Epicode
CS-0124
Pratica S11/L4

Francesco Ficetti

1. Traccia..... 3
2. Svolgimento..... 4
2.1 Tipo di malware..... 4
2.2 Funzioni..... 4
2.3 Persistenza..... 4
2.4 Analisi del codice..... 4

1. Traccia

La figura allegata mostra un estratto del codice di un malware. Identificare:

1. Il tipo di malware in base alle chiamate di funzione utilizzate.
2. Le chiamate di funzione principali aggiungendo una descrizione per ognuna di esse.
3. Il metodo utilizzato dal malware per ottenere la persistenza sul sistema operativo.
4. BONUS. Effettuare 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();	

2. Svolgimento

2.1 Tipo di malware

In base alle chiamate di funzione utilizzare, si può ipotizzare che questo malware sia un keylogger. Lo si può intendere dalla chiamata alla funzione *SetWindowsHooks()*.

2.2 Funzioni

SetWindowsHooks: questa funzione allerta il metodo *hook* ogni qualvolta l'utente digita un tasto sulla tastiera e salva le informazioni su un file di log.
CopyFile: crea una copia di un file esistente.

2.3 Persistenza

Il malware ottiene la persistenza, copiando il suo file eseguibile, nella cartella di avvio di Windows.

2.4 Analisi del codice

```
.text: 00401010        push eax
.text: 00401014        push ebx
.text: 00401018        push ecx
.text: 0040101C        push WH_Mouse          ; hook to Mouse
.text: 0040101F        call SetWindowsHook()
```

Questa parte di codice, tramite l'istruzione *push*, invia i registri sullo stack, che vengono passati come parametri della funzione *SetWindowsHooks()*, chiamata tramite l'istruzione *call*.

```
.text: 00401040        XOR ECX,ECX
```

Questa istruzione azzera il valore del registro ECX.

```
.text: 00401044    mov ecx, [EDI]    EDI = «path to
                                     startup_folder_system»
.text: 00401048    mov edx, [ESI]    ESI = path_to_Malware
```

Questa parte di codice, tramite l'istruzione *mov*, copia gli indirizzi delle cartelle sorgente e destinazione, nei registri.

```
.text: 0040104C    push ecx          ; destination folder
.text: 0040104F    push edx          ; file to be copied
.text: 00401054    call CopyFile();
```

Questa parte di codice, tramite l'istruzione *push*, invia i registri sullo stack, che vengono passati come parametri della funzione *CopyFile()*, chiamata tramite l'istruzione *call*.