

Traccia:

```
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
```

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

Per ottenere la persistenza, un malware modifica le chiavi di registro inerenti l'avvio del sistema operativo in modo tale che il sistema stesso lo avvii nelle fasi di start-up; per fare cio' utilizza la funzione **RegOpenKeyEx** per aprire la chiave di registro che vuole modificare (riga 0040287C dell'immagine) e la funzione **RegSetValueExW** per modificare il valore salvato nella chiave che e' stata aperta (riga 004028AA dell'immagine). I parametri delle funzioni vengono inseriti prima di chiamare la funzione e sono le righe con l'azione PUSH che precedono la riga con l'azione CALL.

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

Il comando lea significa **Load Effective Address** e carica sulla destinazione l'indirizzo effettivo di una certa variabile specificato nella sorgente; a differenza di mov che carica sulla destinazione il valore contenuto nell'indirizzo specificato nella sorgente;

```

.text:00401150 ; :::::::::::::::::::: S U B R O U T I N E ::::::::::::::::::::
.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: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

```

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

Il software utilizzato e' **Internet Explorer versione 8.0** (quadrato rosso dell'immagine) e viene inserito come parametro **szAgent** della funzione **InternetOpenA** che serve a stabilire una connessione 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

L'url e' <http://www.malware12.COM>(quadrato azzurro dell'immagine) e viene inserito come parametro **szUrl** della funzione **InternetOpenUrlA**(quadrato arancione dell'immagine) che appunto consente di stabilire una connessione con un url specificato.