

S11/L4

By Noemi de Martino

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 OGUNA 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

Traccia



.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. Tipo di malware

Basandosi sul codice fornito, il malware sembra essere un Trojan o “Rootkit” con capacità di persistenza. Questo è deducibile dall'uso di hook di sistema (**SetWindowsHook**), dalla copia di sé stesso in una cartella di avvio, e dal tentativo di garantirsi l'esecuzione automatica all'avvio del sistema.

2.Chiamate di Funzioni Principali

1 SetWindowsHook (0040101F):

Questa funzione imposta un hook di sistema globale per monitorare determinati eventi, in questo caso, probabilmente eventi del mouse (WH_Mouse).

Questo potrebbe essere utilizzato per monitorare o manipolare l'input del mouse, che potrebbe far parte del payload del malware o essere un modo per garantire l'esecuzione del codice in determinate azioni dell'utente.



.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.Chiamate di Funzioni Principali

2 CopyFile (00401054):

Questa funzione viene utilizzata per copiare il file del malware da una posizione (probabilmente il percorso di esecuzione corrente) a un'altra, come una cartella di avvio del sistema.

Questo è un metodo comune utilizzato dai malware per garantire la persistenza, copiandosi in una directory che viene eseguita automaticamente durante l'avvio del sistema.

.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();	



3. Metodo di persistenza utilizzato

Il malware utilizza la funzione CopyFile per copiare sé stesso nella cartella di avvio, garantendo così la propria esecuzione ogni volta che il sistema viene avviato.

Questo è un metodo di persistenza comune, poiché i file collocati in certe cartelle di avvio vengono eseguiti automaticamente all'avvio del sistema.

```
.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. Analisi a Basso livello delle Istruzioni

- .text: 00401010 push eax

Salva il contenuto del registro **eax** sullo stack.

- .text: 00401014 push ebx

Salva il contenuto del registro **ebx** sullo stack.

- .text: 00401018 push ecx

Salva il contenuto del registro **ecx** sullo stack.

- .text: 0040101C push WH_Mouse

Salva la costante **WH_Mouse** sullo stack, che indica che si sta impostando un hook del mouse

- .text: 0040101F call SetWindowsHook()

Chiama la funzione **SetWindowsHookEx** per impostare un hook del mouse

4. Analisi a Basso livello delle Istruzioni

-

.text: 00401040

XOR ECX,ECX

Imposta il registro **ecx** a zero.
Questo è un modo comune e volece di azzerare un registro

-

.text: 00401044

mov ecx, [EDI]

Carica il valore contenuto nell'indirizzo puntato da **EDI** nel
registro **ecx**.
EDI sembra puntare alla destinazione dove il malware vuole
copiare se stesso.

-

.text: 00401048

mov edx, [ESI]

Carica il valore contenuto nell'indirizzo puntato da **ESI** nel registro **edx**.
ESI sembra puntare al file del malware che deve essere copiato.

-

.text: 0040104C

push ecx

Salva il contenuto di **ecx** sullo stack, che contiene il percorso del file da
copiare.

4. Analisi a Basso livello delle Istruzioni

-

.text: 0040104F

push edx

-

.text: 00401054

call CopyFile();

Salva il contenuto di **edx** sullo stack, che contiene il percorso del file da copiare

-

Chiama la funzione **CopyFile** per copiare il file del malware dalla sorgente alla destinazione.

Il codice presentato è un esempio di malware progettato per ottenere persistenza all'interno di un sistema operativo.

Utilizza due funzioni chiave: **SetWindowsHookEx** e **CopyFile**.

La **prima** installa un hook di sistema per monitorare gli eventi del mouse, consentendo potenzialmente al malware di intercettare e manipolare gli input dell'utente.

La **seconda** funzione copia un file malevolo nella cartella di avvio del sistema, garantendo così che il malware venga eseguito automaticamente ad ogni riavvio del sistema.