



Cybersecurity Specialist

# Cyber Security & Ethical Hacking

## GIORNO 5 – PROGETTO

---

Alejandro Cristino  
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?
3. Con riferimento alla figura in slide 3, risponde ai seguenti quesiti:
4. Identificare i costrutti noti (creazione dello stack, eventuali cicli, altri costrutti).
5. Ipotizzare il comportamento della funzionalità implementata.
6. Fare una tabella per spiegare il significato delle singole righe di codice.

# 1.Quali librerie vengono importate dal file eseguibile?

Per prima cosa importiamo il nostro file eseguibile Esercizio\_Pratico\_U3\_W2\_L5 su CFF Explorer, un tool utile che ci permette di visualizzare e modificare dettagliatamente le strutture interne di un file eseguibile. E' particolarmente utile per chi lavora nell'ambito della sicurezza informatica e del reverse engineering.

Per analizzare le librerie importate, andiamo su import directory:

The screenshot shows the CFF Explorer interface. On the left is a tree view of the file structure for 'File: Malware\_U3\_W2\_L5.exe', with the 'Import Directory' node selected. A red arrow points from the 'Import Directory' node towards the table below. On the right is a table titled 'Malware\_U3\_W2\_L5.exe' showing imported modules. Two red arrows point from the bottom-left towards the first two rows of the table.

Module Name	Imports	OFTs	TimeStamp	ForwarderChain	Name
szAnsi	(nFunctions)	Dword	Dword	Dword	Dwo
KERNEL32.dll	44	00006518	00000000	00000000	00000000
WININET.dll	5	000065CC	00000000	00000000	00000000

# 1.1.Spiegazione delle librerie

## 1.KERNEL32.dll:

- **Descrizione:** KERNEL32.dll è una delle principali librerie del sistema operativo Windows. Fornisce un'ampia gamma di funzioni di base per la gestione del sistema operativo, inclusi la gestione della memoria, l'input/output (I/O) di file, la gestione dei processi e dei thread, le operazioni di sincronizzazione, e molto altro.
- **Importanza:** È una libreria cruciale per il funzionamento delle applicazioni Windows, in quanto fornisce molte delle API di basso livello necessarie per eseguire operazioni comuni.

## 2.WININET.dll:

- **Descrizione:** WININET.dll è una libreria che fornisce un'interfaccia per le applicazioni Windows per interagire con le risorse di rete tramite i protocolli HTTP e FTP. Include funzioni per la gestione delle connessioni di rete, l'invio di richieste HTTP, il download e l'upload di file via FTP, e altre operazioni legate alla rete.
- **Importanza:** È utilizzata da molte applicazioni per implementare funzionalità di rete, come il browser web Internet Explorer e altre applicazioni che necessitano di comunicare su Internet.

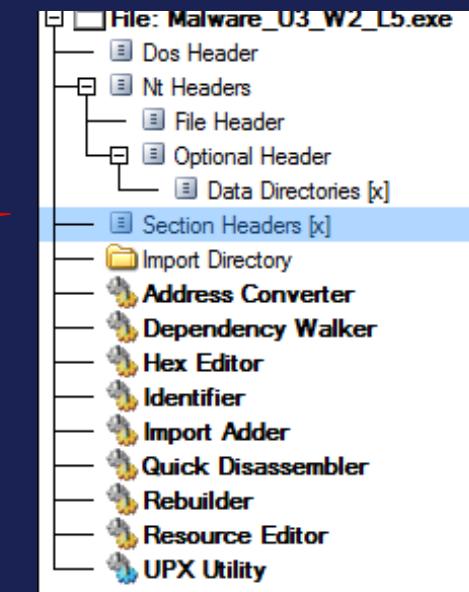
## 2.Quali sono le sezioni di cui si compone il file eseguibile del malware?

Vediamo ora le sezioni di un file eseguibile, spesso in formato PE (Portable Executable). Queste sezioni sono aree di memoria organizzate all'interno del file e contengono diverse tipologie di dati e istruzioni necessarie per il funzionamento del programma.

