

Analisi Dettagliata del Codice Malware



Traccia

In questo compito ci viene richiesto di analizzare un estratto di codice di un malware. Le richieste specifiche sono:

```
.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();
```

01

Identificare il tipo di malware in base alle chiamate di funzione utilizzate.

02

Evidenziare le chiamate di funzione principali aggiungendo una descrizione per ognuna di esse.

03

Spiegare il metodo utilizzato dal malware per ottenere la persistenza sul sistema operativo.

04

BONUS: Effettuare un'analisi a basso livello delle singole istruzioni.

1 Quesito

Identificazione del Tipo di Malware

Il codice mostra due chiamate di funzione principali:

- **SetWindowsHook()**
- **CopyFile()**

Queste chiamate indicano che il malware può essere classificato come un keylogger/spyware e come un malware che utilizza un meccanismo di persistenza.

- **Keylogger/Spyware:** La funzione **SetWindowsHook()** viene utilizzata per intercettare eventi del mouse, il che è tipico dei malware che mirano a raccogliere informazioni sensibili, come i movimenti del mouse o i clic, potenzialmente per tracciare l'attività dell'utente.
- **Persistenza:** La funzione **CopyFile()** viene utilizzata per copiare l'eseguibile del malware in una cartella di avvio, garantendo che il malware venga eseguito ogni volta che il sistema operativo si avvia.



1. SetWindowsHook()

- **Descrizione:** La funzione `SetWindowsHook()` installa una funzione hook che può monitorare vari tipi di eventi di sistema. Nel contesto del codice, viene utilizzato il parametro `WH_Mouse`, che indica un hook per gli eventi del mouse.
- **Utilizzo:** Questo permette al malware di intercettare i movimenti e i clic del mouse, raccogliendo informazioni sensibili o osservando il comportamento dell'utente.

2. CopyFile()

- **Descrizione:** La funzione `CopyFile()` copia un file da una sorgente a una destinazione specificata. Nel contesto del malware, questa funzione viene utilizzata per copiare l'eseguibile del malware stesso in una cartella di avvio del sistema.
- **Utilizzo:** La destinazione è una cartella di avvio (`startup_folder_system`), che garantisce che il malware venga eseguito automaticamente ogni volta che il sistema si avvia.



