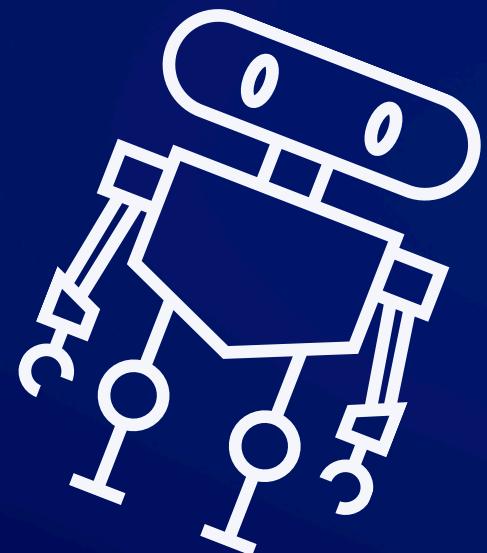
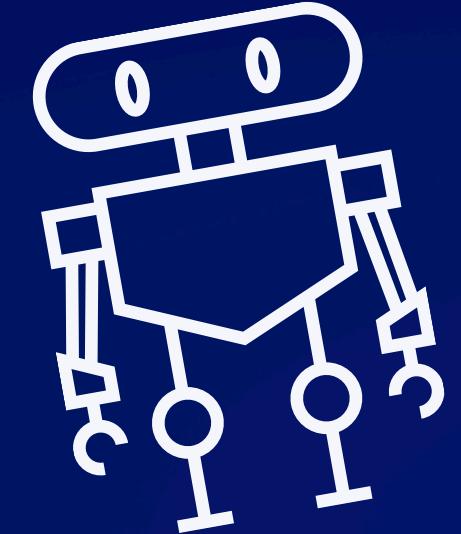




# Funzionalità dei malware



S11/L4



# 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();	

# PARTE 1

## TIPO DI MALWARE



**Quindi analizzando le chiamate possiamo affermare che si tratta del malware:  
KEYLOGGER CON PERSISTENZA**

Un keylogger è un tipo di malware progettato per registrare e monitorare tutte le sequenze di tasti digitati su una tastiera. Questo tipo di software può essere utilizzato per rubare informazioni sensibili, come password, numeri di carte di credito, messaggi, e altre informazioni personali o aziendali.

In particolare in questo caso ha l'obiettivo di registrare i movimenti sul mouse. Inoltre, il codice mostra un tentativo di copiare il malware in una cartella di avvio del sistema, indicando che il malware cerca di ottenere persistenza.

**Il malware analizzato è un keylogger con funzionalità di monitoraggio del mouse e persistenza sul sistema operativo**

## PARTE 2

### CHIAMATE DI FUNZIONE PRINCIPALI



Le chiamate di funzione utilizzate:

- **callSetWindoesHook()**

.text: 0040101F

call SetWindowsHook()

La chiamata di un hook di Windows nella programmazione in genere comporta l'uso dell'API di Windows per monitorare o intercettare eventi come input da tastiera o mouse, messaggi di finestra e così via. Gli hook sono utili per creare hotkey globali, intercettare messaggi inviati a una finestra e altro ancora.

