FUNZIONALITÀ DEI MALWARE

In base all'estratto del codice di un malware

.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();	

- **Il tipo di Malware in base alle chiamate di funzione utilizzate.**
- **Evidenziate le chiamate di funzione principali aggiungendo una descrizione per ognuna di essa**
- **Il metodo utilizzato dal Malware per ottenere la persistenza sul sistema operativo**
- **BONUS: Effettuare anche un'analisi basso livello delle singole istruzioni**

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

Il tipo di Malware in base alle chiamate di funzione utilizzate può essere considerato un **Keylogger**

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

.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();	

 # call SetWindowsHook() – Funzione che installa un metodo "**hook**" dedicato al monitoraggio degli eventi in input di una data periferica. In questo caso il mouse

 # call CopyFile() – Funzione che copia un file esistente in un nuovo file

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

.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();	

Il Malware ottiene la persistenza tramite queste operazioni.

Inizializza il registro **ECX** tramite l'operatore **XOR** (che restituisce **0** se sorgente e destinatario sono uguali)

Una volta azzerato inserisce al suo interno **EDI** (path della cartella startup), stessa cosa con il registro **EDX** dove copia **ESI** (la directory del malware). Una volta che ha impostato "sorgente" e "destinazione" si copia all'interno della cartella startup per ottenere la persistenza all'avvio del os, tramite la funziona **call CopyFile()**.

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

Push eax – Inserisce in cima allo stack di memoria il registro eax

Push ebx - Inserisce in cima allo stack di memoria il registro ebx

Push ecx - Inserisce in cima allo stack di memoria il registro ecx

Push WH_Mouse - Inserisce in cima allo stack di memoria il registro WH_Mouse (monitoraggio periferica mouse)

Call SetWindowsHook() – Chiama la funzione SetWindowsHook, che monitora le periferiche indicate dall'istruzione precedente WH_Mouse

XOR ECX,ECX – Azzera il contenuto del registro ECX (tramite oeratore logico XOR)

Mov ecx, [EDI] – Copia il contenuto dell'alloggio di memoria sorgente (EDI), in quello destinatario (ecx)

Mov edx, [ESI] - Copia il contenuto dell'alloggio di memoria sorgente (ESI), in quello destinatario (edx)

Push ecx - Inserisce in cima allo stack di memoria il registro ecx

Push edx - Inserisce in cima allo stack di memoria il registro edx

Call CopyFile() – Chiama la Funzione Copyfile