CONSEGNA U3 S11 L4

1. **Il tipo di Malware in base alle chiamate di funzione utilizzate.**

In base ad un'analisi delle funzioni, si potrebbe trattare di un **keylogger**. Lo si deduce dalla presenza della funzione **SetWindowsHook()**. Tra l'altro possiede anche la capacità di ottenere la persistenza sul sistema, infatti nel codice si trova la funzione **CopyFile()** per effettuare la copia del **path** dell'eseguibile nella **startup folder** di Windows.

2. **Evidenziate le chiamate di funzione principali aggiungendo una descrizione per ognuna di essa**
   - **SetWindowsHook():** serve per avviare uno procedura di **hook** all'interno del sistema, per monitorare un tipo di evento al verificarsi del quale avviene la sua registrazione. Gli eventi possono essere associati ad uno o più **threads** in esecuzione. Accetta come parametro la tipologia di **hook** da monitorare.
   - **CopyFile():** serve per effettuare il duplicato di un file sul disco. Essa può richiamare una funzione secondaria ogni qual volta una porzione della procedura di duplicazione è stata portata a termine.

3. **Il metodo utilizzato dal Malware per ottenere la persistenza sul sistema operativo**

Il malware tenta di ottenere la persistenza cercando di copiare il percorso del suo eseguibile all'interno della **startup folder** di sistema, per far sì che esso venga avviato ad ogni riavvio della macchina. Si vede infatti, che sullo **stack** è stato eseguito il **push** di vari parametri prima della chiamata della funzione **CopyFile().** Tra questi ci sono, appunto, il percorso della **startup folder** ed il percorso del malware. Questi parametri serviranno poi alla funzione per creare la copia dell'eseguibile nel **path** desiderato.

```
.text: 00401044   mov ecx, [EDI]          EDI = «path to
                                          startup_folder_system»
.text: 00401048   mov edx, [ESI]          ESI = path_to_Malware
.text: 0040104C   push ecx                ; destination folder
.text: 0040104F   push edx                ; file to be copied
.text: 00401054   call CopyFile();
```

4. **BONUS: Effettuare anche un'analisi basso livello delle singole istruzioni**
   push eax → push del registro eax sullo stack
   push ebx → push del registro ebx sullo stack
   push ecx → push del registro ecx sullo stack
   push WH_Mouse → esegue il push sullo stack della procedura hook per registrare gi eventi scatenati dal movimento del puntatore
   call SetWondowsHook() → chiama la funzione che si occupa di gestire gli hook dopo aver caricato sullo stack tutti i suoi parametri

xor ecx, ecx → tramite l'operatore logico xor si azzera il contenuto del registro ecx
mov ecx, [edi] → copia l'indirizzo puntato dal registro edi all'interno del registro ecx
mov edx, [esi] → copia l'indirizzo puntato dal registro esi all'interno del registro edx
push ecx → esegue il push del registro ecx sullo stack
push edx → esegue il push del registro edx sullo stack
call CopyFile() → chiama la funzione per effettuare la copia dei file su disco dopo aver fatto il push dei parametri necessari sullo stack