CONSEGNA S11/L4

Riccardo Lattanzi
Alex Fiorillo
Maria Huapaya
Giulio Zanet

Giuseppe Lupoi
Davide Caldirola
Michael Bonifazi

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

TRACCIA

.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. Analizzando il codice del malware a noi fornito quest'oggi possiamo individuare nella 4° riga la funzione "**push WH_Mouse ;hook to Mouse**" che come visto nella lezione di oggi ci fa capire che stiamo trattando un **keylogger**, in questa funzione vengono **rilevati i movimenti ed i click del mouse** sulla macchina target.

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();	

2. Al secondo punto della traccia vedremo le funzioni principali di questo codice. Dopo aver visto la funzione in riga 4 possiamo identificare in riga 5 la chiamata di funzione “**call SetWindowsHook()**” che va appunto a richiamare la riga sopra dove viene creato lo stack che terrà traccia del mouse.

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();	

3. Al terzo punto vedremo come il malware ottiene la persistenza sul sistema, nelle due righe di codice evidenziate dal rettangolo rosso, possiamo notare come i **valori di EDI ed ESI** vengano rispettivamente copiati nei **registri ECX e EDX** dove andrà a scrivere il **path del malware direttamente nel path dello startup del sistema**. Quest'azione permetterà al malware di avviarsi direttamente all'avvio del sistema operativo.

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();	

BONUS

push eax: questo inserisce in cima allo stack di memoria il registro **eax**.

push ebx: questo inserisce in cima allo stack di memoria il registro **ebx**.

push ecx: questo inserisce in cima allo stack di memoria il registro **ecx**.

push WH_Mouse: questo inserisce in cima allo stack di memoria l'**hook WH_Mouse** per aver traccia della periferica mouse.

call SetWindowsHook: questo chiama la funzione **SetWindowsHook** che tiene traccia delle periferiche indicate nella precedente istruzione.

xor ecx, ecx: l'operatore logico **xor** azzera il contenuto del registro **ecx**.

mov ecx, [EDI]: copia nel registro **ecx** il contenuto dell'indirizzo di memoria **EDI**.

mov edx, [ESI]: copia nel registro **edx** il contenuto dell'indirizzo di memoria **ESI**.

push ecx: questo inserisce in cima allo stack di memoria il registro **ecx**.

push edx: questo inserisce in cima allo stack di memoria il registro **edx**.

call CopyFile(): chiama la funzione **CopyFile**.