S11/L4

EPICODE

Pratica S11/L4 PDF

Esercizio
Funzionalità Malware

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

3

EPICODE

Premi Esc per uscire dalla modalità a schermo intero

Esercizio
Funzionalità Malware

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();
```

4

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

Visualizzando le istruzioni di Assembly, si può ipotizzare che il malware sia un Keylogger per la presenza dell'istruzione push WH_Mouse e della chiamata della funzione successiva call SetWindowshook():

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. Evidenziate le chiamate di funzione principali aggiungendo una descrizione per ognuna di essa

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();	

In giallo: chiamata di funzione per intercettare gli input del mouse.

In viola: chiamata di funzione per la copia di un file.

Le principali funzioni chiamate sono:

- "call SetWindowsHook()" - Questa funzione installa un "hook", un tipo di monitoraggio degli eventi, specificamente per il mouse. L'istruzione "push WH_Mouse" indica che il hook verrà utilizzato per monitorare gli eventi del mouse. Un hook è un punto nel sistema in cui un'applicazione può installare una subroutine per osservare determinati tipi di eventi.
- "call CopyFile()" - Questa funzione copia un file esistente in un nuovo file.

3. Il metodo utilizzato dal Malware per ottenere la persistenza sul sistema operativo
Il software dannoso ottiene la capacità di rimanere attivo nel sistema copiando il suo file eseguibile nella cartella di avvio, nota anche come "Startup Folder". Questa cartella contiene i programmi che il sistema operativo avvia automaticamente all'avvio.

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();	

4

Il registro ECX viene azzerato utilizzando l'operatore XOR. Successivamente, tramite l'istruzione "mov", il malware copia il percorso della cartella di avvio dalla posizione di memoria EDI al registro ECX e, allo stesso modo, copia la directory del malware dalla posizione di memoria ESI al registro EDX. Una volta che la destinazione e la sorgente per l'operazione di copia sono state impostate, i valori dei registri ECX ed EDX vengono spinti nello stack e, attraverso la chiamata alla funzione CopyFile(), il malware copia il suo eseguibile nella cartella di avvio per garantire l'esecuzione automatica al momento dell'avvio del sistema operativo.

4. BONUS: Effettuare anche un'analisi basso livello delle singole istruzioni

- "Push EAX": Mette il valore del registro EAX nello stack di memoria.
- "Push EBX": Mette il valore del registro EBX nello stack di memoria.
- "Push ECX": Mette il valore del registro ECX nello stack di memoria.

- "Push WH_Mouse": Mette l'hook WH_Mouse nello stack di memoria per il monitoraggio del mouse.
- "Call SetWindowsHook()": Chiama la funzione SetWindowsHook per monitorare le periferiche indicate.
- "XOR ECX, ECX": Azzeramento del contenuto del registro ECX tramite l'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": Mette il valore del registro ECX nello stack di memoria.
- "Push EDX": Mette il valore del registro EDX nello stack di memoria.
- "Call CopyFile()": Chiama la funzione CopyFile().