Floriana Feminò

19/01/2023

MALWARE ANALYSIS

ANALISI COMPORTAMENTALE DI CATEGORIE DI MALWARE NOTE

Tasks:

1. **Identificazione del tipo di malware in base alle chiamate di funzione utilizzate**
2. **Identificazione e descrizione delle chiamate di funzione principali**
3. **Individuazione del metodo utilizzato dal malware per ottenere la persistenza sul sistema operativo**
4. **Analisi 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();	

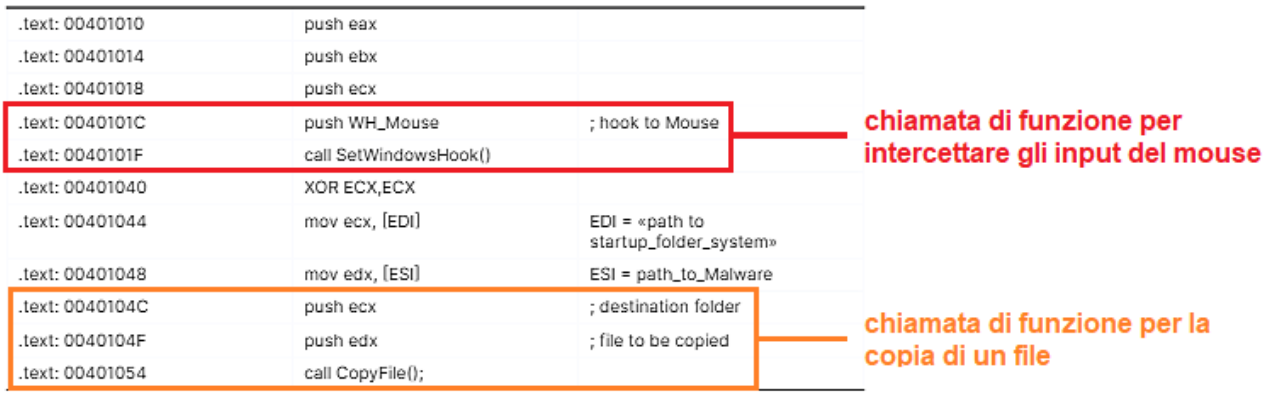
---

1. Identificazione del tipo di malware in base alle chiamate di funzione utilizzate

Analizzando le istruzioni Assembly, possiamo ipotizzare che il malware in analisi sia un **Keylogger** per via della presenza dell'istruzione *push WH_Mouse* e della chiamata di funzione immediatamente successiva *call SetWindowsHook()*:

.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();	

2. Identificazione e descrizione delle chiamate di funzione principali



Le principali chiamate di funzione sono:

call SetWindowsHook() – funzione che installa un metodo "**hook**" dedicato al monitoraggio degli eventi di una determinata periferica, in questo caso il mouse, come si vede dalla precedente istruzione "push WH_Mouse". Un hook è un punto all'interno del meccanismo di gestione degli eventi in cui un'applicazione può installare una subroutine per monitorare certi tipi di eventi. È possibile utilizzare l'hook WH_Mouse per monitorare gli input del mouse.

call CopyFile() – funzione che si occupa di copiare un file esistente in un nuovo file.

3. Individuazione del metodo utilizzato dal malware per ottenere la persistenza sul sistema operativo

Il malware ottiene la persistenza attraverso la copia del suo eseguibile nella **Startup Folder** (= cartella di avvio), che contiene i processi che il sistema operativo inizializzerà al suo avvio. Vediamo come:

.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();	

Prima di tutto, il registro ECX viene inizializzato a zero tramite l'operatore **XOR**. Successivamente, tramite l'istruzione mov, il malware copia il contenuto di **EDI** (path della cartella di startup) all'interno del sopracitato registro ECX e, analogamente, copia il contenuto di **ESI** (directory del malware) all'interno del registro EDX. Una volta impostata la destinazione e la sorgente dell'operazione di copia, i valori dei registri ECX ed EDX vengono pushati in cima allo stack e, tramite la chiamata alla funzione **CopyFile()**, il malware copia il suo eseguibile nella Startup Folder per ottenere la persistenza (= l'esecuzione automatica all'avvio del sistema operativo).

4. Analisi 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 EBX

Push WH_Mouse – Inserisce in cima allo stack di memoria l'hook WH_Mouse per il monitoraggio della periferica mouse)

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

XOR ECX, ECX – Azzera il contenuto del registro ECX tramite operatore logico XOR

Mov ECX, [EDI] – Copia il contenuto dell'indirizzo di memoria sorgente [EDI] nel registro ECX

Mov EDX, [ESI] – Copia il contenuto dell'indirizzo di memoria sorgente [ESI] nel registro 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()*