Cybersecurity Specialist

Cyber Security & Ethical Hacking

ASSEMBLY X86

Alejandro Cristino
S10-L3

TRACCIA

Nella lezione teorica del mattino, abbiamo visto i fondamenti del linguaggio Assembly. Dato il codice in Assembly per la CPU x86 allegato qui di seguito, identificare lo scopo di ogni istruzione, inserendo una descrizione per ogni riga di codice. Ricordate che i numeri nel formato 0xYY sono numeri esadecimali. Per convertirli in numeri decimali utilizzate pure un convertitore online, oppure la calcolatrice del vostro computer (per programmatori).

```
0x00001141 <+8>:     mov    EAX,0x20
0x00001148 <+15>:    mov    EDX,0x38
0x00001155 <+28>:    add    EAX,EDX
0x00001157 <+30>:    mov    EBP, EAX
0x0000115a <+33>:    cmp    EBP,0xa
0x0000115e <+37>:    jge    0x1176 <main+61>
0x0000116a <+49>:    mov    eax,0x0
0x0000116f <+54>:    call   0x1030 <printf@plt>
```

SVOLGIMENTO

0x00001141 <+8>: mov EAX,0x20	**Questa istruzione trasferisce il valore esadecimale 0x20, corrispondente a 32 in decimale, nel registro EAX.**
0x00001148 <+15>: mov EDX,0x38	Il valore esadecimale 0x38, che equivale a 56 in decimale, viene caricato nel registro EDX
0x00001155 <+28>: add EAX,EDX	Questa istruzione somma il contenuto del registro EDX (56) al contenuto del registro EAX (32). Il risultato della somma, 88, viene memorizzato nuovamente in EAX. Questa operazione aggiorna EAX con il risultato della somma.
0x00001157 <+30>: mov EBP,EAX	Il valore attuale di EAX, che è 88, viene copiato nel registro EBP.
0x0000115a <+33>: cmp EBP,0xA	La funzione di confronto cmp viene utilizzata per confrontare il valore di EBP (88) con il valore esadecimale 0xA (10 in decimale). Questo confronto imposta i flag nel registro di stato, che influenzano il comportamento delle istruzioni condizionali successive.

SVOLGIMENTO

0x0000115e <+37>: jge 0x1176 <main+61>	**Se il valore in EBP è maggiore o uguale a 10, questa istruzione esegue un salto all'indirizzo 0x1176. Poiché 88 > 10, il salto viene eseguito, reindirizzando il flusso del programma alla sezione di codice specificata.**
0x0000116a <+49>: mov eax,0x0	Se il salto condizionale non viene eseguito, eax viene impostato a 0. Questo può essere utilizzato per segnalare un particolare stato o come valore di ritorno.
0x0000116f <+54>: call 0x1030 <printf@plt>	Questa istruzione chiama la funzione printf.

CONCLUSIONE

Questo segmento di codice assembly effettua una somma di due valori, poi utilizza il risultato per determinare il flusso successivo del programma tramite un confronto. Se il risultato è maggiore o uguale a 10, il programma salta a una parte specifica del codice. In caso contrario, imposta il registro eax a zero, il che potrebbe essere utilizzato per un’uscita condizionale o per un’ulteriore operazione di stampa tramite printf.