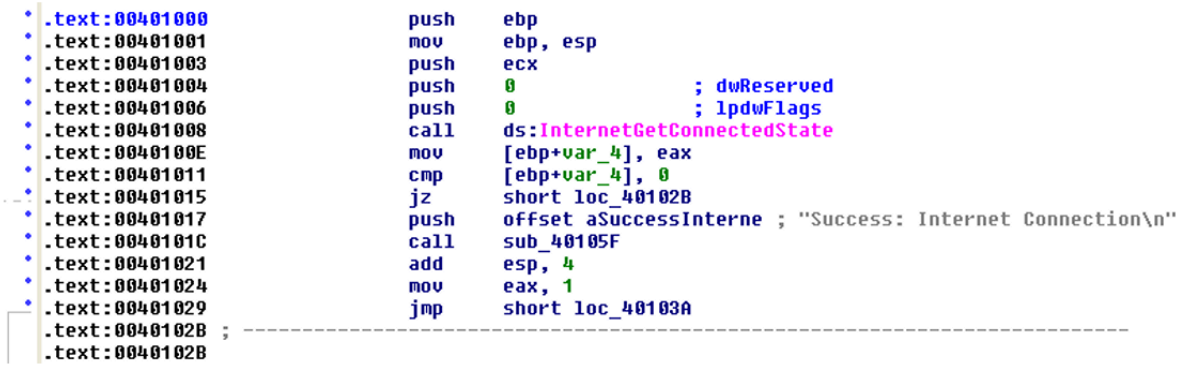
S10/L4

Traccia: 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.
Hint: La funzione internetgetconnectedstate prende in input 3 parametri e permette di controllare se una macchina ha accesso ad internet.
Consegna:
1. Identificare i costrutti noti (es. while, for, if, switch, ecc.)
2. Ipotizzare la funzionalità –esecuzione ad alto livello
3. BONUS: studiare e spiegare ogni singola riga di codice



1.Identificare i costrutti noti

mov [ebp+var_4], eax

cmp [ebp+var_4], 0

jz short loc_40102B

Questi tre comandi assomigliano a un costrutto "if" in C. Se il valore in [ebp+var_4] è uguale a zero, il programma salterà a loc_40102B (etichetta specifica.); altrimenti se il valore di var_4 è diverso da zero, procede con il codice e stamperà il messaggio di “success internet”. Per cui se il confronto precedente non ha dato esito "zero", il programma non è saltato al blocco di codice corrispondente all'istruzione jz, quindi continua eseguendo il codice che segue direttamente il confronto.

2.Ipotizzare la sua funzionalità/esecuzione ad alto livello

Questo codice assembly sembra essere parte di un programma che verifica lo stato della connessione Internet e stampa un messaggio di successo se la connessione è attiva. Inizia impostando il frame pointer (ebp) e lo stack pointer (esp) per gestire le variabili locali e i parametri della funzione.

Chiama la funzione InternetGetConnectedState per verificare lo stato della connessione Internet. Salva il valore restituito dalla funzione nella variabile locale var_4. Controlla se il valore di var_4 è zero (che potrebbe indicare l'assenza di connessione) e, se lo è, salta a un'etichetta specifica (loc_40102B). Se il valore di var_4 non è zero, il programma procede a stampare un messaggio di successo relativo alla connessione Internet. Termina restituendo il valore 1.

Bonus:

push ebp: questa istruzione spinge il valore del registro ebp nello stack. Il registro ebp è comunemente usato come frame pointer per puntare alla base dello stack frame corrente.

mov ebp, esp: sposta il valore dello stack pointer (esp) nel registro base del frame (ebp). Questo solitamente imposta ebp come un punto di riferimento per accedere alle variabili locali e ai parametri della funzione.

push ecx: spinge il valore del registro ecx nello stack. Il registro ecx viene spesso utilizzato come registro di conteggio o per altre operazioni.

push 0 ;dwReserved: Questa istruzione inserisce il valore 0 nello stack. Il commento ;dwReserved indica che questo valore potrebbe essere utilizzato come parametro denominato dwReserved in una funzione

push 0 ;lpdwFlags: Questa istruzione inserisce anche il valore 0 nello stack, con il commento ;lpdwFlags. Anche qui, lo 0 potrebbe rappresentare un parametro denominato lpdwFlags in una chiamata di funzione.

In sintesi, queste istruzioni push preparano lo stack per una possibile chiamata di funzione o di sistema successiva, fornendo valori iniziali per i parametri dwReserved e lpdwFlags.

call ds:InternetGetConnectedState: chiama la funzione InternetGetConnectedState indicata dall'indirizzo ds. Questa funzione è utilizzata per verificare lo stato della connessione Internet.

mov [ebp+var_4], eax: Questa istruzione memorizza il valore restituito dalla chiamata alla funzione InternetGetConnectedState nella variabile locale var_4, che si trova nel frame dello stack corrente.

cmp [ebp+var_4], 0: confronta il valore memorizzato nella variabile var_4 con 0.

jz short loc_40102B: salta all'etichetta loc_40102B se il confronto precedente ha dato esito "zero" (ossia, se il valore memorizzato in var_4 è uguale a zero).

push offset aSuccessInterne: si mette l'indirizzo dell'etichetta a “SuccessInterne” nello stack. L'etichetta sembra indicare un messaggio di successo relativo alla connessione Internet.

call sub_40105F: Questa istruzione chiama la subroutine indicata da sub_40105F. Probabilmente questa subroutine è responsabile di stampare il messaggio di successo.

add esp, 4: si aggiunge 4 al registro dello stack pointer (esp). Questo serve a liberare lo spazio dello stack occupato dai parametri passati alla funzione chiamata.

mov eax, 1: carica il valore 1 nel registro eax. Probabilmente è utilizzato come valore di ritorno della funzione corrente.

jmp short loc_40103A: salta all'etichetta loc_40103A. Presumibilmente, questo è il punto dopo la gestione del successo della connessione Internet.