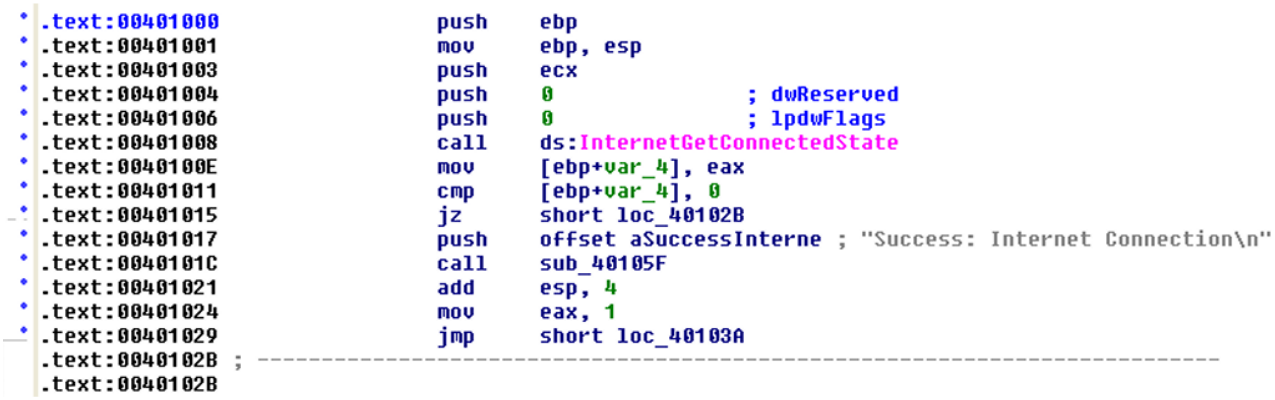
Esercizio linguaggio assembly

La figura seguente mostra un estratto del codice di un malware. Identificare i costrutti noti visti durante la lezione teorica. Provate ad ipotizzare che funzionalità è implementata nel codice assembly.

1. Identificare i costrutti noti (e s. while, for, if, switch, ecc.)
2. Ipotizzare la funzionalità – esecuzione ad alto livello
3. BONUS: studiare e spiegare ogni singola riga di codice



Analisi dei costrutti

.text:00401000 push ebp

.text:00401001 mov ebp, esp

.text:00401003 push ecx

Questi comandi preparano lo stack frame per la funzione.

.text:00401004 push 0 ; dwReserved

.text:00401006 push 0 ; lpdwFlags

.text:00401008 call ds:InternetGetConnectedState

Vengono passati due parametri a InternetGetConnectedState: dwReserved e lpdwFlags, entrambi impostati a 0. Questa funzione verifica lo stato della connessione a Internet.

.text:0040100E mov [ebp+var_4], eax

Il risultato della chiamata a InternetGetConnectedState viene memorizzato nella variabile locale var_4.

.text:00401011 cmp [ebp+var_4], 0

.text:00401015 jz short loc_40102B

Se il risultato è zero (nessuna connessione a Internet), il flusso del programma salta a 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

Se il risultato non è zero (connessione a Internet presente), viene mostrato un messaggio di successo e la funzione restituisce 1.

Ecco una spiegazione riga per riga del codice assembly fornito:

.text:00401000 push ebp

Salva il valore del registro ebp (base pointer) sullo stack. Questo viene fatto per preservare il contesto della funzione chiamante.

.text:00401001 mov ebp, esp

Copia il valore del registro esp (stack pointer) nel registro ebp. Questo imposta ebp come nuovo puntatore di base per il frame dello stack corrente.

.text:00401003 push ecx

Salva il valore del registro ecx sullo stack. Questo viene fatto per preservare il valore di ecx che potrebbe essere modificato all'interno della funzione.

.text:00401004 push 0 ; dwReserved

Spinge il valore 0 sullo stack. Questo sarà il primo parametro (dwReserved) passato alla funzione InternetGetConnectedState.

.text:00401006 push 0 ; lpdwFlags

Spinge il valore 0 sullo stack. Questo sarà il secondo parametro (lpdwFlags) passato alla funzione InternetGetConnectedState.

.text:00401008 call ds:InternetGetConnectedState

Chiama la funzione InternetGetConnectedState che controlla lo stato della connessione a Internet. Il risultato della funzione viene restituito nel registro eax.

.text:0040100E mov [ebp+var_4], eax

Memorizza il valore del registro eax (che contiene il risultato della chiamata a InternetGetConnectedState) nella variabile locale var_4. Questa variabile locale si trova all'indirizzo ebp-4.

.text:00401011 cmp [ebp+var_4], 0

Confronta il valore della variabile locale var_4 con 0. Questo viene fatto per verificare se c'è una connessione a Internet. Se InternetGetConnectedState restituisce 0, significa che non c'è connessione.

.text:00401015 jz short loc_40102B

Se il confronto precedente risulta in zero (cioè, non c'è connessione a Internet), salta all'etichetta loc_40102B. Questo indica una condizione di errore o mancanza di connessione.

.text:00401017 push offset aSuccessInterne ; "Success: Internet Connection"

Spinge l'indirizzo della stringa "Success: Internet Connection\n" sullo stack. Questo è l'argomento per la funzione di stampa sub_40105F.

.text:0040101C call sub_40105F

Chiama la funzione sub_40105F, che probabilmente è una funzione per stampare o loggare il messaggio di successo.

.text:00401021 add esp, 4

Aggiunge 4 a esp per pulire lo stack dall'argomento passato alla funzione sub_40105F.

.text:00401024 mov eax, 1

Imposta il registro eax a 1. Questo valore indica successo e sarà il valore di ritorno della funzione.

.text:00401029 jmp   short loc_40103A

Salta all'etichetta loc_40103A. Questo evita di eseguire il codice all'etichetta loc_40102B e continua con il flusso normale della funzione.

.text:0040102B ;...

Etichetta loc_40102B. Questa è la destinazione del salto condizionale (jz) che gestisce il caso di errore o mancanza di connessione.

In breve

1. Il codice prepara lo stack frame e preserva i registri.
2. Chiama la funzione InternetGetConnectedState per verificare la connessione a Internet.
3. Se c'è una connessione, stampa un messaggio di successo e ritorna 1.
4. Se non c'è connessione, salta a una routine di gestione dell'errore (non visibile nell'estratto)

Conclusioni

Questo estratto di codice assembly verifica se c'è una connessione a Internet utilizzando la funzione InternetGetConnectedState. Se una connessione è presente, viene stampato un messaggio di successo ("Success: Internet Connection\n") e la funzione restituisce 1. Se non c'è una connessione, il flusso del programma salta ad un'altra parte del codice dove probabilmente viene gestito l'errore o la mancanza di connessione.

In sintesi, la funzionalità principale di questo codice è quella di controllare la presenza di una connessione a Internet e agire di conseguenza.