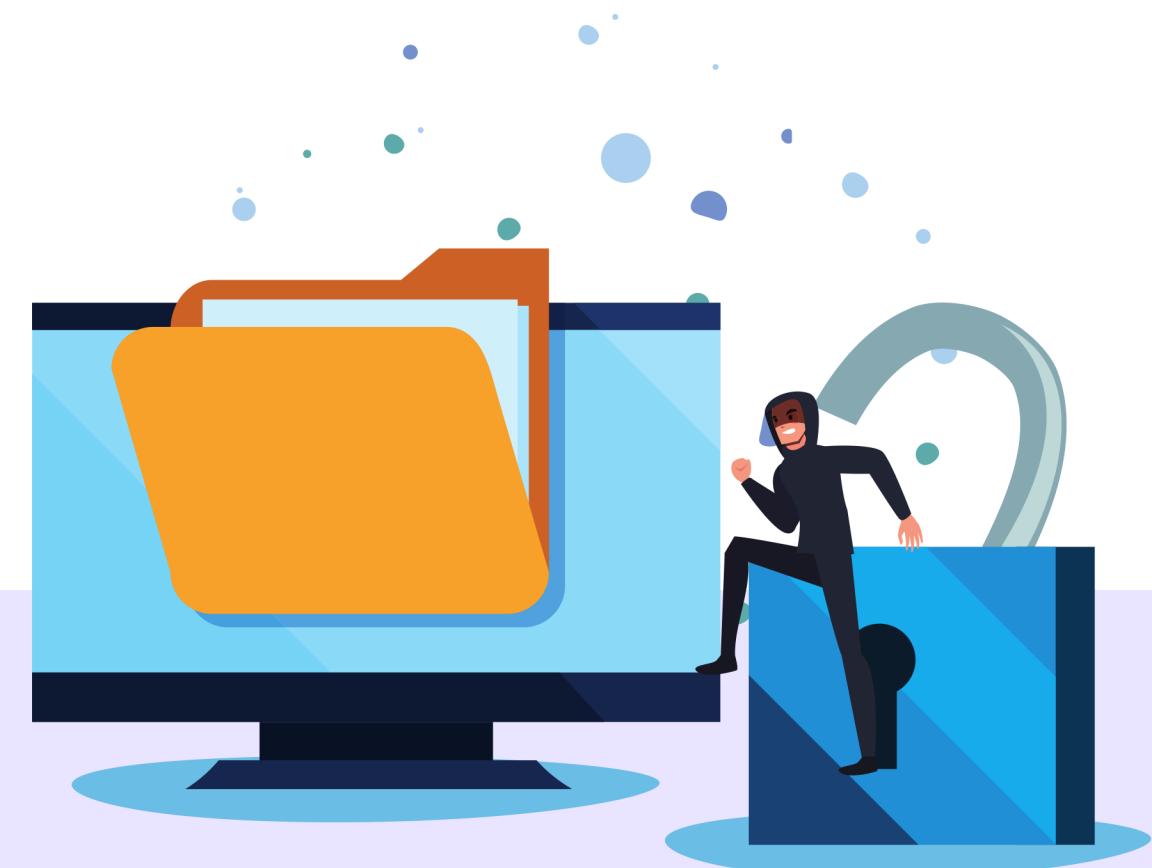
3 Quesito

Metodo di Persistenza

Il malware utilizza la funzione **CopyFile()** per ottenere la persistenza sul sistema operativo. Il metodo seguito è il seguente:

1. Identificazione della Cartella di Destinazione: Utilizza il registro EDI per ottenere il percorso della cartella di startup del sistema.

2. Copia del Malware: Copia il proprio file eseguibile nella cartella di startup utilizzando `CopyFile()`. Questo garantisce che il malware venga eseguito automaticamente ogni volta che il sistema si avvia.



Quesito Bonus PT1

Analisi a Basso Livello delle Istruzioni

Eseguiamo un'analisi dettagliata delle istruzioni assembler riportate:



1. PUSH EAX, EBX, ECX:

- **Descrizione:** Le istruzioni PUSH salvano i registri EAX, EBX, ECX nello stack. Questo è spesso fatto per preservare lo stato di questi registri prima di chiamare una funzione che potrebbe modificarli.

2.PUSH WH_Mouse:

- **Descrizione:** Pusha un valore costante (che rappresenta il tipo di hook, in questo caso per il mouse) nello stack come parametro per SetWindowsHook().

3. CALL SetWindowsHook():

- **Descrizione:** Chiama la funzione SetWindowsHook() con il parametro precedentemente pushato (WH_Mouse). Questa funzione installa un hook che monitora gli eventi del mouse.

4.XOR ECX, ECX:

- **Descrizione:** Azzera il registro ECX impostandolo a 0. Questa è una tecnica comune per reimpostare i registri.

5.MOV ECX, [EDI]:

- **Descrizione:** Carica il valore memorizzato all'indirizzo puntato da EDI nel registro ECX. EDI contiene il percorso della cartella di startup.

Quesito Bonus PT2



6. MOV EDX, [ESI]:

- **Descrizione:** Carica il valore memorizzato all'indirizzo puntato da ESI nel registro EDX. ESI contiene il percorso del file del malware da copiare.

7. PUSH ECX:

- **Descrizione:** Pusha il percorso della cartella di startup (contenuto in ECX) nello stack come parametro per CopyFile().

8. PUSH EDX:

- **Descrizione:** Pusha il percorso del file del malware (contenuto in EDX) nello stack come parametro per CopyFile().

9. CALL CopyFile():

- **Descrizione:** Chiama la funzione CopyFile() per copiare il file del malware nella cartella di startup. Questa operazione garantisce che il malware venga eseguito automaticamente all'avvio del sistema operativo.

Conclusione

L'analisi del codice mostra chiaramente le intenzioni del malware di intercettare eventi del mouse per potenzialmente raccogliere informazioni sensibili e garantire la propria persistenza copiandosi in una cartella di avvio. Questo malware utilizza tecniche comuni di keylogging e persistente meccanismo, che sono elementi chiave per mantenere la sua presenza e funzionamento nel sistema infetto. Queste tecniche non solo permettono al malware di eseguire le sue funzioni malevole, ma anche di sopravvivere ai riavvii del sistema, rendendo più difficile la sua rimozione senza un'analisi e una pulizia approfondite.

