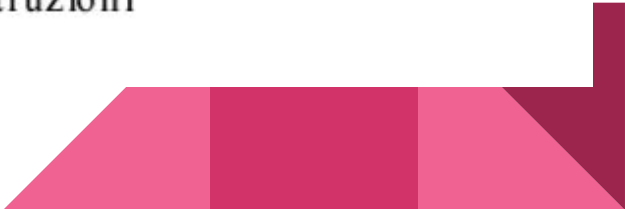


Esercizio S11/L4

Analisi comportamentale di un 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
- 

.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();	

Nel primo punto ci viene chiesto di identificare il tipo di malware. In base alle chiamate di funzione possiamo dire che si tratti di un keylogger, cioè un malware che intercetta gli input/output delle periferiche della macchina infetta, come ad esempio un mouse o una tastiera. Questo possiamo dedurlo dalla funzione “SetWindowsHook” a cui viene passato il parametro “WH_Mouse”, ovvero il mouse del computer. La funzione permettere di installare un metodo (hook) con cui monitorare gli eventi di una periferica.

.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();	

Il codice contiene due funzioni principali una di cui abbiamo discusso nella slide precedente, mentre la seconda è “CopyFile”, evidenziata nella figura accanto, che permette al malware di ottenere la persistenza. Inizialmente viene copiato il percorso della cartella di startup di sistema nel registro “ecx” e poi il percorso del malware in “edx”. In questo modo il malware verrà eseguito all’avvio del sistema indipendentemente dall’utente usato per loggarsi. Poi questi parametri vengono passati alla funzione tramite i comandi “push”. Infine la funzione viene chiamata tramite comando “call”.

.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

Le istruzioni “push” servono per passare i parametri o le variabili alle funzioni. Il comando “call” ha il compito di chiamare ed eseguire le funzioni. L’istruzione “xor”, detta or esclusivo, restituisce vero solo se uno solo dei due operandi è vero; in caso contrario restituirà falso. Lo “xor” viene usato per azzerare il registro “ecx”. I comandi “mov” servono per copiare l’operando sorgente nell’operando destinazione.

.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();	
