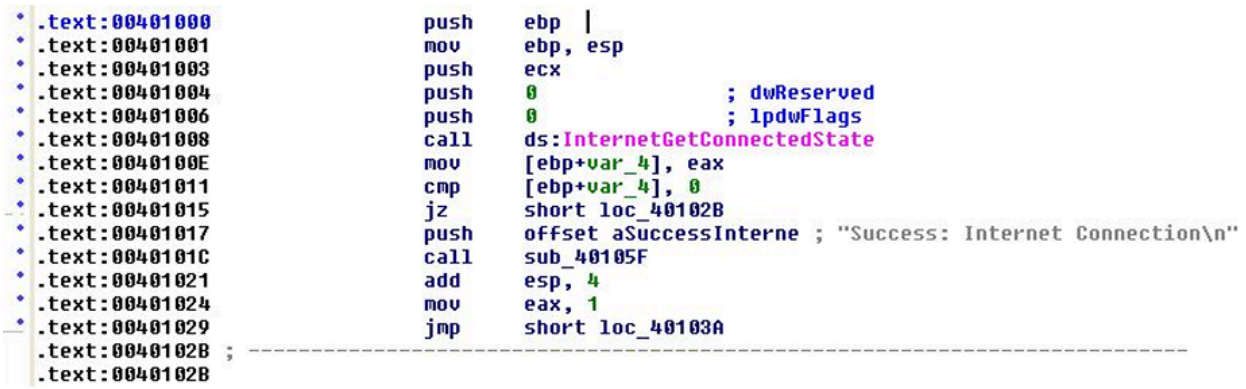
Esercitazione WEEK 22 D4

Costrutti C - Assembly x86

Ettore Farris

Descrizione sintetica

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.

Analisi costrutti

Il codice assembly dell'esempio è un frammento di un programma che verifica la connessione ad internet di un macchina.

1. *Creazione dello stack frame*

push ebp
mov ebp, esp

Con queste due istruzioni viene creato lo stack frame necessario per salvare in memoria le variabili locali necessarie per chiamare la funzione *InternetGetConnectedState.* Quando si esegue l'istruzione *push ebp*, il valore corrente del registro *ebp* (base dello stack) viene salvato nello stack , precisamente all'indirizzo puntato dal registro *esp* (cima dello stack) della funzione chiamante. Quindi la funzione chiamata corrisponderà alla cima della funzione chiamante.

2. *Chiamata della funzione InternetGetConnectedState*

push ecx
push 0 ;lpdwFlags
push 0 ;dwReserved
call ds:InternetGetConnectedState

Con questi comandi viene caricato il valore del registro *ecx* e i valori delle variabili locali (che probabilmente verranno assegnati a *lpdwFlags* e *dwReserved*) che serviranno alla funzione chiamata *InternetGetConnectedState.* La funzione infine viene chiamata con l'istruzione *call.* Questa funzione è utilizzata per determinare lo stato della connessione di rete dell'utente.

3. *Costrutto if*

mov [ebp+var_4], eax
cmp [ebp+var_4], 0
jz short loc_40102B
push offset aSuccessInternet ;"Success Internet Connection.\n"
call sub_40105F

Questa serie di comandi include un costrutto if. Il contenuto del registro *eax* viene passato alla variabile *var_4.* Successivamente il valore di questa variabile viene confrontato con 0 e, se i due valori sono uguali (Z Flag = 0), verrà eseguita un'istruzione presente nella posizione *loc_40102B* (situato fuori dal nostro codice).

Se i due valori non coincidono viene probabilmente stampato il messaggio di successo *"Success Internet Connection.\n"* tramite la call di funzione *sub_40105F* (probabilmente responsabile della print). Il codice potrebbe essere:

```
if(var_4==0){

        printf("Success Internet Connection.\n");
} else {
        //fai qualcos'altro
}
```

4. *Pulizia stack e jump non condizionale*

add esp, 4
mov eax, 1
jmp short loc_40103A

Le istruzioni dopo la call servono con tutta probabilità a restituire il valore di ritorno della funzione e a ripulire lo stack. Dopo questa esecuzione, viene effettuato un jump non condizionale alla posizione di memoria *loc_40103A.*