MALWARE ANALYSIS

COSTRUTTI C-ASSEMBLY X86

S10/L4

NELL'ESERCIZIO DI OGGI ANDIAMO AD ANALIZZARE IL CODICE ASSEMBLY DI UN MALWARE

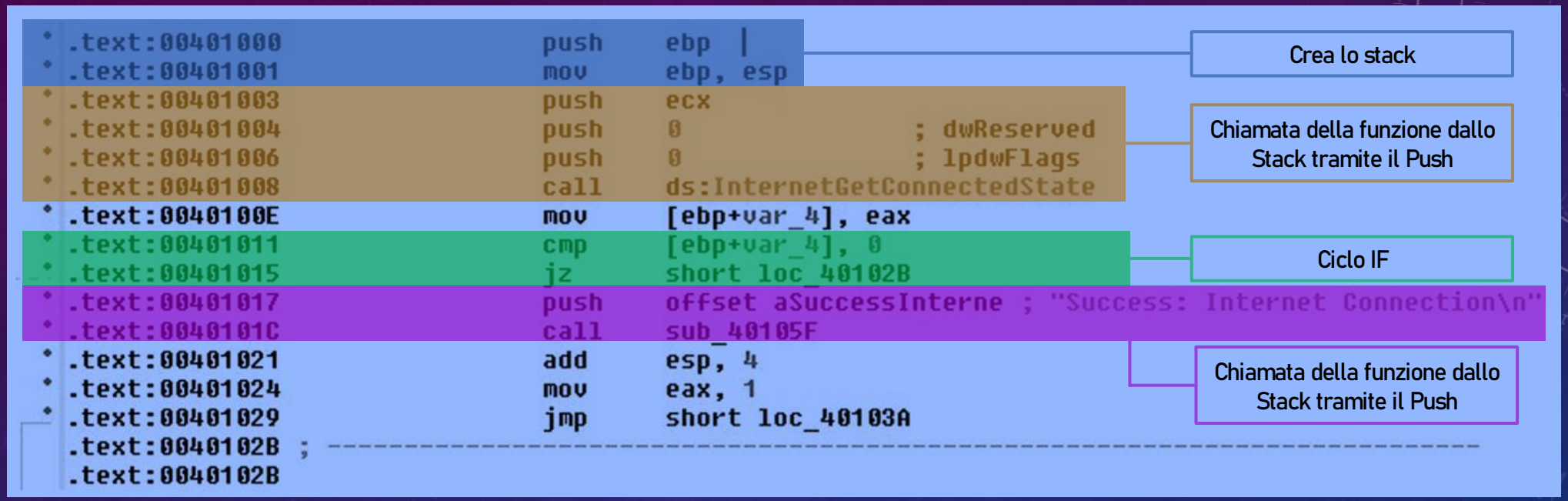
```
.text:00401000                push    ebp
.text:00401001                mov     ebp, esp
.text:00401003                push    ecx
.text:00401004                push    0               ; dwReserved
.text:00401006                push    0               ; lpdwFlags
.text:00401008                call    ds:InternetGetConnectedState
.text:0040100E                mov     [ebp+var_4], eax
.text:00401011                cmp     [ebp+var_4], 0
.text:00401015                jz      short loc_40102B
.text:00401017                push    offset aSuccessInterne ; "Success: Internet Connection\n"
.text:0040101C                call    sub_40105F
.text:00401021                add     esp, 4
.text:00401024                mov     eax, 1
.text:00401029                jmp     short loc_40103A
.text:0040102B ; ---------------------------------------------------------------------------
.text:0040102B
```

Sappiamo che la funzione ***«InternetGetConnectedState»*** prende in input 3 parametri e permette di controllare se una macchina ha accesso ad internet.

Cerchiamo di identificare lo scopo di ogni istruzione, facciamo una descrizione per ogni riga del nostro codice.

Sappiamo quindi che, questo codice controlla se c'è una connessione Internet attiva sulla macchina su cui viene eseguito, e restituisce 1 se c'è una connessione attiva e 0 altrimenti.

Analizzando il codice troviamo 4 costrutti con le rispettive funzionalità.



Dal codice del malware, si può dedurre che la sua funzionalità principale consiste nell'eseguire una chiamata di funzione a InternetGetConnectedState, che controlla lo stato della connessione Internet. Il codice verifica il valore di ritorno di questa funzione tramite un'istruzione condizionale. Se il valore di ritorno della funzione è diverso da zero, indica che la connessione è attiva. Dopo aver determinato che la connessione Internet è attiva, il malware inserisce nello stack di memoria una stringa costante contenente il messaggio "Success: Internet connection".e successivamente richiamato tramite una chiamata della funzione.

ISTRUZIONI	DESCRIZIONE
push ebp e mov ebp, esp	stanno impostando il frame del stack per questa funzione.
push ecx	mette il valore del registro ECX nello stack, che potrebbe essere usato come parametro per una chiamata di funzione successiva.
push 0 e push 0	stanno preparando due parametri nulli da passare a una funzione successiva.
call ds: InternetGetConnectedState	è una chiamata di funzione per controllare lo stato della connessione Internet. Il risultato della funzione è memorizzato nella variabile locale [ebp+var_4].
mov [ebp+var_4], eax:	Questa istruzione memorizza il valore di ritorno della funzione InternetGetConnectedState in una variabile locale allo stack.
cmp [ebp+var_4], 0	confronta il valore restituito dalla chiamata InternetGetConnectedState con 0.
jz Short loc_40102B	salta a loc_40102B se il valore restituito è zero, il che potrebbe significare che non c'è connessione Internet.
push offset aSuccessInterne e call Sub_40105F:	Se il controllo arriva al punto loc_40102B, è stampato un messaggio di successo indicando che c'è una connessione Internet attiva.
mov eax, 1 e jmp short loc_40103A:	Infine, il valore di ritorno della funzione corrente viene impostato su 1 (mov eax, 1) e il controllo salta a loc_40103A.