S11-L1

Windows Malware

Emanuele Di Stefano

Analisi del Malware basata sui Tracce Assembly Fornite

Di seguito ho riportato un'analisi del codice assembly tratto da un malware reale, rispondendo alle domande chiave relative alla persistenza del malware, alla connessione Internet e al significato del comando assembly "lea".

1. Persistenza del Malware

Il malware ottiene la persistenza tramite la manipolazione del registro di sistema di Windows. Questo viene realizzato aggiungendo una chiave di registro nella sezione "Run" del registro, che assicura l'esecuzione automatica del malware all'avvio del sistema.

Codice Rilevante:

- push offset SubKey imposta la chiave del registro di destinazione ("Software\Microsoft\Windows\CurrentVersion\Run").
- push HKEY_LOCAL_MACHINE specifica il contesto del registro.
- call esi ; RegOpenKeyExW apre la chiave di registro specificata.
- call ds:RegSetValueExW scrive un valore nella chiave del registro, impostando il programma da eseguire automaticamente all'avvio.
- Commento: Il codice mostra come il malware si assicuri che venga eseguito ad ogni avvio del sistema modificando il registro di Windows.

2. Identificazione del Client Software per la Connessione a Internet

Il malware utilizza Internet Explorer 8.0 per stabilire la connessione a Internet. Questo è indicato dalla stringa offset szAgent seguita dalla stringa "Internet Explorer 8.0", passata alla funzione InternetOpen.

Codice Rilevante:

- push offset szAgent ; "Internet Explorer 8.0"
- call ds:InternetOpenA utilizza l'agente specificato (Internet Explorer 8.0) per aprire una sessione Internet.

3. Identificazione dell'URL e della Funzione di Connessione

Il malware tenta di connettersi all'URL http://www.malwareiz.com

Codice Rilevante:

- push offset szUrl ; "http://www.malwareiz.com"
- call edi ; InternetOpenUrlA è la chiamata di funzione che permette al malware di connettersi all'URL specificato.

4. Significato e Funzionamento del Comando Assembly "lea"

Il comando "lea" (Load Effective Address) è utilizzato in assembly per calcolare l'indirizzo effettivo di un'operazione di memoria e caricarlo in un registro, senza eseguire l'effettiva operazione di accesso alla memoria. È spesso utilizzato per ottenere puntatori o per fare semplici operazioni aritmetiche efficientemente.

Esempio nel Codice:

- lea ecx, [esp+424h+Data] carica l'indirizzo effettivo di Data (un'offset) nel registro ecx, utilizzato successivamente per operazioni.