Per analizzare le sezioni, andiamo su section headers:



Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N
00000208	00000210	00000214	00000218	0000021C	00000220	00000224	00000228
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word
.text	00004A78	00001000	00005000	00001000	00000000	00000000	0000
.rdata	0000095E	00006000	00001000	00006000	00000000	00000000	0000
.data	00003F08	00007000	00003000	00007000	00000000	00000000	0000



## 2.1. Spiegazione delle sezioni

- **.text:** Contiene le istruzioni (le righe di codice) che la CPU eseguirà una volta che il software sarà avviato.  
Generalmente questa è l'unica sezione di un file eseguibile che viene eseguita dalla CPU, in quanto tutte le altre sezioni contengono dati o informazioni a supporto.
- **r.data:** Include generalmente le informazioni circa le librerie e le funzioni importate ed esportate dall'eseguibile, informazioni che possiamo ricavare con CFF Explorer.
- **.data:** Contiene tipicamente i dati/le variabili globali del programma eseguibile, che devono essere disponibili da qualsiasi parte del programma. Una variabile si dice globale quando non è definita all'interno di un contesto di una funzione, ma bensì è globalmente dichiarata ed è di conseguenza accessibile da qualsiasi funzione all'interno dell'eseguibile.

## 2.2.Ricerca su virus total

Effettuando una ricerca su VirusTotal tramite l'hash ricavato da CFF Explorer, dal risultato emerge che il malware analizzato potrebbe essere un Trojan.

40 / 74

Community Score

① 40/74 security vendors flagged this file as malicious

b71777edbf21167c96d20ff803cbcb25d24b94b3652db2f286cd6efd3d8416a

Malware\_U3\_W2\_L5.exe

Size 40.00 KB | Last Analysis Date 14 days ago | EXE

peexe direct-cpu-clock-access runtime-modules armadillo checks-network-adapters

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 7

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

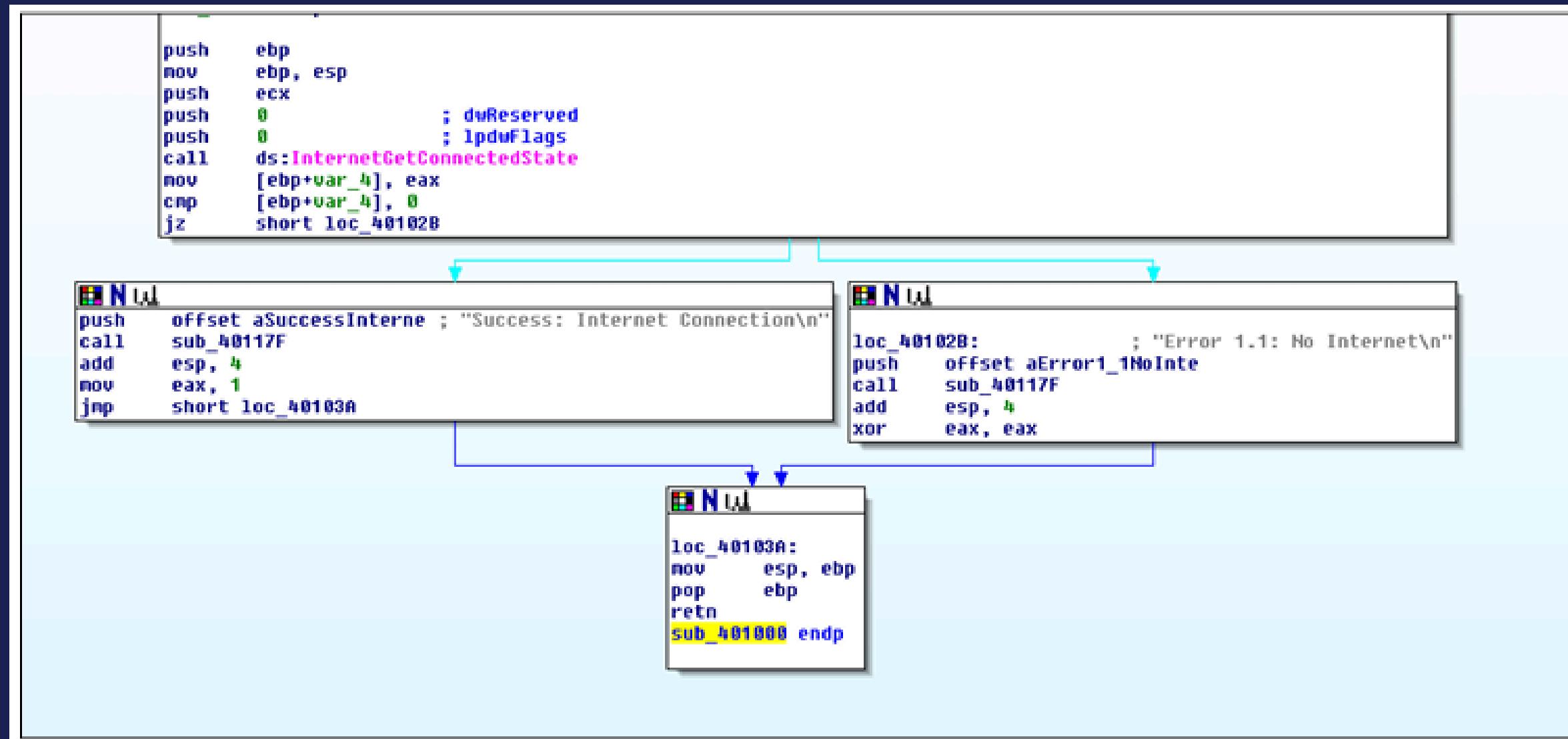
Popular threat label ① trojan.r002c0pdm21/ymacco Threat categories trojan Family labels r002c0pdm21 ymacco

