CS0424IT — ESERCITAZIONE S11L1
ANALISI MALWARE WINDOWS

Simone La Porta

GitHub

---

19 agosto 2024

INDICE

1 TRACCIA

Con riferimento agli estratti di un malware reale successivi, 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".

```
00040286F   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, [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
```

```
.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
```

2 SVOLGIMENTO

2.1 *Come il malware ottiene la persistenza*

Il malware ottiene la persistenza modificando il registro di Windows per assicurarsi che il codice venga eseguito ogni volta che il sistema viene avviato. Nella prima immagine di codice assembly, il malware utilizza la chiamata alla funzione `RegSetValueExW` per scrivere nel registro di Windows. La chiave di registro interessata è:

`HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run`,

una posizione comune utilizzata per configurare i programmi che si avviano automaticamente all'accensione del computer.

Più nel dettaglio:

- **`push`**: Il comando push inserisce valori nello stack, che saranno utilizzati come parametri per le chiamate di funzione.
- **`lea ecx, [esp+424h+Data]`**: Questo comando `lea` (Load Effective Address) carica l'indirizzo effettivo di `Data` nel registro `ecx`. Questo è un esempio del comando `lea`, che è fondamentale per calcolare indirizzi senza caricare direttamente i dati.
- **`call ds:RegSetValueExW`**: Questa funzione è cruciale per garantire la persistenza. Inserisce il valore specificato nel registro di Windows, assicurando che il malware venga eseguito all'avvio del sistema.

```
0040286F push 2                 ; samDesired (specifica i diritti di accesso)
00402871 push eax               ; ulOptions (opzioni specifiche per l'apertura della
    chiave)
00402872 push offset SubKey     ; "Software\\Microsoft\\Windows\\CurrentVersion\\Run
   " (chiave di registro da modificare)
00402877 push HKEY_LOCAL_MACHINE ; hKey (il contesto della chiave del registro)
0040287C call esi ; RegOpenKeyExW (apre la chiave di registro specificata)
00402881 test eax, eax          ; Verifica se l'apertura della chiave ha avuto
   successo
00402883 jnz short loc_4028C5   ; Se non ha successo, salta alla locazione di errore
00402882 loc_402882:
00402886 lea ecx, [esp+424h+Data] ; lpString (carica l'indirizzo della stringa in
   ecx)
```

```
0040288B mov ecx, eax          ; Memorizza il valore dell'handle della chiave di
    registro
0040288D call ds:lstrlenW      ; Calcola la lunghezza della stringa
0040288F lea edx, [eax+eax+2]  ; Prepara i dati per l'inserimento nel registro
00402894 mov edx, [esp+428h+hKey] ; Carica l'handle della chiave di registro in edx
00402899 lea eax, [esp+428h+Data] ; Carica l'indirizzo dei dati da scrivere
0040289C push eax              ; lpData (dati da scrivere nel registro)
0040289D push 1                ; dwType (tipo di dati: REG_SZ)
0040289F push 0                ; Reserved (riservato, impostato a 0)
004028A8 push ecx              ; hKey (handle della chiave di registro)
004028AA call ds:RegSetValueExW ; Scrive i dati nel registro per garantire la
    persistenza
```

2.2 *Il client software utilizzato dal malware per la connessione ad Internet*

Il malware utilizza Internet Explorer come client per connettersi a Internet. Questo è evidenziato nella seconda immagine dal codice che utilizza la funzione `InternetOpenA` con la stringa "Internet Explorer 8.0", che specifica l'agente utente per le connessioni HTTP.

In particolare:

- **`push offset szAgent`**: Questo comando inserisce l'indirizzo della stringa "Internet Explorer 8.0" nello stack, definendo così l'user-agent utilizzato dal malware.
- **`call ds:InternetOpenA`**: Inizializza una connessione Internet, utilizzando l'user-agent specificato. Questo permette al malware di operare in modo simile a un normale browser.
- **`call ds:InternetOpenUrlA`**: Questa chiamata di funzione apre una connessione all'URL specificato. In questo caso, si tratta dell'URL maligno `http://www.malware12.COM`.

```
.text:00401150 push offset szAgent ; "Internet Explorer 8.0" (specifica l'user-
    agent)
.text:00401155 call ds:InternetOpenA ; Inizializza una connessione Internet
.text:0040115A mov edi, ds:InternetOpenUrlA ; Prepara la funzione per aprire l'URL
.text:0040115F push offset szUrl   ; "http://www.malware12.COM" (URL di
    destinazione)
.text:00401167 call edi            ; Esegue la connessione all'URL specificato
```

2.3 *URL al quale il malware tenta di connettersi*

L'URL a cui il malware tenta di connettersi è `http://www.malware12.com`. Questo è visibile nella seconda immagine di codice assembly dove il malware utilizza la funzione `InternetOpenUrlA`, passando l'URL come argomento.

2.4 *Funzionamento del comando assembly "lea"*

Il comando assembly `lea` (Load Effective Address) è utilizzato per caricare l'indirizzo effettivo di una locazione di memoria nel registro. Non carica il valore della memoria, ma calcola l'indirizzo effettivo basato su un'espressione e lo memorizza nel registro. Questo è utile per calcoli di indirizzi in modo efficiente senza usare risorse extra per il carico dei dati. Ad esempio, `lea eax, [ebx+ecx*2]` calcola l'indirizzo risultante da `ebx+ecx*2` e lo carica in `eax`.