Report sull'Analisi del Codice Malware

Data 15/02/24

Introduzione

L'estratto del codice del malware mostra una serie di chiamate di funzione e istruzioni che rivelano le sue caratteristiche e il suo comportamento. Attraverso un'analisi attenta, è possibile identificare il tipo di malware, le sue principali funzionalità e il metodo utilizzato per ottenere la persistenza sul sistema operativo.

Di seguito è riportato il blocco di codice assembly in esame :

.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();	

Analisi delle Chiamate di Funzione

Dalle chiamate di funzione utilizzate nell'estratto del codice, è possibile identificare il malware come un Keylogger. Le chiamate di funzione coinvolte indicano un'intenzione di sfruttare risorse del sistema e interagire con i driver di input, tipiche delle azioni di un Keylogger.

Chiamate di Funzione Principali e Descrizioni

Chiamata alla Funzione SetWindowsHook:

Questa chiamata di funzione è utilizzata dal programma in esame per installare un hook di Windows. Gli hook di Windows consentono al malware di intercettare e monitorare eventi di sistema specifici, come movimenti del mouse o pressioni di tasti. L'installazione di un hook potrebbe consentire al malware di raccogliere informazioni sulle attività dell'utente o di compromettere il normale funzionamento del sistema.

L'hook di windows richiamato è "WH_Mouse", questo ci fa subito intuire che il programma registra le azioni effettuate dall'utente con il mouse.

Chiamata alla Funzione CopyFile:

Questa chiamata di funzione è utilizzata dal programma per copiare un file da una posizione a un'altra. Analizzando il codice notiamo che il malware sfrutta questa funzionalità per propagarsi copiandosi su un altra cartella. Nel dettaglio in base alle indicazioni fornite dalla traccia possiamo confermare che il malware viene copiato nella cartella "startup_folder_system".

Metodo per Ottenere Persistenza sul Sistema Operativo

Il malware sembra utilizzare la funzione "CopyFile()" per accedere e modificare il registro di sistema. Questo suggerisce che il malware sta cercando di ottenere la persistenza sul sistema operativo registrando sé stesso all'interno del path corrispondente alla cartella "sturtup_folder_system" per essere eseguito all'avvio del sistema.

Analisi Basso Livello delle Singole Istruzioni

1. **.text: 00401010 push eax**: Questa istruzione mette il valore contenuto nel registro EAX nello stack. Questo potrebbe essere un parametro da passare a una funzione successiva.
2. **.text: 00401014 push ebx**: Questa istruzione mette il valore contenuto nel registro EBX nello stack. Anche questo potrebbe essere un parametro da passare a una funzione successiva.
3. **.text: 00401018 push ecx**: Questa istruzione mette il valore contenuto nel registro ECX nello stack. Ancora una volta, questo potrebbe essere un parametro da passare a una funzione successiva.

4. **.text: 0040101C push WH_Mouse**: Questa istruzione mette il valore **WH_Mouse** nello stack. Potrebbe essere un valore costante o una variabile utilizzata come parametro per una funzione successiva.
5. **.text: 0040101F call SetWindowsHook()**: Questa istruzione chiama una funzione denominata **SetWindowsHook()**, probabilmente passando i valori precedentemente messi nello stack come parametri. La funzione **SetWindowsHook()** sembra essere responsabile dell'impostazione di un hook per catturare gli eventi del mouse.
6. **.text: 00401040 XOR ECX,ECX**: Questa istruzione esegue un'operazione XOR sul registro ECX, impostandolo a zero.
7. **.text: 00401044 mov ecx, [EDI]**: Questa istruzione carica il valore memorizzato all'indirizzo di memoria contenuto nel registro EDI nel registro ECX. Presumibilmente, EDI contiene un puntatore al percorso della cartella di avvio del sistema.
8. **.text: 00401048 mov edx, [ESI]**: Questa istruzione carica il valore memorizzato all'indirizzo di memoria contenuto nel registro ESI nel registro EDX. Presumibilmente, ESI contiene un puntatore al percorso del malware.
9. **.text: 0040104C push ecx**: Questa istruzione mette il valore contenuto nel registro ECX (presumibilmente il percorso della cartella di avvio del sistema) nello stack. Potrebbe essere un parametro da passare a una funzione successiva.
10. **.text: 0040104F push edx**: Questa istruzione mette il valore contenuto nel registro EDX (presumibilmente il percorso del malware) nello stack. Anche questo potrebbe essere un parametro da passare a una funzione successiva.
11. **.text: 00401054 call CopyFile()**: Questa istruzione chiama una funzione denominata **CopyFile()**, probabilmente passando i valori precedentemente messi nello stack come parametri. La funzione **CopyFile()** sembra essere responsabile della copia di un file da una cartella di origine a una cartella di destinazione.