Security vendors' analysis ① Do you want to automate checks?

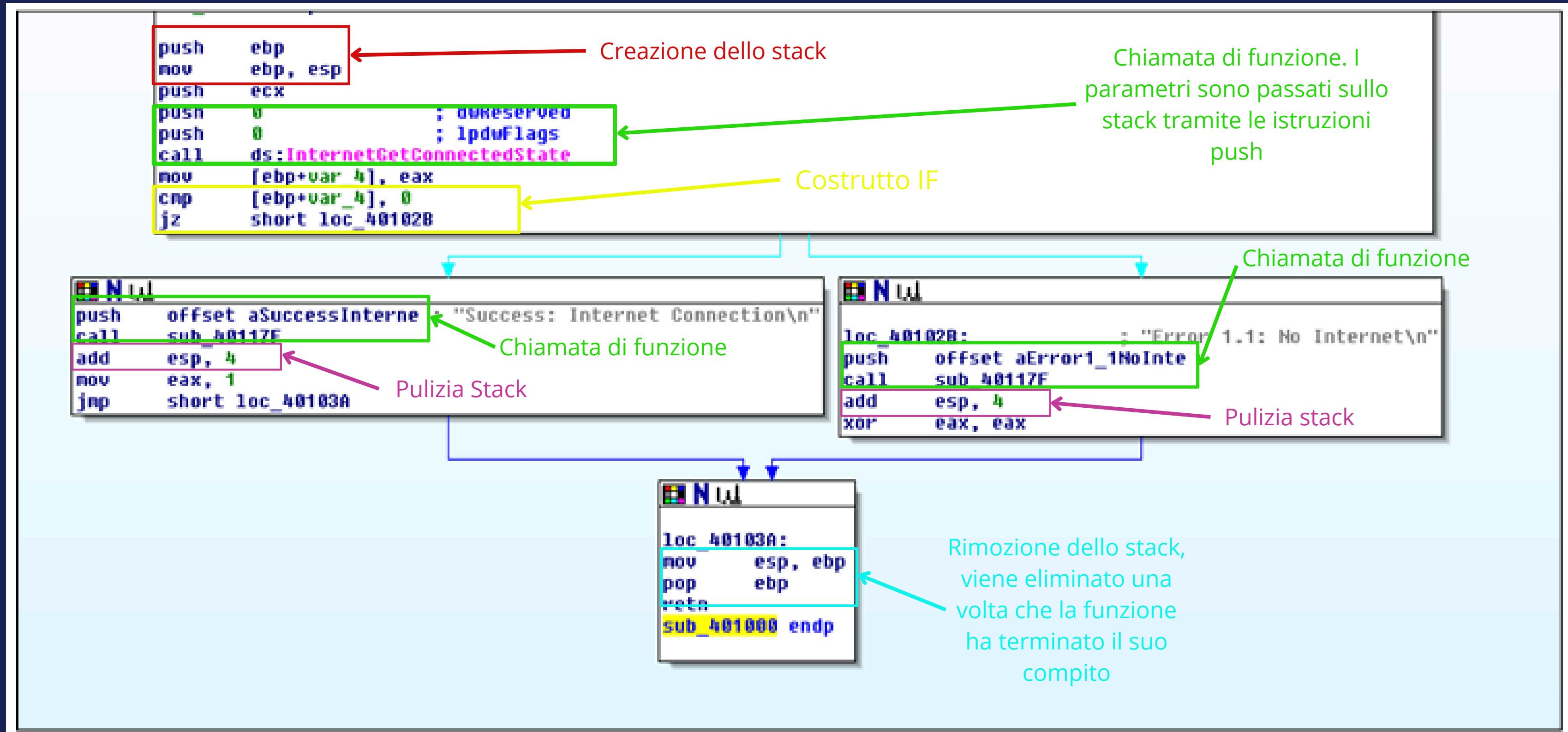
Alibaba	① Trojan:Win32/Generic.2cc376c1	AliCloud	① Trojan:Win/Ymacco.AMH1
Antiy-AVL	① Trojan/Win32.BTSGeneric	Avast	① Win32:PUP-gen [PUP]
AVG	① Win32:PUP-gen [PUP]	CrowdStrike Falcon	① Win/malicious_confidence_100% (W)
Cylance	① Unsafe	DeepInstinct	① MALICIOUS

