Report di Analisi del Codice Malware S11L4

ush eax

.text: 00401014 push ebx

.text: 00401018 push ecx

.text: 0040101C push WH_Mouse

.text: 0040101F call SetWindowsHookEx()

.text: 00401040 XOR ECX,ECX

.text: 00401044 mov ecx, [EDI]

.text: 00401048 mov edx, [ESI]

.text: 0040104C push ecx

.text: 0040104F push edx

.text: 00401054 call CopyFile()

Descrizione Passo per Passo delle Istruzioni

1. push eax

- Descrizione: Questa istruzione spinge il valore contenuto nel registro `EAX` sullo stack.

- Scopo: Salvare il valore corrente di `EAX` sullo stack per preservare il contesto del registro prima di effettuare chiamate di funzione che potrebbero alterarlo.

2. push ebx

- Descrizione: Questa istruzione spinge il valore contenuto nel registro `EBX` sullo stack.

- Scopo: Salvare il valore corrente di `EBX` sullo stack per preservare il contesto del registro prima di effettuare chiamate di funzione che potrebbero alterarlo.

3. push ecx

- Descrizione: Questa istruzione spinge il valore contenuto nel registro `ECX` sullo stack.

- Scopo: Salvare il valore corrente di `ECX` sullo stack per preservare il contesto del registro prima di effettuare chiamate di funzione che potrebbero alterarlo.

4. push WH_Mouse

- Descrizione: Questa istruzione spinge il valore costante `WH_MOUSE` sullo stack. `WH_MOUSE` è una costante che indica che l'hook da installare è un hook per gli eventi del mouse.

- Scopo: Passare il tipo di hook come argomento alla funzione `SetWindowsHookEx`.

5. call SetWindowsHookEx()

- Descrizione: Questa istruzione chiama la funzione `SetWindowsHookEx`

- Scopo: Installare un hook per monitorare gli eventi del mouse. L'hook permette al malware di intercettare e potenzialmente registrare o manipolare gli input del mouse.

6. XOR ECX, ECX

- Descrizione: Questa istruzione esegue un'operazione XOR tra il registro `ECX` e se stesso, azzerandolo (ECX = 0).

- Scopo: Resettare il registro `ECX` a 0, probabilmente per prepararlo all'uso successivo.

7. mov ecx, [EDI]

- Descrizione: Questa istruzione muove il valore puntato da `EDI` nel registro `ECX`.

- Scopo: Caricare il percorso della cartella di avvio del sistema operativo nel registro `ECX`. `EDI` contiene l'indirizzo di memoria del percorso della cartella di avvio.

8. mov edx, [ESI]

- Descrizione: Questa istruzione muove il valore puntato da `ESI` nel registro `EDX`.

- Scopo: Caricare il percorso del file del malware nel registro `EDX`. `ESI` contiene l'indirizzo di memoria del percorso del malware.

9. push ecx

- Descrizione: Questa istruzione spinge il valore contenuto nel registro `ECX` sullo stack.

- Scopo: Passare il percorso della cartella di avvio come argomento alla funzione `CopyFile`.

10. push edx

- Descrizione: Questa istruzione spinge il valore contenuto nel registro `EDX` sullo stack.

- Scopo: Passare il percorso del file del malware come argomento alla funzione `CopyFile`.

11. call CopyFile()

- Descrizione: Questa istruzione chiama la funzione `CopyFile`.

- Scopo: Copiare il file del malware nella cartella di avvio del sistema operativo. Questo assicura che il malware venga eseguito automaticamente all'avvio del sistema, garantendo la sua persistenza. Il malware utilizza diverse funzioni di sistema per ottenere la persistenza e intercettare gli eventi del mouse. Installando un hook sugli eventi del mouse con `SetWindowsHookEx` e copiando se stesso nella cartella di avvio con `CopyFile`, il malware può rimanere persistente nel sistema e monitorare l'attività dell'utente.