REPORT 19/01/2023

FACCILONGO DOMENICO

FUNZIONALITA' DEI MALWARE

TASK

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

ANALISI E VALUTAZIONE

In riferimento alla figura di seguito andare a rispondere ai quesiti dettati dalla task:

.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. **IDENTIFICARE IL TIPO DI MALWARE**
   Analizzando il codice, sulla base delle chiamate di funzioni utilizzate da esso, possiamo dire che la sua funzione è quella di creare un <<**keylogger**>>, da cui ne deriva il suo nome, con <<**persistenza**>> programmato per intercettare tutto ciò che l'utente della macchina infetta digita, in questo caso, con mouse, all'avvio del sistema operativo.

2. **EVIDENZIARE E DESCRIVERE LE CHIAMATE DI FUNZIONI PRINCIPALI**
   Possiamo notare nel codice che si hanno due chiamate di funzioni principali, e che sono:

.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();	

- <<**call SetWindowsHook()**>>: Questa è una chiamata di funzione utilizzata per installare un metodo chiamato **hook**, dedicato al monitoraggio degli eventi di una data periferica, come nel nostro caso *mouse*. Questo metodo verrà eseguito ogni qual volta che l'utente digiterà un tasto sul mouse e ne salverà le informazioni su di un file log.

- <<**call CopyFile()**>>: Chiamata di funzione con finalità di copiare il file eseguibile in un nuovo file ad un determinato path.

3. DESCRIVERE IL METODO UTILIZZATO DAL MALWARE PER OTTENERE LA PERSISTENZA

```
.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 malware in questione sfrutta la persistenza tramite la tecnica di utilizzo della <<**Startup folder**>>. Quest'ultima è una cartella del sistema operativo, che viene controllata all'avvio del sistema, ed i programmi al suo interno vengono eseguiti. I sistemi Windows possiedo due tipi di cartelle startup: **dedicata agli utenti** e **generica del sistema operativo**. Nel nostro caso fa parte della cartella generica del sistema operativo, il cui malware, se riesce correttamente a copiare se stesso all'interno della stessa cartella, verrà eseguito di conseguenza ed automaticamente all'avvio del sistema.

4. EFFETTUARE UN ANALISI DELLE SINGOLE ISTRUZIONI

ISTRUZIONI	SIGNIFICATO
push eax	Spinge il registro eax in cima allo stack
push ebx	Spinge il registro ebx in cima allo stack
push ec	Spinge il registro ecx in cima allo stack
push WH_Mouse ; hook to Mouse	Spinge la procedura **hook** in cima allo stack
call SetWindowsHook()	Chiamata di funzione alla funzione **SetWindowsHook()** per monitorare gli input del mouse
XOR ecx, ecx	Operatore logico che mette a confronto due operandi e darà 0 come risultato se sorgente e destinazione sono uguali

mov ecx, [EDI] EDI = <<path to startup_folder_system>>	Copia il contenuto del puntatore di registro di memoria edi nel registro ecx
mov edx, [ESI] ESI = path_to_Malware	Copia il contenuto del puntatore di registro di memoria esi nel registro edx
push ecx ; destination folder	Spinge il registro ecx in cima allo stack
push edx ; file to be copied	Spinge il registro edx in cima allo stack
call CopyFile();	Chiamata di funzione che copia il file esistente in un nuovo file