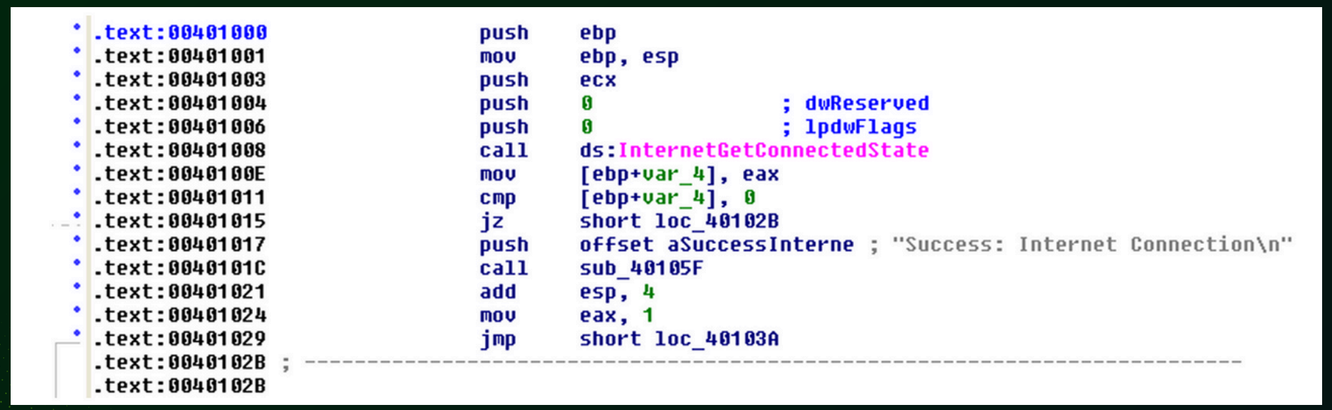
COSTRUTTI C - ASSEMBLY X86

MALWARE ANALYSIS- S10L4

COSA FARE



```
.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
```

La figura seguente mostra un estratto del codice di un malware. Identificare i costrutti noti Esercizio Linguaggio Assembly visti durante la lezione teorica. Provate ad ipotizzare che funzionalità è implementata nel codice assembly.

Consegna:

1. Id e notificare i costrutti noti (e s. while, for, if, switch, ecc.)
2. Ipotizzare la funzionalità – esecuzione ad alto livello
3. BONUS: studiare e spiegare ogni singola riga di codice

IDENTIFICAZIONE COSTRUTTI

IF-ELSE

```
.text:0040100E mov     [ebp+var_4], eax
.text:00401011 cmp     [ebp+var_4], 0
.text:00401015 jz      short loc_40102B
```

Queste istruzioni implementano un controllo condizionale. Il valore restituito dalla funzione (‘eax’) viene confrontato con 0. Se il valore è 0, si verifica un salto condizionale (‘jz’) all'etichetta ‘loc_40102B’, che gestisce il caso di fallimento.

FINALITÀ DELL’ESECUZIONE

La funzione assembly fornita verifica se il computer è connesso a Internet utilizzando la funzione **InternetGetConnectedState**.

Se è connesso:

1. Stampa un messaggio di successo ("Success: Internet Connection\n").
2. Imposta il valore di ritorno a 1.

Se non è connesso:

1. Salta alla sezione di gestione del fallimento (il codice non è mostrato nell'immagine).

CODICE AD ALTO LIVELLO: LINGUAGGIO C

```c
int checkInternetConnection() {
    // Setup iniziale della funzione
    save current_state_of_registers;

    // Chiamata alla funzione di controllo della connessione
    int connectionState = InternetGetConnectedState(0, 0);

    // Verifica del risultato della connessione
    if (connectionState == 0) {
        // Gestione del caso di fallimento
        handle_failure();
    } else {
        // Gestione del caso di successo
        print("Success: Internet Connection\n");
        return 1;
    }

    // Epilogo della funzione
    restore_state_of_registers;
    return failure_value;  // Valore di fallimento
}
```

CODICE PASSO PASSO

- **.text:00401000 push ebp**: Salva il valore del registro base (ebp) sullo stack. Questo è necessario per poter ripristinare lo stato del registro base quando la funzione termina.
- **.text:00401001 mov ebp, esp**: Imposta il registro base (ebp) al valore corrente del puntatore dello stack (esp). Questo stabilisce un nuovo stack frame per la funzione.
- **.text:00401003 push ecx**: Salva il valore del registro ecx sullo stack. Questo è fatto per preservare il valore del registro ecx durante l'esecuzione della funzione.
- **.text:00401004 push 0 ; dwReserved**: Passa 0 come primo parametro alla funzione InternetGetConnectedState. Questo parametro è riservato e deve essere zero.
- **.text:00401006 push 0 ; lpdwFlags**: Passa 0 come secondo parametro alla funzione InternetGetConnectedState. Questo è un puntatore opzionale che può ricevere i flag che indicano lo stato della connessione. Qui è impostato a NULL (0).
- **.text:00401008 call ds:InternetGetConnectedState**: Chiama la funzione InternetGetConnectedState che verifica se il computer è connesso a Internet. Il risultato della funzione viene restituito nel registro eax.
- **.text:0040100E mov [ebp+var_4], eax**: Salva il valore restituito da InternetGetConnectedState (che si trova nel registro eax) in una variabile locale ([ebp+var_4]).
- **.text:00401011 cmp [ebp+var_4], 0** : Confronta il valore della variabile locale con 0. Questo verifica se il computer è connesso a Internet. Se il valore è 0, significa che non c'è connessione a Internet.
- **.text:00401015 jz short loc_40102B**: Se il confronto precedente è vero (cioè il valore è 0), salta all'etichetta loc_40102B, che gestisce il caso in cui non c'è connessione a Internet.
- **.text:00401017 push offset aSuccessInterne ; "Success: Internet Connection\n"**: Spinge l'indirizzo del messaggio "Success: Internet Connection\n" sullo stack. Questo è l'argomento per la funzione di stampa successiva.
- **.text:0040101C call sub_40105F**:Chiama una funzione che probabilmente stampa il messaggio di successo. L'indirizzo del messaggio è stato passato come parametro.
- **.text:00401021 add esp, 4**: Ripristina lo stack pointer (esp) rimuovendo il parametro della funzione di stampa (l'indirizzo del messaggio).
- **.text:00401024 mov eax, 1**: Imposta il valore di ritorno della funzione a 1, indicando successo (connessione a Internet presente).
- **.text:00401029 jmp short loc_40103A**: Salta all'etichetta loc_40103A, che probabilmente contiene l'epilogo della funzione.
- **.text:0040102B loc_40102B**: Etichetta che indica il punto di gestione del fallimento. Questa parte del codice verrebbe eseguita se non c'è connessione a Internet.

COME MIGLIORARLO

Possiamo migliorare il codice aggiungendo istruzioni per ripristinare i registri salvati e terminare la funzione, come:

```
pop ecx
mov esp, ebp
pop ebp
ret
```

- pop ecx: Ripristina il valore originale di ecx.
- mov esp, ebp: Ripristina lo stack pointer (esp) al valore originale.
- pop ebp: Ripristina il registro base (ebp).
- ret: Ritorna dalla funzione.

THANK YOU