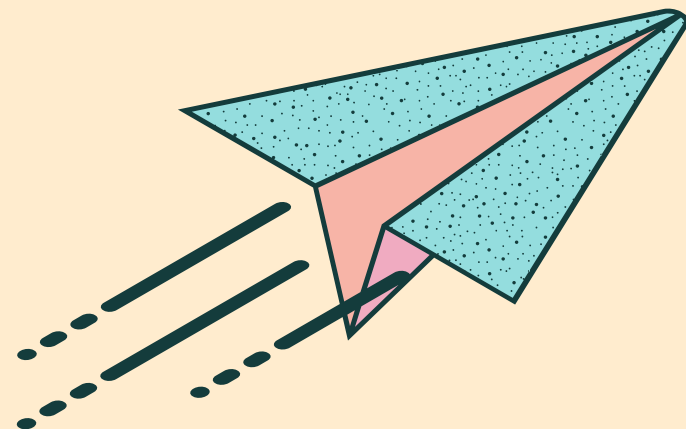
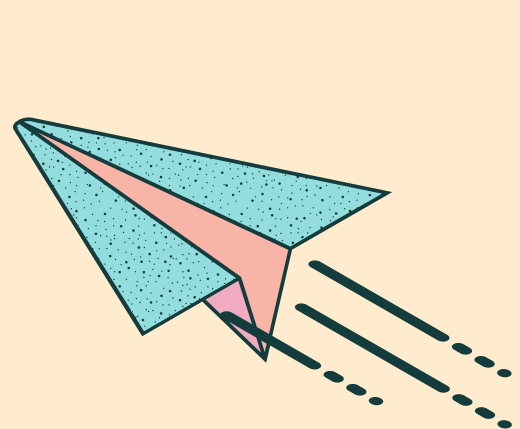


Analisi statica

In base alle analisi eseguite:

- CFF Explorer: Non sono state trovate anomalie strutturali nel file iexplorer.exe.
- VirusTotal: Il file è risultato pulito da tutte le scansioni antivirus.
- Firma Digitale: Il file è firmato digitalmente e la firma è risultata valida.

Il file iexplorer.exe analizzato è un file legittimo e non maligno. Non sono state trovate evidenze che suggeriscano un comportamento malevolo o una compromissione della sicurezza.



**GRAZIE PER
L'ATTENZIONE**

