





Funzionalità Dei Malware

- 
1. Il tipo di Malware in base alle chiamate di funzione utilizzate.
 2. Evidenziate le chiamate di funzione principaliaggiungendo 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 codice in esame, una chiamata di rilievo è quella alla funzione 'SetWindowsHook()'. Questo comando è utilizzato per creare un 'hook', una funzione che costantemente monitora una periferica specifica del sistema, come nel caso del mouse (individuato dal parametro WH_MOUSE). Questo potrebbe essere un indicatore della presenza di un keylogger.

La modalità con cui il malware mantiene la sua persistenza all'interno del sistema operativo è attraverso la 'Startup folder', come evidenziato nel comando 'mov ecx, [EDI]'. Ciò garantisce che il malware venga avviato automaticamente all'avvio del dispositivo, pronto a monitorare gli input provenienti dal mouse, sfruttando la cartella di avvio di Windows. I dati di input verranno successivamente registrati in un file di log.

| | | |
|-----------------|---------------------------|---------------------------------------|
| .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 | <u>XOR ECX,ECX</u> | |
| .text: 00401044 | mov ecx, [EDI] | EDI = «path to startup_folder_system» |
| .text: 00401048 | mov edx, [ESI] | <u>ESI = path_to_Malware</u> |
| .text: 0040104C | push ecx | ; destination folder |
| .text: 0040104F | push edx | ; file to be copied |
| .text: 00401054 | call CopyFile(); | |

- 
1. *push eax, push ebx, push ecx*: Queste istruzioni inseriscono i valori dei registri *eax*, *ebx* e *ecx* nello stack. Questo potrebbe essere parte della preparazione per la chiamata di funzione successiva, dove questi valori saranno probabilmente utilizzati come parametri.
 2. *push WH_Mouse*: Mette il valore della costante *WH_Mouse* nello stack. Questa costante probabilmente rappresenta il tipo di hook da impostare per l'intercettazione degli eventi del mouse.
 3. *call SetWindowsHook()*: Chiama la funzione *SetWindowsHook()* per impostare l'hook per l'intercettazione degli eventi del mouse. Questo consentirà al malware di monitorare e intercettare i movimenti del mouse.
 4. *XOR ECX, ECX, mov ecx, [EDI], mov edx, [ESI]*: Queste istruzioni eseguono operazioni di manipolazione dei registri *ecx*, *edi* e *esi*. Presumibilmente, stanno preparando i parametri necessari per la chiamata successiva alla funzione *CopyFile()*.
 5. *push ecx, push edx*: Mettono i valori di *ecx* e *edx* nello stack. Questi valori probabilmente contengono i percorsi di destinazione e di origine per la copia del file.
 6. *call CopyFile()*: Chiama la funzione *CopyFile()* per copiare il malware in una nuova posizione nel sistema operativo. Questo è un comportamento tipico di malware che cerca di propagarsi o nascondersi, copiando in diverse directory o rinominando.