S11 LEZIONE 4

Traccia:
La figura nella slide successiva mostra un estratto del codice di un malware.
Identificate:

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

2. Evidenziate le chiamate di funzione principali aggiungendo una descrizione per ognuna di essa.

3. Il metodo utilizzato dal Malware per ottenere la persistenza sul sistema operativo 4.

4. BONUS: Effettuare anche un'analisi basso livello delle singole istruzioni.

```
.text: 00401010    push eax
.text: 00401014    push ebx
.text: 00401018    push ecx
.text: 0040101C    push WH_Mouse                ; hook to Mouse
.text: 0040101F    call SetWindowsHook()
.text: 00401040    XOR ECX,ECX
.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();
```

1-2. In base alle chiamate a funzione utilizzate dal malware preso in esame è plausibile ipotizzare che sia un trojan
I trojan sono un tipo di malware che si presenta come un software legittimo ma il reale scopo è quello di infettare il sistema operativo.

Le chiamate a funzione sono:

- ***SetWindowsHook():*** questa funzione viene utilizzata per installare un hook sul sistema operativo, esso è un programma che viene eseguito ogni qual volta che si verifica un evento specifico, in questo caso si parla del click del mouse (quindi ogni click del mouse viene monitorato)

- ***CopyFile():*** questa funzione viene invece utilizzata per copiare un file, in questo caso il malware fa un copia di se stesso nella cartella di avvio del sistema operativo. In tal modo il malware viene eseguito ad ogni avvio della macchina infettata.

3. Il malware ottiene persistenza sulla macchina vittima eseguendo una copia di se stesso tramite la funzione *CopyFile* nella cartella di avvio.

4.

push eax : mette il contenuto di eax nello stack della memoria

push ebx : mette il contenuto di ebx nello stack della memoria

push ecx : mette il contenuto di ecx nello stack della memoria

push WH_Mouse : mette il valore della variabile WH_Mouse nello stack della memoria, è un'etichetta/puntatore ad una funzione

call SetWindowsHook() : chiama la funzione

xor ecx, ecx : resetta il registro ecx

mov ecx, [edi] : carica il contenuto della memoria all'indirizzo edi nel registro ecx

mov edx, [esi] : fa la stessa cosa di sopra

push ecx : mette ecx nello stack

push edx : mette edx nello stack

call CopyFile() : chiamata a funzione