




UNITÀ 3 SETTIMANA 2

COMPITO 4


IDENTIFICAZIONE DEI COSTRUTTI NOTI

```
* .text:00401000      push    ebp
* .text:00401001      mov     ebp, esp
* .text:00401003      push    ecx
* .text:00401004      push    0             ; dwReserved
* .text:00401006      push    0             ; lpdwFlags
* .text:00401008      call   ds:InternetGetConnectedState
* .text:0040100E      mov     [ebp+var_4], eax
* .text:00401011      cmp     [ebp+var_4], 0
* .text:00401015      jz      short 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
* .text:0040102B ; -----
* .text:0040102B
```



Ecco i costrutti noti che possono essere identificati:

- 1-push:** Mette il valore specificato sullo stack.
- 2-mov:** Muove/copiare un valore da una posizione a un'altra.
- 3-call:** Chiama una funzione o un'etichetta.
- 4-cmp:** Confronta due valori.
- 5-jz:** Salta all'etichetta specificata se il confronto precedente ha dato esito "zero".
- 6-push offset:** Mette l'indirizzo di un'etichetta o di una stringa sullo stack.
- 7-add:** Aggiunge due operandi.
- 8-jmp:** Salta all'etichetta specificata.



Ecco i costrutti noti i costrutti noti che ho trovato sono
2 if

```
♦ .text:00401011      cmp     [ebp+var_4], 0  
♦ .text:00401015      jz      short loc_40102B
```




Ipotizzare la funzionalità –esecuzione ad alto livello

Dall'analisi del codice assembly fornito, possiamo ipotizzare la funzionalità generale del malware a un livello più alto:

Verifica della connessione Internet: Il malware inizia verificando lo stato della connessione Internet utilizzando la funzione `InternetGetConnectedState`. Questa operazione potrebbe essere utile per determinare se la vittima è connessa a Internet e quindi se il malware può eseguire azioni che coinvolgono la rete.

Messaggio di notifica: Se la connessione Internet è attiva, il malware stampa un messaggio di successo utilizzando la subroutine `sub_40105F`. Questo potrebbe essere un tentativo di ingannare l'utente o di fornire una falsa sensazione di sicurezza.

Ritorno di un valore: Dopo aver verificato lo stato della connessione Internet e stampato il messaggio di notifica, il malware imposta il registro EAX a 1 prima di terminare. Questo potrebbe essere un valore di ritorno utilizzato per indicare lo stato di successo al chiamante del malware.



spiegazione dettagliata di ogni riga di codice del codice assembly:

1-Push ebp: Questa istruzione spinge il valore attuale del registro di base (ebp) nello stack. Questo viene fatto per salvare il valore corrente di ebp e preparare lo stack per l'uso di ebp come frame pointer.

2-mov ebp, esp: Questa istruzione muove il valore corrente dello stack pointer (esp) nel registro di base (ebp). Questo stabilisce un frame pointer per il frame della funzione corrente, facilitando l'accesso alle variabili locali e ai parametri tramite offset rispetto a ebp.


3-push ecx: Questa istruzione spinge il contenuto del registro ecx nello stack. Il registro ecx potrebbe essere utilizzato come registro di contatore o per altri scopi temporanei.

4-push 0: Questa istruzione spinge il valore zero nello stack. Questo potrebbe essere un parametro per una funzione successiva o un valore iniziale per una variabile locale.

5-push 0: Questa istruzione spinge nuovamente il valore zero nello stack. Potrebbe essere utilizzato come secondo parametro per una funzione successiva o per altri scopi.

6-call ds:InternetGetConnectedState: Questa istruzione chiama la funzione InternetGetConnectedState del segmento di dati (ds). Questa funzione controlla lo stato della connessione Internet e il risultato viene restituito in eax.

7-mov [ebp+var_4], eax: Questa istruzione memorizza il valore restituito dalla funzione InternetGetConnectedState nel registro eax nella variabile locale [ebp+var_4].



8-cmp [ebp+var_4], 0: Questa istruzione confronta il valore memorizzato nella variabile locale [ebp+var_4] con zero.

9-jz short loc_40102B: Questa istruzione salta a loc_40102B se il risultato del confronto precedente è zero. In altre parole, se il risultato restituito dalla funzione InternetGetConnectedState indica che non c'è connessione Internet, il programma salta a loc_40102B.

10-push offset aSuccessInterne: Questa istruzione spinge l'indirizzo del messaggio "Success: Internet Connection\n" nello stack. Questo indirizzo verrà utilizzato come parametro per la funzione di stampa successiva.

11-call sub_40105F: Questa istruzione chiama una subroutine, identificata come sub_40105F. Questa subroutine è probabilmente responsabile della stampa del messaggio sullo schermo.

12-add esp, 4: Questa istruzione aggiunge 4 byte allo stack pointer (esp), deallocando lo spazio dello stack utilizzato per il parametro passato alla funzione di stampa.

13-mov eax, 1: Questa istruzione assegna il valore 1 al registro eax. Potrebbe essere un codice di ritorno che indica il successo dell'operazione.

14-jmp short loc_40103A: Questa istruzione effettua un salto incondizionato a loc_40103A, ignorando le istruzioni successive. Potrebbe essere utilizzato come parte di un flusso di controllo condizionale.