ANALISI CODICE ASSEMBLY

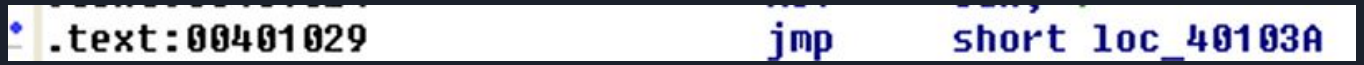
```
.text:00401000      push ebp
.text:00401001      mov ebp, esp
.text:00401003    push ecx
.text 00401004    push    0    ; dwReserved
.text:00401006    push    0    ; lpdwFlags
.text:00401008      call    ds:InternetGetConnectedState
.text:0040100E    mov    [ebp+var_4], eax
.text:00401011      cmp    [ebp+var_4], 0
.text:00401015    jz    short loc_40102B
.text:00401017    push    offset asuccessInterne ; ”Succes 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
```

- IDENTIFICAZIONE COSTRUTTI C :

IF

1. L'istruzione cmp [ebp+var_4], 0 confronta un valore (probabilmente memorizzato in una variabile chiamata var_4) con zero.
2. L'istruzione jz short loc_40102B è un salto condizionale. Se il confronto precedente ha dato come risultato che la variabile è uguale a zero, allora il programma salta a una certa parte del codice (l'etichetta loc_40102B

GOTO



Quando l'istruzione viene eseguita, il flusso del programma salta immediatamente all'indirizzo specificato dall'etichetta loc_40103A. Questo è esattamente come l'uso di goto in C, dove ti consente di passare direttamente ad una parte del codice e continuare da lì.

- Ipotesi Funzionamento

Il codice sembra avere il compito di verificare lo stato della connessione a Internet, utilizzando “InternetGetConnectedState’ e in caso positivo stampa un messaggio di successo.

- Spiegazione Codice

.text:00401000 push ebp: Mette il valore attuale del registro EBP nello stack.

.text:00401001 mov ebp, esp: Copia il valore dello stack di ESP in EBP.

.text:00401003 push ecx: Mette il valore di ECX nello stack.

.text:00401004 push 0 ; dwReserved: Mette il valore 0 nello stack. E lo passiamo come argomento dwReserved per la funzione.

.text:00401006 push 0 ; lpdwFlags: Mette il valore 0 nello stack. E lo passiamo come argomento lpdwFlags alla funzione.

.text:00401008 call ds:InternetGetConnectedState: Chiama la funzione InternetGetConnectedState che controlla lo stato della connessione a Internet.

.text:0040100E mov [ebp+var_4], eax: Memorizza il valore restituito dalla funzione InternetGetConnectedState in [ebp+var_4].

.text:00401011 cmp [ebp+var_4], 0: Confronta il valore memorizzato con 0.

.text:00401015 jz short loc_40102B: Passa ad loc_40102B se il valore nella variabile è zero. Quindi se non c'è connessione a Internet, il programma salta a loc_40102B.

.text:00401017 push offset asuccessInterne: Mette l'indirizzo della stringa asuccessInterne nello stack. Che sembra essere un messeggio che indica la corretta connesione a internet.

.text:0040101C call sub_40105F: Chiama una subroutine a sub_40105F, per stampare il messaggio di successo sulla connessione.

.text:00401021 add esp, 4: Ripristina lo stack dopo la chiamata alla funzione.

.text:00401024 mov eax, 1: Imposta il registro EAX a 1.

.text:00401029 jmp short loc_40103A: Passa a loc_40103A. Questa azione viene eseguita a prescindere se e presente una connesione a internet.

.text:0040102B ;---: e un commento.