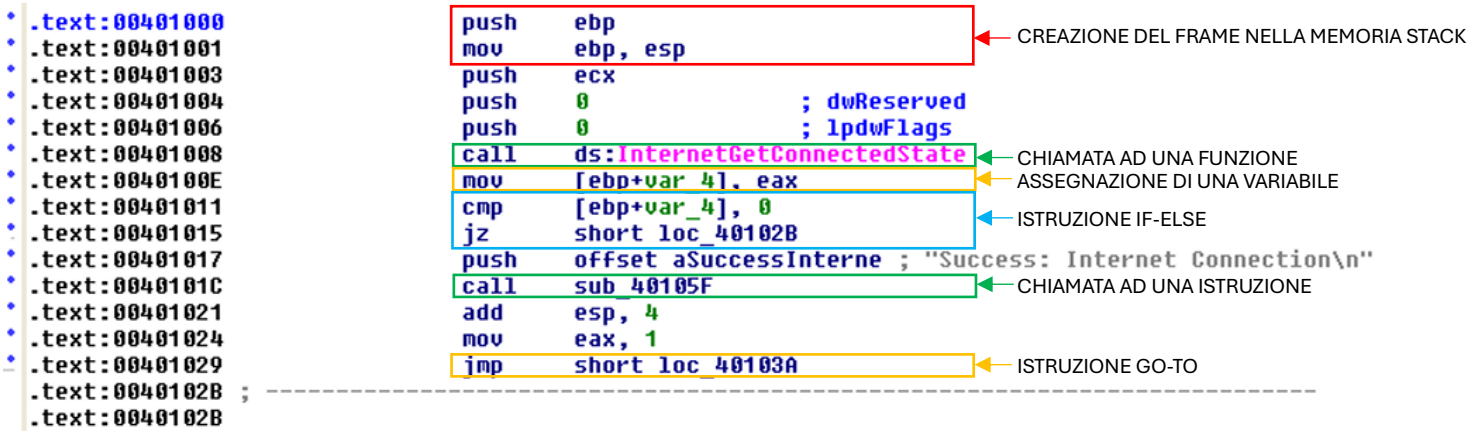
CONSEGNA SETTIMANA 10 LEZIONE 4

Costrutti C in assembly x86

Il progetto di oggi prevede saper descrivere alcuni costrutti C espressi in linguaggio assembly. Nel nostro caso abbiamo un frammento di codice di un malware. Ci viene chiesto di identificare i costrutti noti e descrivere il comportamento del malware.



Analisi dettagliata del codice:

- **push ebp** *//salva lo stato del registro inserendolo nello stack*
- **mov ebp, esp** *//copia il valore di esp nel registro di base ebp*

Questa sezione del codice viene utilizzata per **creare un frame di stack** per il programma in uso, viene usato il registro ebp per accedere alle variabili locali tramite un offset rispetto a ebp.

- **push ecx** *//salva lo stato del registro ecx nello stack*
- **push 0 :dwReserved** *//salva il valore 0 nello stack in riferimento alla flag dwReserved*
- **push 0 :lpdwFlags** *//salva il valore 0 nello stack in riferimento alla flag lpdwFlags*

In questa parte del codice vengono inizializzati gli argomenti per la funzione che verrà chiamata in seguito (InternetGetConnectedState).

- **call ds:InternetGetConnectedState** *//chiama la funzione InternetGetConnectedState*
- **mov [ebp+var_4], eax** *//salva il valore di ritorno della funzione nella variabile locale var_4*

Viene chiamata la funzione InternetGetConnectedState e viene salvato il valore di ritorno nella variabile ebp+var_4.

- **cmp [ebp+var_4], 0** *// confronta il valore contenuto in var_4 con 0*
- **jz short loc_40102B** *//salta alla posizione 40102B se il risultato del confronto precedente è 0 (ovvero se non c'è connessione internet)*
- **push offset aSuccessInterne** *//salva nello stack la stringa "success internet connection"*
- **call sub_40105F** *//chiama la funzione alla posizione 40105F*

Viene eseguito un confronto tra il valore di ebp+var_4 e 0. Se il risultato del confronto è uguale a 0, il codice passerà direttamente alla location 40102B. Diversamente, se sarà diverso da 0 viene eseguita un'operazione di stampa, per confermare l'avvenuta connessione a internet.

- **add esp, 4** *//aggiorna lo stack pointer per poter poi rimuovere i valori dallo stack*
- **mov eax, 1** *//salva il valore 1 nel registro eax*
- **jmp short loc_40103A** *//salta alla posizione 40103A*

In sintesi, qual'è il funzionamento di questo malware?

Questo frammento di codice pare essere usato per verificare lo stato della connessione a Internet e agisce di conseguenza in base al risultato. Se la connessione è presente viene stampato un messaggio di conferma, altrimenti è possibile che vengano eseguite altre istruzioni oppure terminato il programma, ma avendo solamente questa porzione di codice non ci è dato saperlo.