Identificare i costrutti noti in assembly

Nella lezione di oggi abbiamo visto alcuni dei principali costrutti di c in assembly.
Di seguito troviamo un codice da dove analizzare alla ricerca dei costrutti più noti.

```
.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
```

Possiamo subito individuare la **creazione** dello **stack** all'interno delle prime due righe, e la mancanza della terza riga classica per creare uno spazio per le variabili locali.

Subito dopo vediamo la creazione di "**STDCALL**" verso "**internetgetconnectedstate**", la quale permette di controllare se una macchina ha accesso ad internet, riconoscibile per la presenza di 3 variabili in "**PUSH**" e dalla mancanza della funzione per ripulire il suo **stack**.

Segue il costrutto **if** che, utilizzando una variabile globale, presumibilmente ottenuta da "internetgetconnectedstate", va a **compararla** con **0**. se il valore ottenuto è **diverso** da **0** allora crea un nuovo **push** e fa una nuova **chiamata**, presumibilmente una "**CDECL**" in quanto fa la **pulizia** con la riga dopo, imposta la variabile globale a 1 e fa un **jump** ad una zona che non vediamo del programma.
Altrimenti se il valore ottenuto dalla prima chiamata **è 0** salta il passaggio precedente e fa un **jump** in una sezione del programma oltre la nostra visuale.

Possiamo ipotizzare che questo sia un componente di un malware che **ricerca** una connessione ad **internet** e a seconda del risultato vada a svolgere operazioni differenti(come si può vedere dai jump) ma non possiamo sapere con precisione cosa solo da questo pezzo di codice