# Epicode Unit 3 Week 3 L4

# Malware Analysis

**Traccia:**

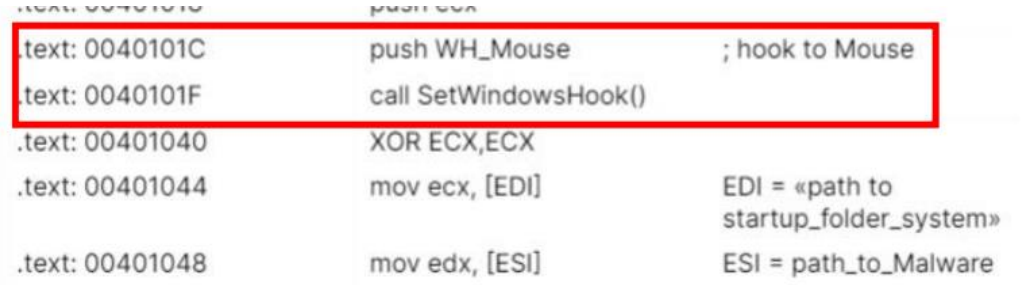
La figura nella slide successiva mostra un estratto del codice di un malware.
Identificate:

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

**Figura 1:**

| | | |
|---|---|---|
| .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) **Non risulta complesso, vista l'individuazione della funzione "SetWindowsHook", ed il commento "hook to Mouse" ipotizzare che il Malware in analisi sia un Keylogger.**



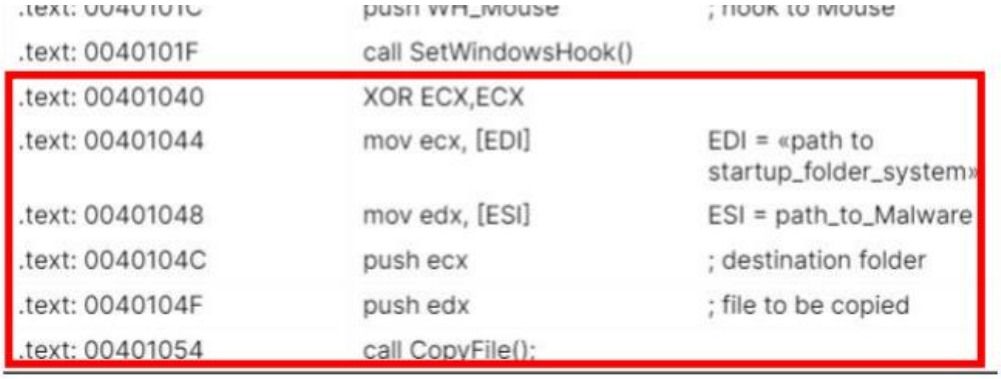

2) **Nel codice assembly fornito, si notano due funzioni:**

- **"SetWindowsHook()", già citata precedentemente, che usa il metodo "hook", appunto, per agganciare come dice il nome stesso la funzione a cui il parametro viene passato ad una data periferica, un mouse in questo caso.**
- **La funzione "CopyFile()", utilizzata per copiare un file esistente in**

un nuovo file, potenzialmente nascosto, per ottenere la Persistenza sulla macchina infetta.

| | | |
|---|---|---|
| .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_syster |
| .text: 00401048 | mov edx, [ESI] | ESI = path_to_Malwar |
| .text: 0040104C | push ecx | ; destination folder |
| .text: 0040104F | push edx | ; file to be copied |
| .text: 00401054 | call CopyFile(); | |

3) Si può notare che l'eseguibile oggetto di analisi utilizzi la funzione "CopyFile" per andarsi a copiare nel path che si può individuare dai commenti: "path to startup_folder_system", una directory che contiene gli eseguibili avviati in automatico quando la macchina viene accesa.



| | | |
|---|---|---|
| .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(); | |

**BONUS:**

Analisi a basso livello del codice Assembly fornito.

- Push EAX: inserisce il registro EAX in cima alla stack;
- Push EBX: inserisce il registro EBX in cima alla stack;
- Push ECX: inserisce il registro ECX (probabile accumulatore) in cima alla stack;
- Push WH_Mouse: inserisce l'hook del mouse in cima alla stack per poterlo passare alla funzione;
- Call WindowsHook(): Chiama la funzione necessaria al monitoraggio della periferica
- XOR ECX, ECX: l'operatore logico "OR esclusivo" azzera il registro ECX

- **Mov ECX, [EDI]: copia il contenuto del registro EDI nel registro ECX;**
- **Mov EDX, [ESI]: copia il contenuto del registro ESI nel registro EDX, dai commenti notiamo che le ultime due istruzioni vengono utilizzate come già anticipato per l'ottenimento della Persistenza sul Sistema infetto;**
- **Push ECX: mette in cima alla stack il registro ECX, che contiene la directory di destinazione del file;**
- **Push EDX: mette in cima alla stack il registro EDX, contenente il file da copiare nella directory precedente;**
- **Call CopyFile(): chiama la funzione "CopyFile" per copiare il malware nella directory di destinazione.**