S11/L1

INTRO E CONCETTI DI WINDOWS AVANZATI

Flavio Scognamiglio

Traccia

Con riferimento agli estratti di un malware reale presenti nelle prossime slide, rispondere alle seguenti domande:

- Descrivere come il malware ottiene la persistenza , evidenziando il codice assembly dove le relative istruzioni e chiamate di funzioni vengono eseguite.

- Identificare il client software utilizzato dal malware per la connessione ad Internet.

- Identificare l'URL al quale il malware tenta di connettersi ed evidenziare la chiamata di funzione che permette al malware di connettersi ad un URL.

- BONUS: qual è il significato e il funzionamento del comando assembly "lea".

Traccia

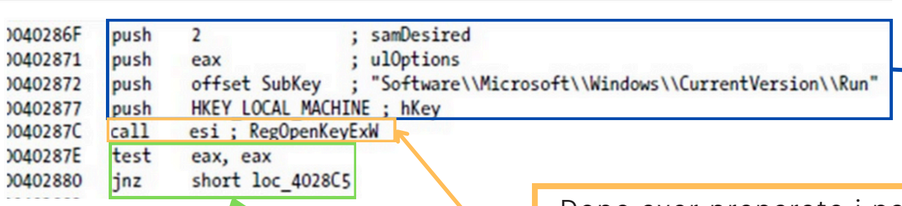
```
)040286F   push    2               ; samDesired
)0402871   push    eax             ; ulOptions
)0402872   push    offset SubKey   ; "Software\\Microsoft\\Windows\\CurrentVersion\\Run"
)0402877   push    HKEY_LOCAL_MACHINE ; hKey
)040287C   call    esi ; RegOpenKeyExW
)040287E   test    eax, eax
)0402880   jnz     short loc_4028C5
)0402882
)0402882 loc_402882:
)0402882   lea     ecx, [esp+424h+Data]
)0402886   push    ecx             ; lpString
)0402887   mov     bl, 1
)0402889   call    ds:lstrlenW
)040288F   lea     edx, [eax+eax+2]
)0402893   push    edx             ; cbData
)0402894   mov     edx, [esp+428h+hKey]
)0402898   lea     eax, [esp+428h+Data]
)040289C   push    eax             ; lpData
)040289D   push    1               ; dwType
)040289F   push    0               ; Reserved
)04028A1   lea     ecx, [esp+434h+ValueName]
)04028A8   push    ecx             ; lpValueName
)04028A9   push    edx             ; hKey
)04028AA   call    ds:RegSetValueExW
```

```
.text:00401150 ; ||||||||||||||||| S U B R O U T I N E |||||||||||||||||||||||||||||||||||||||
.text:00401150
.text:00401150
.text:00401150 ; DWORD __stdcall StartAddress(LPVOID)
.text:00401150 StartAddress    proc near               ; DATA XREF: sub_401040+EC↑o
.text:00401150                 push    esi
.text:00401151                 push    edi
.text:00401152                 push    0               ; dwFlags
.text:00401154                 push    0               ; lpszProxyBypass
.text:00401156                 push    0               ; lpszProxy
.text:00401158                 push    1               ; dwAccessType
.text:0040115A                 push    offset szAgent  ; "Internet Explorer 8.0"
.text:0040115F                 call    ds:InternetOpenA
.text:00401165                 mov     edi, ds:InternetOpenUrlA
.text:0040116B                 mov     esi, eax
.text:0040116D
.text:0040116D loc_40116D:                             ; CODE XREF: StartAddress+30↓j
.text:0040116D                 push    0               ; dwContext
.text:0040116F                 push    80000000h       ; dwFlags
.text:00401174                 push    0               ; dwHeadersLength
.text:00401176                 push    0               ; lpszHeaders
.text:00401178                 push    offset szUrl    ; "http://www.malware12COM
.text:0040117D                 push    esi             ; hInternet
.text:0040117E                 call    edi ; InternetOpenUrlA
.text:00401180                 jmp     short loc_40116D
.text:00401180 StartAddress    endp
.text:00401180
```

Persistenza del malware

Il malware in questione **ottiene la <u>persistenza</u>** attraverso l'apertura (funzione RegOpenKey) e la modifica (RegSetValue) di una chiave all'interno del registro utilizzata da Windows, <u>la quale contiene i programmi che si eseguiranno all'avvio del sistema:</u>
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run

Sia la prima immagine che le relative istruzioni, li ho grossomodo già visti nella lezione teorica. Rivediamo nel dettaglio:



In queste istruzioni il malware prepara **diversi valori** per essere utilizzati nelle chiamate di funzione, spingendoli sullo stack tramite l'istruzione **push**. Ad esempio, il valore 2 viene utilizzato per indicare il livello di accesso desiderato, identificato come **samDesired**. Il malware vuole poter scrivere.

Poi viene passato un valore dal registro eax, che in questo contesto rappresenta l'assenza di opzioni speciali (ulOptions = 0). Il malware utilizza quindi l'offset che punta alla stringa:
"Software\\Microsoft\\Windows\\CurrentVersion\\Run", che viene spinto sullo stack per indicare la **sottochiave di registro che intende aprire**.

Viene poi specificato il contesto in cui operare, utilizzando **HKEY_LOCAL_MACHINE**, una delle principali **root key** del registro di Windows, viste anche nella teoria.

