Esercizio S11-L4

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();
```

Risposta al quesito 1-2:

Il primo quesito ci richiede di identificare il tipo di malware in base al tipo di funzioni richiamate dal codice. Andiamo dunque ad analizzare il codice fornito ed in particolare le funzioni in esso citate.

```
.text: 0040101C    push WH_Mouse          ; hook to Mouse
.text: 0040101F    call SetWindowsHook()
```

Come possiamo vedere in questo estratto il codice passa allo stack il parametro WH_Mouse e poi chiama la funzione SetWindowsHook. La funzione SetWindowsHook Installa una procedura di hook definita dall'applicazione in una catena di hook. In genere una procedura di hook viene installata per monitorare il sistema per determinati tipi di eventi. Questi eventi sono associati a un thread specifico o a tutti i thread nello stesso desktop del thread chiamante. A questa funzione viene passato il parametro WH_Mouse, che installa una procedura di hook che monitora i messaggi del mouse. Da questa funzione ci è possibile determinare che il programma è un malware di tipologia keylogger, che in questo caso registra le interazioni effettuate col mouse.

Risposta al quesito 2-3:

Continuiamo l'analisi del codice prendendo sotto osservazione la prossima funzione chiamata.

.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();	

Vediamo che il codice chiama la funzione CopyFile(), questa è una funzione molto semplice che copia il contenuto di un file esistente in un nuovo file. Possiamo vedere che vengono passati due parametri a questa funzione ovvero il contenuto di edx ed ecx, rispettivamente sono la sorgente e la destinazione del file da copiare.

Risposta al quesito 3:

.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();	

Dalle prime due righe di codice possiamo osservare che la sorgente del file da copiare corrisponde al path dov'è contenuto il file malware, mentre la destinazione della copia del malware sarebbe lo startup_folder_system. Questo è un folder molto particolare del sistema, la sua peculiarità è che tutti i programmi al suo interno vengono automaticamente eseguiti all'avvio della macchina. Possiamo dunque dedurre che lo scopo di questo codice è di assicurare la persistenza al malware facendo in modo che venga eseguito automaticamente dalla macchina all'avvio.