2024

EPICODE

CS0124

REPORT

Week 11 Lesson 4

PREPARED BY : Bruno Falconi

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

Figura 1:

```
.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. **Il tipo di Malware in base alle chiamate di funzione utilizzate.**

Data l'analisi del codice assembly fornitoci e delle sue funzionalità, insieme alla strategia di ottenere la persistenza copiandosi in una posizione specifica del sistema operativo, è possibile che questo malware sia un tipo di Trojan. Ecco perché:

Persistenza: I Trojan spesso cercano di ottenere la persistenza nel sistema copiandosi in posizioni specifiche, come le cartelle di avvio del sistema, in modo da essere eseguiti automaticamente all'avvio del sistema operativo.

Installazione di hook: Sebbene l'installazione di un hook del mouse non sia tipica dei Trojan, potrebbe essere utilizzata per scopi di monitoraggio o per intercettare input dell'utente. Tuttavia, non è la caratteristica principale di un Trojan.

Copiarsi in posizioni specifiche: La strategia di copiare se stesso in una posizione specifica del sistema operativo è comune nei Trojan, poiché consente loro di ottenere una persistenza duratura e di avviarsi automaticamente insieme al sistema.

2. **Evidenziate le chiamate di funzione principali aggiungendo una descrizione per ognuna di essa**

sembra che ci sia una sola chiamata di funzione principale nel codice assembly:

call SetWindowsHook(): Questa istruzione chiama la funzione SetWindowsHook(). Questa funzione è probabilmente una funzione di sistema o una funzione API di Windows utilizzata per installare un hook nel sistema operativo. Gli hook sono meccanismi utilizzati per intercettare eventi o messaggi nel sistema operativo, e possono essere utilizzati per vari scopi, come monitorare l'input dell'utente o modificare il comportamento del sistema. Nel contesto del malware, questa chiamata di funzione potrebbe essere utilizzata per installare un hook del mouse per intercettare e monitorare gli eventi del mouse.

È possibile che nel codice non fornito ci siano ulteriori chiamate di funzione, ad esempio per operazioni di copia del file o per altre azioni dannose.

3. **Il metodo utilizzato dal Malware per ottenere la persistenza sul sistema operativo**

Dal codice assembly e dalla relativa analisi, sembra che il malware utilizzi un approccio basato sulla copia di se stesso in una posizione specifica del sistema operativo per ottenere la persistenza. Ecco come potrebbe funzionare:

Installazione di un hook del mouse: La funzione SetWindowsHook() viene chiamata per installare un hook del mouse nel sistema operativo. Questo hook del mouse potrebbe essere utilizzato per intercettare eventi del mouse, ma non è direttamente correlato all'ottenimento della persistenza.

Preparazione dei percorsi: I percorsi dei file vengono caricati dai registri EDI ed ESI. Presumibilmente, EDI contiene il percorso della cartella di avvio del sistema e ESI contiene il percorso del malware.

Copia del malware: Anche se il codice fornito non mostra esplicitamente l'istruzione per la copia del malware, le istruzioni pushecx e pushedx mettono i percorsi della cartella di destinazione e del file da copiare nello stack. Questi valori potrebbero essere passati come parametri a una funzione successiva non mostrata nel codice, che effettua effettivamente la copia del malware nella cartella di avvio del sistema.

Quindi, il malware sembra ottenere la persistenza copiando se stesso nella cartella di avvio del sistema. Quando il sistema operativo viene avviato, il malware verrà automaticamente eseguito insieme all'avvio del sistema, consentendo al malware di mantenere la persistenza nel sistema operativo e di eseguire le sue azioni dannose all'avvio successivo del sistema.

4. **BONUS: Effettuare anche un'analisi basso livello delle singole istruzioni**

Di seguito un'analisi più dettagliata delle singole istruzioni del codice assembly fornito:

pusheax, pushebx, pushecx: Queste istruzioni mettono i contenuti dei registri EAX, EBX e ECX nello stack. Questo è comunemente fatto prima di chiamare una funzione per salvare i registri e passare eventuali parametri alla funzione.

pushWH_Mouse: Questa istruzione sembra essere un typo o un'abbreviazione per un valore relativo all'hook del mouse. Potrebbe essere un parametro per la funzione SetWindowsHook().

call SetWindowsHook(): Questa istruzione chiama la funzione SetWindowsHook(), che è probabilmente una funzione del sistema operativo Windows utilizzata per installare un hook del mouse o di un altro dispositivo di input.

XOR ECX, ECX: Questa istruzione esegue un'operazione XOR tra il registro ECX e se stesso, il che comporta l'azzeramento del registro ECX. Questo può essere utilizzato per impostare un registro a zero prima di utilizzarlo per un'operazione successiva.

movecx, [EDI]: Questa istruzione sposta il contenuto della memoria all'indirizzo contenuto nel registro EDI nel registro ECX. Presumibilmente, l'indirizzo contenuto in EDI punta al percorso della cartella di avvio del sistema.

movedx, [ESI]: Questa istruzione sposta il contenuto della memoria all'indirizzo contenuto nel registro ESI nel registro EDX. Presumibilmente, l'indirizzo contenuto in ESI punta al percorso del malware.

pushecx, pushedx: Queste istruzioni mettono i contenuti dei registri ECX ed EDX nello stack. Questi valori potrebbero essere passati come parametri a una funzione successiva, probabilmente una funzione che gestisce la copia del file.