Esercizio 06-04

Traccia: la figura sotto mostra un estratto del codice di un malware. Identificare:

1. Il tipo di Malware in base alle chiamate di funzione utilizzate.
2. Evidenziare 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

.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.

Dalle chiamate di funzione che effettua il Malware nell'estratto di codice si deduce che è un keylogger che cattura gli input del Mouse del dispositivo della vittima e in maniera persistente.

Nei prossimi punti viene approfondita la spiegazione a livello logico del perchè è un keylogger

2.

Le funzioni principali chiamate dal Malware sono:

- La funzione SetWindowsHook(), è una funzione che installa un "hook" (una funzione) su una data periferica per monitorarne gli eventi, la periferica "vittima" in questo caso e il Mouse del PC dove il Malware è in esecuzione.
- La funzione CopyFile(), è una funzione che copia un file in una sorgente, questi due parametri devono essere forniti ("pushati sullo stack") precedentemente alla chiamata

.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.

Nel caso del codice proposto, al fine di ottenere la persistenza, il Malware utilizza il metodo "Startup folder": il path del Malware viene copiato all'interno della cartella dei sistemi che si avviano all'accensione del PC, così da catturare i dati della periferica monitorata (in questo caso il Mouse) fin dall'accensione del dispositivo.

.text: 00401018	push ecx	
.text: 0040101C	push WH_Mouse	; hook to Mouse
.text: 0040101F	call SetWindowsHook()	
.text: 00401040	XOR ECX,ECX	

BONUS.

Analisi di basso livello dell'estratto di codice:

Push eax	=>	inserimento sullo stack del parametro contenuto in eax
Push ebx	=>	inserimento sullo stack del parametro contenuto in ebx
Push ecx	=>	inserimento sullo stack del parametro contenuto in ecx
Push WH_Mouse	=>	inserimento del parametro WH_Mouse, specifica alla funzione la periferica da monitorare
Call SetWindowsHook()	=>	chiamata della funzione che avvia l'attività di monitorazione sulla periferica specificata precedentemente
XOR ECX, ECX	=>	azzeramento del registro ECX grazie alla istruzione XOR, che resituisce "0" in output quando riceve in input due valori di bit uguali
Mov ecx, [EDI]	=>	copia del valore contenuto in [EDI] nel registro ecx, in questo caso il valore di [EDI] è il path per arrivare allo "startup_folder_system"
Mov edx, [ESI]	=>	copia del valore contenuto in [ESI] nel registro edx, in questo caso il valore di [ESI] è il path del Malware
Push ecx	=>	inserimento sullo stack del parametro contenuto in ecx (path di destinazione)
Push edx	=>	inserimento sullo stack del parametro contenuto in edx (file da copiare)
Call CopyFile();	=>	chiamata alla funzione che copia un file in una destinazione, i due parametri vengono precedentemente inseriti sullo stack tramite l'istruzione "push"