S10-L4 Emanuele Di Stefano

Report sull'Analisi del Codice Assembly del Malware

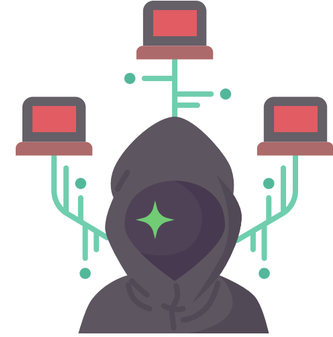
1. Identificazione dei Costrutti Noti

Nel codice assembly fornito, possiamo identificare i seguenti costrutti noti

- Chiamata di funzione (call): Utilizzato per chiamare funzioni specifiche, come InternetGetConnectedState e sub_40105F.
- Confronto (cmp): Utilizzato per confrontare due valori.
- Istruzione condizionale (jz): Salta a un'etichetta se il confronto precedente ha trovato che i valori sono uguali (zero flag è impostato).
- Istruzione di salto (jmp): Salta incondizionatamente a un'altra istruzione.
- Movimentazione dati (mov): Trasferisce dati tra registri o tra memoria e registri.
- Aritmetica (add): Esegue operazioni aritmetiche, in questo caso aggiunge 1 a un registro.
- Spinta e recupero dallo stack (push e pop): Usato per gestire lo stack durante le chiamate di funzione.

2. Ipotesi della Funzionalità

Il codice assembly sembra controllare se c'è una connessione Internet attiva utilizzando la funzione InternetGetConnectedState. Se la connessione è presente, il programma stampa un messaggio di successo "Success: Internet Connection". Dopo di che, chiama un'altra funzione sub_40105F, che probabilmente esegue ulteriori operazioni se la connessione è attiva.



3. Studio e Spiegazione di Ogni Singola Riga di Codice

```
-.text:00401000    push    ebp                 ; Salva il vecchio valore di base pointer
-.text:00401001    mov     ebp, esp            ; Imposta il base pointer per il nuovo stack frame
-.text:00401003    push    ecx                 ; Salva il registro ecx sullo stack
-.text:00401004    push    0                   ; Passa 0 come parametro dwReserved alla funzione
-.text:00401006    push    0                   ; Passa 0 come parametro lpdwFlags alla funzione
-.text:00401008    call    ds:InternetGetConnectedState ; Chiama la funzione per controllare lo stato della connessione Internet
-.text:0040100E    mov     [ebp+var_4], eax    ; Salva il risultato della chiamata di funzione in [ebp+var_4]
-.text:00401011    cmp     [ebp+var_4], 0      ; Confronta il risultato con 0
-.text:00401015    jz      short loc_40102A    ; Salta a loc_40102A se il risultato è 0 (no connessione)
-.text:00401017    push    offset aSuccessInterne  ; "Success: Internet Connection\n" ; Carica l'indirizzo della stringa di successo sullo stack
-.text:0040101C    call    sub_40105F          ; Chiama una funzione per stampare il messaggio
-.text:00401021    add     esp, 4              ; Ripristina lo stack pointer
-.text:00401024    add     eax, 1              ; Incrementa eax di 1
-.text:00401027    jmp     short loc_401030    ; Salta a loc_401030
-.text:00401029 loc_40102A:                    ; Etichetta per la condizione senza connessione
-.text:0040102A    ; Codice per gestire la condizione di no connessione (non mostrato nel frammento)
-.text:0040102B    ; Altre istruzioni (non mostrate nel frammento)
```

Il codice fornito è un frammento di un malware che verifica se una macchina ha una connessione Internet attiva. Utilizza la funzione InternetGetConnectedState per questo scopo.
Se viene rilevata una connessione, il malware stampa un messaggio di successo e chiama una funzione per eseguire ulteriori azioni. Se non c'è connessione, esegue altre operazioni (non mostrate nel frammento di codice).

BONUS: Dettagli Aggiuntivi

-La funzione InternetGetConnectedState è parte della libreria WinINet e viene utilizzata per determinare lo stato della connessione Internet. I parametri passati (due zeri) indicano che non ci sono flag specifici né riserve.

-sub_40105F è una funzione non mostrata nel frammento di codice, ma probabilmente esegue azioni legate alla gestione della connessione Internet.