S11-L4

Traccia:

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

Codice da analizzare:

.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();	

Task 1

Dal codice possiamo apprezzare la funzione SetWindowsHook(); questa funzione permette di monitorare il sistema per determinati tipi di eventi.
Può essere utilizzata per intercettare, ad esempio, input della tastiera (utile per i Keylogger).

Task 2

- SetWindowsHook(), funzione precedentemente analizzata, qui possiamo vedere come sia stata caricata nello stack WH_Mouse, lasciando intendere come il malware stia cercando di monitorare gli eventi riguardanti il mouse;
- CopyFile(), questa funzione permette di copiare un file. Prima di copiare il file è stato inserito nello stack sia il percorso di destinazione che il percorso del file da copiare.

Task 3

La persistenza viene ottenuta dato che il malware copia se stesso nel path “startup_folder_system”, una cartella che gli permetterà di avviarsi automaticamente quando il sistema viene avviato.

Task 4

- push eax, ebx, ecx — Carica il contenuto di questi registri nello stack;
- push MH_Mouse — Carica MH_Mouse nello stack. Questo parametro ci servirà dopo (in SetWindowsHook);
- call SetWindowsHook — Spiegato nella task 1
- xor ecx ecx — Poichè compara un registro a se stesso è come se stesse pulendo (azzerando) il registro
- mov ecx, [EDI] — Sposta il valore dell’indirizzo EDI nel registro ecx
- mov edx [ESI] — Sposta il valore dell’indirizzo ESI nel registro edx
- push ecx edx — carichiamo i due valori nello stack
- call CopyFile — Spiegata nella task 2