




# CONSEGNA S10-L4

Costrutti C in assembly x86



Nella consegna di oggi veniva chiesto di esaminare e spiegare i costrutti usati nel codice assembly che vedremo di seguito e di ipotizzare una possibile funzione di esso



```
push ebp
mov epb,esp
push ecx
push 0 ; dwreserved
push 0 ; ipdwFlags
call ds: InternetGetConnectedState
mov [ebp+var_4], eax
cmp [ebp+var_4], 0
jz short loc__40102B
push Offset aSuccesinterne
call sub_40105F
add esp,4
mov eax,1
jmp short loc_40103A
```



Il codice visto sopra usa alcuni comandi utilizzati anche in linguaggio C:

**mov:** Istruzione usata per copiare i dati da una zona all'altra. In questo codice viene usata per avviare il registro "ebp" e per immagazzinare il risultato della funzione "InternetGetStateConnected"

**call e Jump:** utilizzati per richiamare una funzione o spostare il controllo ad una subroutine specifica. Nel codice fornito "jmp" viene usato per saltare ad una determinata istruzione, mentre "call" si utilizza per richiamare la funzione "InternetGetStateConnected"

**Offset:** Viene utilizzata per ottenere l'etichetta all'interno di un programma. Nel nostro caso si usa per ottenere l'offset di una stringa da dare come parametro ad una subroutine per mandare un messaggio di successo

**push e pop:** Sono istruzioni usate per inserire o rimuovere valori dallo stack. In questo caso sono usati per mandare parametri alla funzione "InternetGetConnectedState"

**cmp e jz:** Queste due istruzioni sono utilizzate per confrontare due valori e raggiungere un'etichetta specifica nel caso fossero uguali. Nel nostro caso vengono utilizzate per verificare che il valore sia 0



Quanto visto e analizzato nel codice ci fa presupporre abbia a che fare con le attività di rete. Ciò vuol dire che, essendo questo codice un malware , possiamo ipotizzare il fatto che venga utilizzato per monitorare appunto le attività di rete di una determinata macchina target o di un server