

Relazione S11L1

L'obiettivo è quello di analizzare del codice assembly (riportato qui di seguito) e dare le seguenti informazioni:

```
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, [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+EC70
.text:00401150         push    esi
.text:00401151         push    edi
.text:00401152         push    0                ; dwFlags
.text:00401154         push    0                ; lpzProxyBypass
.text:00401156         push    0                ; lpzProxy
.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:00401168         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                ; lpzHeaders
.text:00401178         push    offset szUrl      ; "http://www.malware12.COM
.text:0040117D         push    esi              ; hInternet
.text:0040117E         call    edi ; InternetOpenUrlA
.text:00401180         jmp     short loc_40116D
.text:00401180 StartAddress  endp
.text:00401180
-----
.text:00401180 - -----
```

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

Per quanto riguarda il primo punto possiamo vedere che il malware ottiene la persistenza chiamando le seguenti funzioni:

```
00402872  push    offset SubKey    ; "Software\\Microsoft\\Windows\\CurrentVersion\\Run"
00402877  push    HKEY_LOCAL_MACHINE ; hKey
0040287C  call    esi ; RegOpenKeyExW

004028A8  push    ecx                ; lpValueName
004028A9  push    edx                ; hKey
004028AA  call    ds:RegSetValueExW
```

In pratica il malware prima apre la chiave del registro e dopo ne setta il valore. In questo modo il malware potrebbe avviarsi ogni volta che si avvia il pc.

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

```
.text:0040115A  push    offset szAgent    ; "Internet Explorer 8.0"
.text:0040115F  call    ds:InternetOpenA
```

Come si può vedere da questa immagine, il client software è internet explorer.

3. 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

```
.text:00401178  push    offset szUrl      ; "http://www.malware12COM
.text:0040117D  push    esi                ; hInternet
.text:0040117E  call    edi ; InternetOpenUrlA
```

In queste tre righe si vede come viene inizializzato il parametro dell'url e passato alla funzione **InternetOpenUrl**, in questo modo verrà utilizzato l'handler che gestisce la connessione **hInternet** e verrà aperto l'url.

4. Bonus: qual è il significato e il funzionamento del comando assembly “lea”

Lea sta per Load Effective Address, e consiste nel caricare un indirizzo di memoria in un registro. In pratica quindi non viene caricato l'effettivo valore che risiede a quell'indirizzo ma solamente l'indirizzo stesso.

Davide Lecci