



# Compito S11-L1



# Traccia

Con riferimento agli estratti di un malware reale presenti nelle prossime slide, rispondere alle seguenti domande:

- 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 e il funzionamento del comando assembly "lea".



# 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:lstlenW
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
```

# Traccia

```
.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 ; 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
.text:0040116D loc_40116D: ; CODE XREF: StartAddress+304j
.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
.text:00401180
.text:00401180
```



## Esercizio: Punto 1

Il malware ottiene la persistenza inserendo un nuovo valore all'interno della chiave di registro "**Software\\Microsoft\\Windows\\CurrentVersion\\Run**", che include tutti i programmi avviati durante l'avvio del sistema operativo. Le funzioni utilizzate sono:

1. **RegOpenKey**, che consente di aprire la chiave selezionata. I parametri sono passati sullo stack tramite le istruzioni "push" che precedono la chiamata della funzione.
2. **RegSetValueEx**, che consente al malware di inserire un nuovo valore all'interno della chiave di registro appena aperta.

## Esercizio: Punto 2

Il client utilizzato dal malware per connettersi ad internet è **Internet Explorer**.

```
.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 ; 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
.text:0040116D loc_40116D: ; CODE XREF: StartAddress+30j
.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.malware12COM
.text:0040117D push esi ; hInternet
.text:0040117E call edi ; InternetOpenUrlA
.text:00401180 jmp short loc_40116D
.text:00401180 StartAddress endp
.text:00401180
.text:00401180
```

## Esercizio: Punto 3

Il malware tenta di stabilire una connessione all'URL `www.malware12.com`. Per effettuare questa connessione, utilizza la funzione `InternetOpenURL`. L'URL viene passato come parametro a tale funzione mediante l'istruzione "push" nello stack.

```
.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+EC↑o
.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: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 ; 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
.text:00401180
.text:00401180
```



## Esercizio: Bonus

Farò un esempio per spiegarlo. L'istruzione assembly `lea eax, [esp+428h+Data]` utilizza “lea” (Load Effective Address) per calcolare l'indirizzo effettivo. Qui, sommiamo il valore corrente dello stack pointer (esp) con 428 in esadecimale (428h) e il valore di Data. L'indirizzo risultante viene quindi caricato nel registro eax.

In sintesi il comando viene utilizzato per caricare l'indirizzo di un operando anziché il suo valore. È spesso usato per eseguire operazioni di aritmetica sugli indirizzi.