EPICODE

Esercizio S10 – L4

TRACCIA

La figura seguente mostra un estratto del codice di un malware. Identificare i costrutti noti visti durante la lezione teorica.

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

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

Segue soluzione

SOLUZIONE

1. Identificare i costrutti noti (es. while, for, if, switch, creazione stack ecc.)  
  
Si identificano almeno due costrutti noti:

1. Creazione di uno stack: In questa situazione lo stack viene creato con “push ecx”. Serve a creare spazio per una variabile locale senza preoccuparsi del valore di ecx. Per esempio, il compilatore Visual C++ usa spesso “push ecx” e “pop ecx” come alternative a “sub esp, 4” e “add esp, 4”.

2. Si tratta di un “if”, in quanto in base al risultato della funzione salterà, o meno, ad una locazione diversa.

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente

2. Ipotizzare la funzionalità – esecuzione ad alto livello

Il malware in questione testa la connessione ad Internet e richiama poi una funzione che, probabilmente, lo connette ad un server remoto per ricevere altre istruzioni.

3. BONUS: studiare e spiegare ogni singola riga di codice

push ebp:  
Salva il valore di ebp nella pila. ebp è il registro che indica il puntatore di base della funzione.

mov ebp, esp:  
Copia il valore di esp in ebp. esp è il registro che indica il puntatore di pila, cioè la posizione “in alto” in cui inserire o estrarre i dati dalla pila.

push ecx :  
Salva il valore di ecx nella pila e crea spazio per una variabile locale.

push 0 ; dwReserved:  
Passa 0 come secondo parametro alla funzione InternetGetConnectedState. Questo parametro è riservato e deve essere sempre 0.

push 0 ; lpdwFlags:  
Passa 0 come primo parametro alla funzione InternetGetConnectedState. Questo parametro è un puntatore a una variabile che riceve il tipo di connessione a Internet.

call ds:InternetGetConnectedState:  
Chiama la funzione InternetGetConnectedState, che verifica lo stato della connessione a Internet del sistema locale e restituisce un valore booleano (vero o falso) in eax. eax è il registro che viene usato spesso per memorizzare il valore di ritorno di una funzione.

mov [ebp+var\_4], eax:  
Salva il valore restituito dalla funzione (vero o falso) in una variabile locale.

cmp [ebp+var\_4], 0:  
Confronta il valore della variabile locale con 0.

jz short loc\_40102B:  
Salta a loc\_40102B se il valore della variabile locale è 0, cioè se non c'è connessione a Internet. Questo salto dipende dal valore di ZF (zero flag), che viene impostato a 1 se il confronto precedente ha dato come risultato 0 (in quanto 0 = 0).

push offset asuccessinterne : "Success: Internet Connection\n":  
Passa l'indirizzo di una stringa "Success: Internet Connection\n" come parametro alla funzione sub\_40105F. Questa stringa potrebbe essere usata per comunicare con un server remoto.

call sub\_40105F:  
Chiama la funzione sub\_40105F, che è una funzione definita dall'utente e non fa parte delle funzioni standard di assembly. Per sapere cosa fa, bisognerebbe analizzare il codice della funzione. Tuttavia, è probabile che la funzione serva a comunicare con un server remoto.

add esp, 4:  
Ripristina il puntatore di pila dopo aver passato il parametro, sommando 4 byte al valore di esp. Questo serve per mantenere la coerenza della pila e per evitare di sovrascrivere i dati salvati in precedenza.

mov eax, 1:  
Assegna 1 al registro eax come valore di ritorno della funzione corrente (connessione stabilita).

jmp short loc\_40103A:  
Salta incondizionatamente a loc\_40103A, che potrebbe essere la fine della funzione corrente o il punto in cui il malware esegue altre operazioni malevole.