Dopo aver preparato i parametri, c'è la chiamata alla funzione **RegOpenKey**, che permette l'apertura della chiave al fine poi di modificarla. Un po' come aprire un libro per poterlo leggere.

Con test "verifica" il contenuto del registro eax, che contiene il risultato della chiamata. **Se la funzione RegOpenKey ha avuto successo, eax sarà pari a 0**, altrimenti il flusso di esecuzione verrà **deviato** (jnz) per gestire l'errore.

Persistenza del malware

Per impostare il valore nella chiave di registro di cui ho parlato sopra, e garantire quindi la persistenza, il malware utilizza la funzione **RegSetValueExW.** Prima ovviamente, fa una serie di operazioni per preparare i parametri.

Per comodità inserisco tutte le istruzioni in un unico riquadro, altrimenti i singoli riquadri occuperebbero troppo spazio.

```
00402882  lea     ecx, [esp+424h+Data]
00402886  push    ecx             ; lpString
00402887  mov     bl, 1
00402889  call    ds:lstrlenW
0040288F  lea     edx, [eax+eax+2]
00402893  push    edx             ; cbData
00402894  mov     edx, [esp+428h+hKey]
00402898  lea     eax, [esp+428h+Data]
0040289C  push    eax             ; lpData
0040289D  push    1               ; dwType
0040289F  push    0               ; Reserved
004028A1  lea     ecx, [esp+434h+ValueName]
004028A8  push    ecx             ; lpValueName
004028A9  push    edx             ; hKey
004028AA  call    ds:RegSetValueExW
```

- lea ecx, [esp+424h+Data]: Carica l'indirizzo della stringa Data.
- push ecx: Prepara la stringa Data per l'inserimento nel registro.
- call ds lstrlenW: Calcola la lunghezza della stringa Data.
- lea edx, [eax+eax+2]: Calcola la dimensione dei dati.
- push edx: Imposta la dimensione dei dati da scrivere.
- lea eax, [esp+428h+hKey]: Carica l'handle della chiave di registro.
- push eax: Prepara l'handle per la chiamata.
- push 1: Specifica il tipo di dati (REG_SZ).
- push 0: Riservato, sempre 0.
- lea ecx, [esp+434h+ValueName]: Carica il nome del valore.
- push ecx: Prepara il nome del valore per la scrittura.
- push edx: Passa l'handle della chiave di registro.
- call ds RegSetValueExW: **Crea o modifica** il valore nel registro per ottenere persistenza.

CLIENT CONNESSIONE AD INTERNET E URL

Il client utilizzato dal malware per la connessione a Internet è **Internet Explorer**. Questa parte l'ho presa nella seconda immagine fornita dalla traccia.

Il parametro **szAgent** passato alla funzione **InternetOpenA** è "Internet Explorer 8.0". Questo viene utilizzato per identificarsi come un'istanza di Internet Explorer.

```
.text:00401150 ; ||||||||||||||| S U B R O U T I N E |||||||||||||||||||||||||||||||||||||||
.text:00401150
.text:00401150
.text:00401150 ; DWORD __stdcall StartAddress(LPVOID)
.text:00401150 StartAddress    proc near               ; DATA XREF: sub_401040+EC↑o
.text:00401150                 push    esi
.text:00401151                 push    edi
.text:00401152                 push    0               ; dwFlags
.text:00401154                 push    0               ; lpszProxyBypass
.text:00401156                 push    0               ; lpszProxy
.text:00401158                 push    1               ; dwAccessType
.text:0040115A                 push    offset szAgent  ; "Internet Explorer 8.0"
.text:0040115F                 call    ds:InternetOpenA
.text:00401165                 mov     edi, ds:InternetOpenUrlA
.text:0040116B                 mov     esi, eax
.text:0040116D
.text:0040116D loc_40116D:                             ; CODE XREF: StartAddress+30↓j
.text:0040116D                 push    0               ; dwContext
.text:0040116F                 push    80000000h       ; dwFlags
.text:00401174                 push    0               ; dwHeadersLength
.text:00401176                 push    0               ; lpszHeaders
.text:00401178                 push    offset szUrl    ; "http://www.malware12COM
.text:0040117D                 push    esi             ; hInternet
.text:0040117E                 call    edi ; InternetOpenUrlA
.text:00401180                 jmp     short loc_40116D
.text:00401180 StartAddress    endp
.text:00401180
```

Il malware utilizza la funzione **InternetOpenUrlA** per connettersi a questo URL: **http://www.malware12.com**

SIGNIFICATO DEL COMANDO LEA

Questo codice utilizza l'istruzione **LEA**, che non abbiamo ancora studiato. LEA calcola e carica l'indirizzo di memoria di una variabile in un registro **senza accedere alla memoria**. In questo codice, viene usata per calcolare rapidamente gli indirizzi dei dati e dei nomi dei valori da scrivere nel registro di Windows, ottimizzando il processo e riducendo il numero di istruzioni necessarie per manipolare gli indirizzi. Questo rende il malware più efficiente nel preparare i parametri per la persistenza.

Senza LEA si sarebbero dovute effettuare una serie di operazioni in più con **MOV** e **ADD**.

GRAZIE

FLAVIO SCOGNAMIGLIO