### 3. Con riferimento alla figura in slide, risponde ai seguenti quesiti:

- Identificare i costrutti noti (creazione dello stack, eventuali cicli, altri costrutti).
- Ipotizzare il comportamento della funzionalità implementata.
- Fare una tabella per spiegare il significato delle singole righe di codice.



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



## 5. Ipotizzare il comportamento della funzionalità implementata.

Da questa porzione di codice assembly, possiamo dedurre che il malware invoca la funzione InternetGetConnectedState. Questa funzione restituisce un valore che può essere "0" o diverso da "0". Utilizzando un'istruzione condizionale IF, il programma determina se è presente una connessione a Internet. Se la connessione è attiva, verrà visualizzato il messaggio: "Success: Internet connection". Altrimenti, apparirà il messaggio: "Error 1.1: No Internet".

Pseudocodice C:

```
1 int main()
2 {
3     state = internetGetConnectedState(par1, 0, 0);
4
5     if (state != 0)
6     {
7         printf("success, internet connection\n");
8     }
9     else
10    {
11        printf("error 1.1: no internet\n");
12    }
13
14    return 0;
15 }
```

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

push ebp	Salva il valore del registro di base (EBP) nello stack.
mov ebp, esp	Crea un nuovo frame della pila, copiando il valore di "esp" (puntatore alla cima dello stack) nel registro di base "ebp".
push ecx	Salva il valore del registro ECX nello stack, al fine di conservare il valore originale del registro prima di eventuali modifiche.
push 0 ; dwReserved	Inserisce il valore 0 nello stack. Questo verrà utilizzato come parametro dwReserved per la chiamata successiva.
push 0 ; lpdwFlags	Inserisce nuovamente il valore 0 nello stack. Questo sarà il parametro lpdwFlags per la chiamata successiva.
call ds:InternetGetConnectedState	Chiama la funzione InternetGetConnectedState, per la quale i parametri sono stati posti precedentemente nello stack.
mov [ebp+var_4], eax	Memorizza il risultato della chiamata InternetGetConnectedState nella variabile locale [ebp+var_4].
cmp [ebp+var_4], 0	Confronta il valore memorizzato in [ebp+var_4] con 0.
jz short loc_40102B	Se il risultato della comparazione è zero, salta a loc_40102B.
push offset aSuccessInterne ; "Success: Internet Connection\n"	Inserisce l'indirizzo della stringa "Success: Internet Connection\n" nello stack come argomento per la successiva chiamata.
call sub_40117F	Chiama la funzione sub_40117F.
add esp, 4	Libera lo spazio nello stack precedentemente occupato dai parametri della chiamata.
mov eax, 1	Imposta eax a 1.
jmp short loc_40103A	Salta a loc_40103A.
loc_40102B : ; "Error 1.1: No Internet\n"	Posizione a cui salta il programma in caso di assenza di connessione ad internet.

push offset aError1_1_NoInte	Inserisce l'indirizzo della stringa "Error 1.1: No Internet\n" nello stack come argomento per la successiva chiamata.
call sub_40117F	Chiama la funzione sub_40117F.
add esp, 4	Libera lo spazio nello stack precedentemente occupato dai parametri della chiamata.
xor eax, eax	Esegue un XOR tra eax e se stesso, azzera il registro.
loc_40103A:	Posizione a cui il programma salta in ogni caso.
mov esp, ebp	Ripristina il valore originale di esp dalla variabile ebp.
pop ebp	Ripristina il valore originale del registro di base ebp dalla pila.
retn	Restituisce il controllo alla funzione chiamante.
sub_401000 endp	Questa riga indica la fine della procedura.



Cybersecurity Specialist

# Cyber Security & Ethical Hacking

GRAZIE.