Analisi statica del malware

Punto 1) Identificazione librerie importate

Utilizzando il tool CFF explorer possiamo trovare le librerie importate dal malware:

KERNEL32.dll	44	00006518	00000000	00000000	000065EC	00006000
WININET.dll	5	000065CC	00000000	00000000	00006664	000060B4

Vediamo cosa sono:

- La libreria KERNEL32.dll è una parte essenziale del sistema operativo Windows. Contiene funzionalità fondamentali per molte operazioni di basso livello come la gestione della memoria, la gestione dei file e la gestione dei processi.

- La libreria WININET.dll è un componente del sistema di Microsoft Windows che fornisce funzionalità per la comunicazione di rete attraverso Internet.

Punto 2) Identificazione delle sezioni

Sempre con il tool CFF explorer possiamo vedere le sezioni del malware:

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	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

Vediamo cosa sono:

- .text: contiene le istruzioni che la CPU eseguirà una volta che il programma sarà avviato.
- .rdata: contiene le informazioni sulle librerie e le funzioni importate ed esportate dal programma.
- .data: contiene i dati e le variabili globali del programma eseguibile, che devono essere disponibili da qualsiasi parte del programma.

Analisi del codice assembly

Punto 3) Identificazione dei costrutti

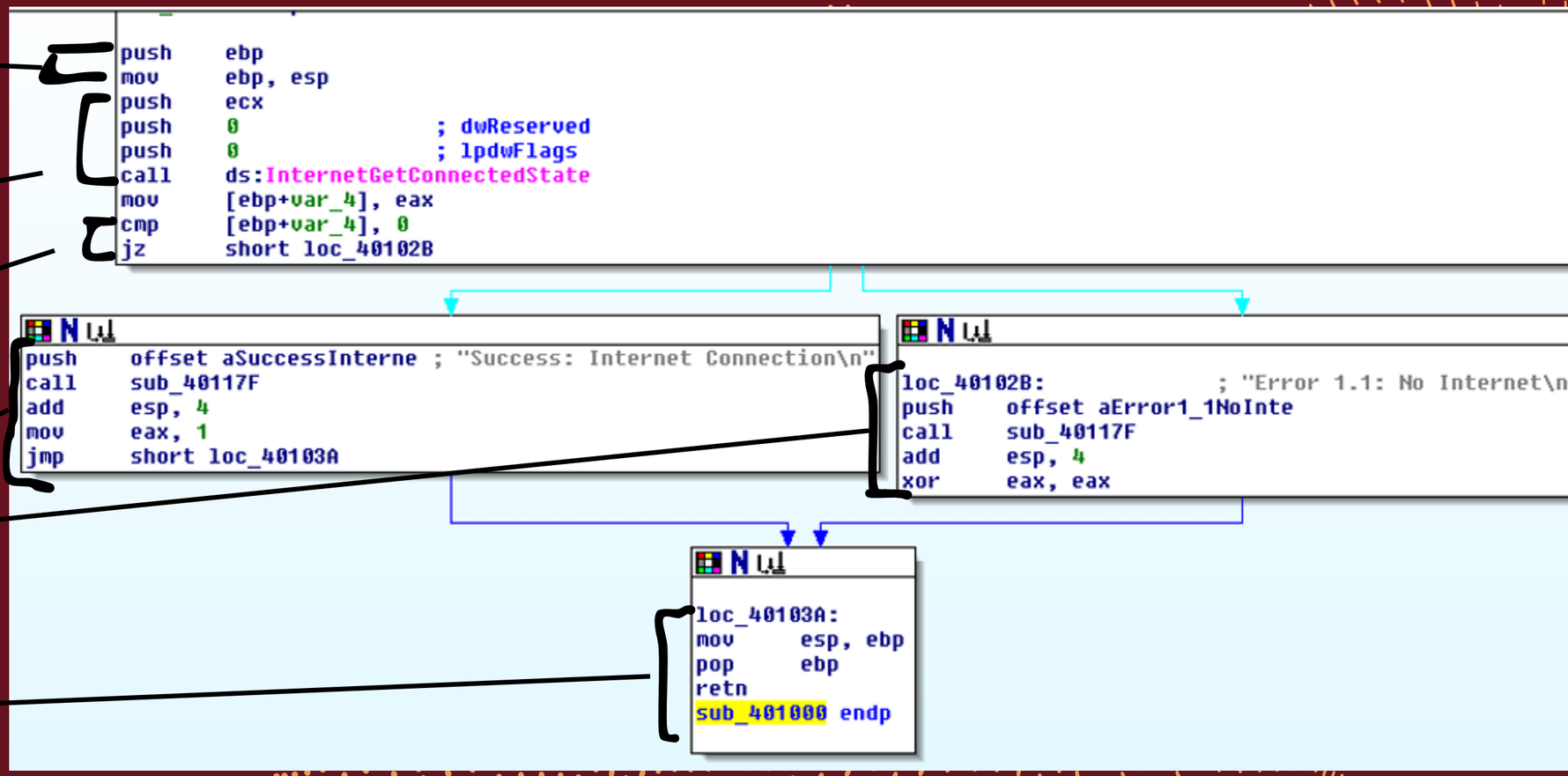
Creazione dello stack

Preparazione e chiamata della funzione

Creazione del ciclo if

le due opzioni del ciclo if

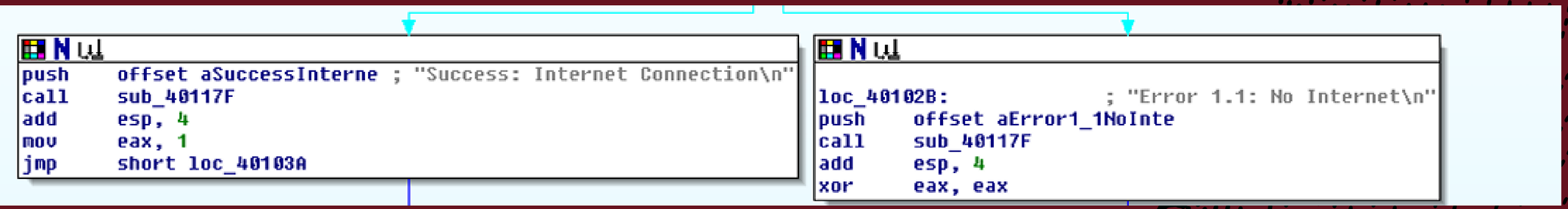
conclusione della funzione



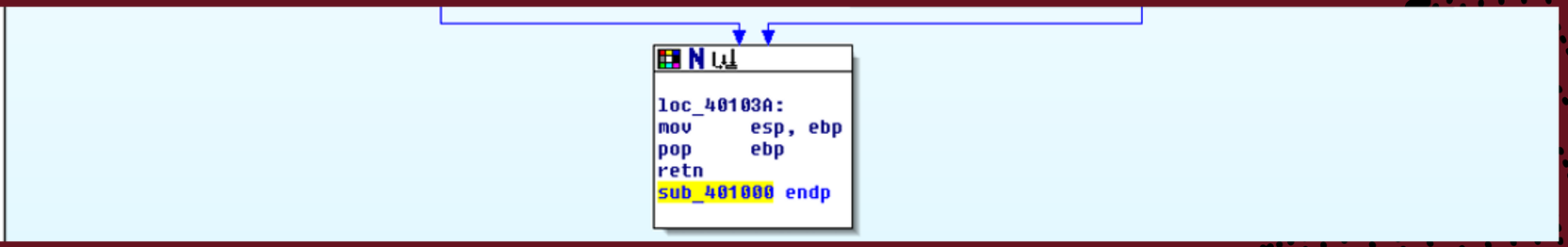
Punto 4) Cosa fa il codice

```
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
```

Questa prima parte del codice controlla se c'è una connessione a Internet chiamando la funzione InternetGetConnectedState e quindi confronta il risultato per determinare se la connessione è disponibile



Queste due parti gestiscono rispettivamente i casi di successo e di errore dopo il controllo dello stato della connessione a Internet nel codice precedente.



Questa ultima parte del codice termina il programma una volta eseguita una delle condizioni precedenti (di successo o di errore).