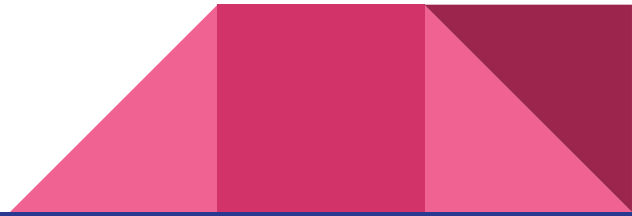
COSTRUTTI C - ASSEMBLY x86

S10L4

INDICE

- TRACCIA
- IDENTIFICO I COSTRUTTI
- ESECUZIONE AD ALTO LIVELLO
- BONUS



TRACCIA

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

```
.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
```

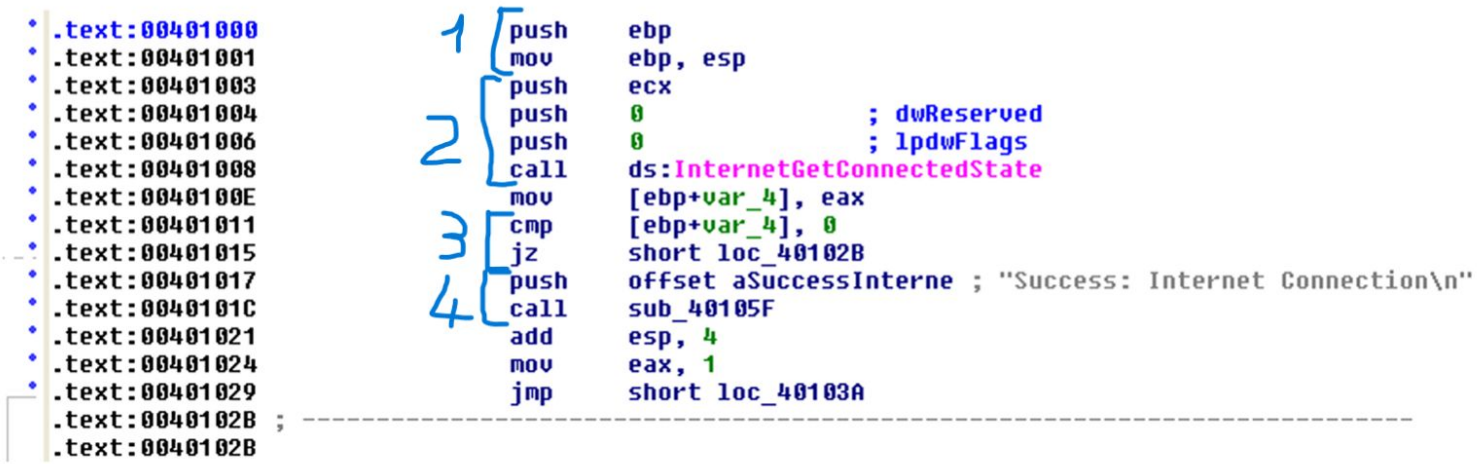
TRACCIA

Consegna:

1. Identificare i costrutti noti (es. while, for, if, switch, creazione/distruzione stack, ecc.)
2. Ipotizzare la funzionalità – esecuzione ad alto livello

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

IDENTIFICO I COSTRUTTI



Come evidenziato in figura abbiamo i seguenti 4 costrutti:

1. Creazione dello stack
2. Funzione chiama
3. Construtto “if”
4. Funzione chiama

ESECUZIONE AD ALTO LIVELLO

Possiamo ipotizzare che il programma faccia un collegamento ad una rete internet grazie alla funzione “InternetGetConnectedState”.

BONUS

“Push ebp” siginifica che mette il valore contenuto nel registro “ebp” sulla pila dello stack.

“mov ebp, esp” significa che <<muove>> il contenuto di esp in ebp.

“Push ecx” siginifica che mette il valore contenuto nel registro “ecx” sulla pila dello stack.

“Push 0; dwReserved” significa che mette il valore zero sulla pila dello stack. Mentre per quanto riguarda “dwReservede” credo sia un commento. “dw” lo posso identificare con la parola double-word.

“Push 0; lpdwFlags” uguale a sopra ma lpdw significa long-pointer-double-word

BONUS

“call ds:InternetGetConnectedState” siginfica che fa una chiamata ad una funzione che probabilmente si collegherà ad internet.

“mov [ebp+var_4], eax” significa che muove il contenuto del registro eax nella variabile [ebp+var_4].

“cmp [ebp+var_4], 0” significa che compara il valore zero con la variabile [ebp+var_4].

“jz short loc_40102B” significa che salta alla location 40102B se lo “zero flag” è true.

“Push offset aSuccessInternet” significa che inserisce nello stack la stringa aSuccessInternet.

BONUS

“Call sub_40105F” significa che fa una chiamata ad una subroutine 40105F.

“Add esp, 4” significa che aggiunge 4 Byte all’indicatore dello stack pointer ripristinando lo stack.

“Mov eax, 1” significa che imposta il registro eax ad 1.

“Jmp short loc_40103A” significa che salta alla location 40103A uscendo dal costrutto If-Else.

; —---------- è un commento.

GRAZIE