- **call CopyFile()**

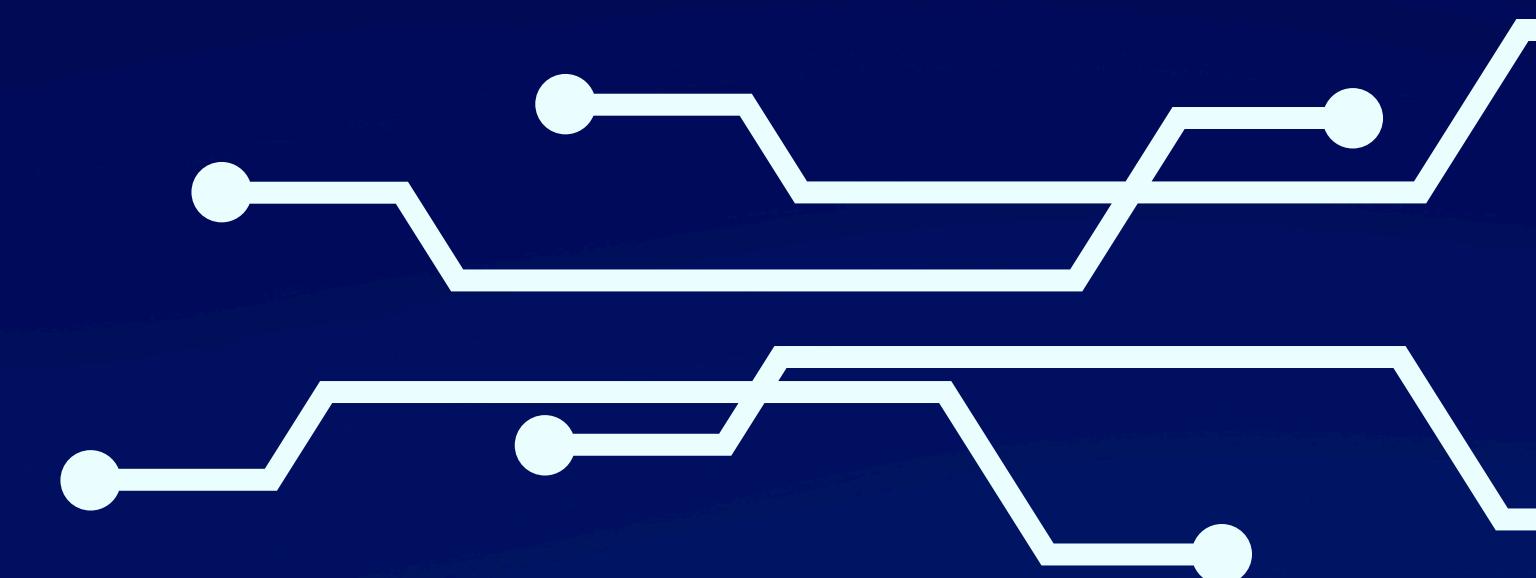
.text: 00401054

call CopyFile();

La funzione **CopyFile** nell'API di Windows è usata per copiare un file esistente in un nuovo file. Questa funzione è comunemente usata nelle applicazioni C++ o C che interagiscono con il sistema operativo Windows. La **CopyFile** funzione consente di creare un duplicato di un file, conservandone o modificandone gli attributi.

## PARTE 3

### METODO UTILIZZATO PER LA PERSISTENZA



Il malware tenta di ottenere persistenza copiando se stesso nella cartella di avvio del sistema (`startup_folder_system`).

Questo garantisce che il malware venga eseguito automaticamente ogni volta che il sistema operativo viene avviato.

La persistenza è ottenuta tramite la funzione `CopyFile`, che copia il malware dalla sua posizione originale a una cartella del sistema dove verrà eseguito all'avvio.

# BONUS

## ANALISI BASSO LIVELLO DELLE SINGOLE ISTRUZIONI

```
.text: 00401010  
.text: 00401014  
.text: 00401018
```

```
push eax  
push ebx  
push ecx
```

Queste istruzioni salvano i valori attuali dei registri eax, ebx, e ecx nello stack. Questo viene spesso fatto prima di chiamare una funzione per preservare lo stato dei registri.

```
.text: 0040101C
```

```
push WH_Mouse
```

```
; hook to Mouse
```

Questa istruzione push mette il valore associato a WH\_MOUSE nello stack, che indica che l'hook verrà impostato per monitorare gli eventi del mouse.

```
.text: 0040101F
```

```
call SetWindowsHook()
```

L'istruzione call esegue la funzione SetWindowsHookEx, impostando l'hook per monitorare i movimenti del mouse o i clic.

```
.text: 00401040
```

```
XOR ECX,ECX
```

L'istruzione XOR ECX, ECX azzera il registro ECX (effetto identico a MOV ECX, 0). Questa operazione viene spesso utilizzata per preparare il registro a essere usato come contatore o per cancellare il suo contenuto.

# BONUS

## ANALISI BASSO LIVELLO DELLE SINGOLE ISTRUZIONI

.text: 00401044                mov ecx, [EDI]

EDI = «path to  
startup\_folder\_system»

Questa istruzione carica nel registro ECX il valore memorizzato all'indirizzo puntato da EDI, che contiene il percorso della cartella di avvio del sistema.

.text: 00401048                mov edx, [ESI]

ESI = path\_to\_Malware

Simile alla precedente, questa istruzione carica nel registro EDX il valore memorizzato all'indirizzo puntato da ESI, che contiene il percorso del file malware.

.text: 0040104C                push ecx  
.text: 0040104F                push edx

; destination folder  
; file to be copied

Le istruzioni push ecx e push edx mettono i percorsi (cartella di destinazione e file del malware) nello stack come parametri per la successiva chiamata a CopyFile.

.text: 00401054                call CopyFile();

Questa istruzione chiama la funzione CopyFile, che copia il file del malware nella cartella di avvio, garantendo che venga eseguito automaticamente all'avvio del sistema.