S10 L4

Assembly: tecniche di ingegneria inversa

Daniele Zizzi

.text:00401000 push ebp Salva il valore del registro ebp nello stack
.text:00401001 mov ebp, esp Copia il valore di esp in ebp
.text:00401003 push ecx Salva il valore del registro ecx nello stack
.text:00401004 push 0 ;dwReserved Serve per mantenere l'allineamento dello stack
.text:00401006 push 0 ;lpdwFlags Serve per mantenere l'allineamento dello stack
.text:00401008 call ds:InternetGetConnectedState Chiama la funzione e restituisce lo stato della connessione internet
.text:0040100E mov [ebp+var_4], eax Copia eax nella variabile var_4
.text:00401011 cmp [ebp+var_4], 0 Compara se il valore della variabile con 0, quindi verifica se la connessione è attiva. Se 0 non è attiva. Quindi come un if(a==0)
.text:00401015 jz short loc_40102B Se la connessione non è attiva, quindi uguale a 0, salta all'indirizzo di memoria indicato
.text:00401017 push offset aSuccessInterne ; "Success: Internet Connection\n" Mette il messaggio nello stack
.text:0040101C call sub_40105F Stampa il messaggio "Success: Internet Connection"
.text:00401021 add esp, 4 Aggiunge 4 al registro ESP ripulisce lo stack e prepara il valore di ritorno per la funzione
.text:00401024 inc eax, 1 EAX viene incrementato di uno ripulisce lo stack e prepara il valore di ritorno per la funzione
.text:00401029 jmp short loc_40103A Salta all'indirizzo di memoria

Il programma in questione, verifica se la macchina è raggiungibile dall’esterno. Attraverso la funzione InternetGetConnectedState, riceviamo in output lo stato della connessione, se attiva o meno. Questo però non vuol dire che la macchina sia vulnerabile o meno all’attacco che si sta per fare. Lo stato della connessione, può essere utilizzato da un hacker, per attaccare l’host in questione attraverso l’uso di bind shell, DDoS/DoS e l’utilizzo di exploit.