CONSEGNA S11L4

L’obbiettivo di oggi, con riferimento all’estratto del codice di un malware , è quello di Identificare:

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

Il codice è il seguente:

```
.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) In base alla funzione chiamata, la **SetWindowsHook()**, possiamo dedurre che si tratti si un **Keylogger**. Questa funzione installa un metodo (hook) dedicato al monitoraggio degli eventi di una periferica, che in questo caso è il mouse (vedi ; hook to mouse). Ogni volta che l’utente usa il dispositivo, il metodo verra allertato e salverà le informazioni su un file.

2) **Le funzioni principali sono:**

SetWindowsHook(): La funzione SetWindowsHook() in Windows è una funzione di programmazione dell'API di Windows che permette di installare un **hook**. Un hook è un meccanismo tramite il quale un'applicazione può intercettare eventi o messaggi destinati ad altre applicazioni o al sistema operativo stesso. Questo permette di monitorare o modificare il comportamento delle applicazioni o del sistema operativo per eventi specifici, come la pressione dei tasti, i movimenti del mouse, il cambiamento della configurazione del sistema, e così via.

La funzione è stata aggiornata con **SetWindowsHookEx()**, che offre una maggiore flessibilità e controllo nell'installazione dei hook. Consente agli sviluppatori di specificare esattamente il tipo di hook, la funzione di callback del hook, l'applicazione o il processo target, e se l'hook deve essere applicato a livello di thread o di sistema.

Il parametro **WH_Mouse** ci va ad indicare che è stato installato un look per il mouse.

CopyFile(): La funzione CopyFile nelle API di Windows è utilizzata per copiare un file da una posizione sorgente a una destinazione. Questa funzione offre un modo semplice e diretto per copiare file senza dover aprire manualmente i file sorgente e destinazione, leggere il contenuto del file sorgente e scrivere tale contenuto nel file destinazione.

3) Il malware ottiene la persistenza tramite le stratup folder, cartelle del'sistema operativo che vengono controllate all'avvio del sistema, i programmi al suo interno vengono quindi eseguiti. Queste cartelle possono essere generiche per il sistema, o dedicate agli utenti

```
.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();
```

Fig.1

(quindi la cartella è diversa per ogni utente del sistema). (Fig.1)

4). **Ecco il codice spiegato riga per riga**

	A	B	C	D	E	F	G
1	**Indirizzo**	**Codice**	**Desxcrizione**				
2	.text: 00401010	push eax	Salva il valore attuale del registro EAX sullo stack.				
3	.text: 00401014	push ebx	Salva il valore attuale del registro EBX sullo stack.				
4	.text: 00401018	push ecx	Salva il valore attuale del registro ECX sullo stack.				
5	.text: 0040101C	push WH_Mouse	Mette l'identificativo per un hook del mouse sullo stack.				
6	.text: 0040101F	call SetWindowsHookEx()	Installazione di un hook di tipo mouse.				
7	.text: 00401040	xor ecx, ecx	Azzera il registro ECX impostandolo a 0.				
8	.text: 00401044	mov ecx, [EDI]	Copia il contenuto della memoria puntata da EDI in ECX.				
9	.text: 00401048	mov edx, [ESI]	Copia il contenuto della memoria puntata da ESI in EDX.				
10	.text: 0040104C	push ecx	Mette il percorso della cartella di destinazione sullo stack.				
11	.text: 0040104F	push edx	Mette il percorso del file sorgente sullo stack.				
12	.text: 00401054	call CopyFile	Chiama la funzione CopyFile con i percorsi forniti.				