Malware Analysis

Richiesta: Dato il seguente estratto in assembly 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();
```

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

Tipo di Malware

Il codice sembra trattarsi di un keylogger in base alla funzione chiamata SetWindowsHookEx(): Questa è una funzione del sistema Windows che permette di installare un hook. Un hook è un meccanismo che permette di intercettare eventi del sistema, come la pressione dei tasti o i movimenti del mouse.
In questo frammento di codice viene utilizzato come parametro WH_MOUSE, il che indica che l'hook è destinato a monitorare gli eventi del mouse.

Chiamate di funzione e persistenza

Oltre alla chiamata di funzione vista in precedenza, utilizzata per intercettare determinati eventi di sistema, troviamo:
CopyFile(): Questa funzione copia un file da una posizione specificata a un'altra. Nel frammento di codice in questione, sembra che il malware stia tentando di copiare se stesso in una posizione che sarà eseguita automaticamente all'avvio del sistema, in modo tale da ottenere la persistenza.

Bonus analisi basso livello

push eax: Questa istruzione inserisce il contenuto del registro EAX (che potrebbe contenere un dato o un indirizzo) nella cima dello stack.

push ebx: Questa istruzione mette il contenuto del registro EBX nello stack.

push ecx: Questa istruzione sposta il contenuto del registro ECX nello stack.

push WH_Mouse: Qui viene passata WH_Mouse come parametro per la successiva chiamata di funzione.

call SetWindowsHookEx(): Questa istruzione chiama la funzione SetWindowsHookEx, passando i valori che sono stati inseriti nello stack (i registri e WH_Mouse). La funzione installerà un hook che monitora gli eventi del mouse.

XOR ECX,ECX: Un'operazione XOR tra due registri identici (in questo caso ECX) azzera il registro.

mov edx, [ESI]: Simile alla precedente, questa istruzione copia il dato puntato da ESI nel registro EDX. ESI potrebbe contenere l'indirizzo del percorso del malware.

push ecx: Mette il contenuto del registro ECX (ora contenente il percorso della cartella di destinazione) nello stack.

push edx: Mette il contenuto del registro EDX (ora contenente il percorso del file malware) nello stack.

call CopyFile(): Questa chiamata a funzione invoca CopyFile, che copierà il file dal percorso indicato in EDX al percorso indicato in ECX.