



Cybersecurity Specialist

# Cyber Security & Ethical Hacking

## GIORNO 1 – MALWARE ANALYSIS

---

Alejandro Cristino  
S11-L1

# TRACCIA

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

Esercizio Windows malware

1. Descrivere come il malware ottiene la persistenza , evidenziando il codice assembly dove le relative istruzioni e chiamate di funzioni vengono eseguite
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
4. BONUS: qual è il significato e il funzionamento del comando assembly "lea"

```
040286F push    2          ; samDesired
0402871 push    eax        ; ulOptions
0402872 push    offset SubKey ; "Software\\Microsoft\\Windows\\CurrentVersion\\Run"
0402877 push    HKEY_LOCAL_MACHINE ; hKey
040287C call    esi ; RegOpenKeyExW
040287E test    eax, eax
0402880 jnz     short loc_4028C5
0402882 loc_402882:
0402882 lea     ecx, [esp+424h+Data]
0402886 push   ecx        ; lpString
0402887 mov     bl, 1
0402889 call   ds:lstrlenW
040288F lea     edx, [eax+eax+2]
0402893 push   edx        ; cbData
0402894 mov     edx, [esp+428h+hKey]
0402898 lea     eax, [esp+428h+Data]
040289C push   eax        ; lpData
040289D push   1          ; dwType
040289F push   0          ; Reserved
04028A1 lea     ecx, [esp+434h+ValueName]
04028A8 push   ecx        ; lpValueName
04028A9 push   edx        ; hKey
04028AA call   ds:RegSetValueExW
```

```
.text:00401150 ; ||||||| S U B R O U T I N E |||||||
.text:00401150
.text:00401150
.text:00401150 ; DWORD __stdcall StartAddress(LPUOID)
.text:00401150 StartAddress proc near ; DATA XREF: sub_401040+ECTo
.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 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.malware12COM"
.text:0040117D             push    esi        ; hInternet
.text:0040117E             call    edi ; InternetOpenUrlA
.text:00401180             jmp    short loc_40116D
.text:00401180 StartAddress endp
.text:00401180
```

# 1. Analisi di persistenza

Il malware utilizza funzioni API di Windows, come RegOpenKeyExW, per accedere alla chiave di registro specificata. Questa chiave di registro, situata in HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run, è comunemente utilizzata per gestire i programmi che vengono eseguiti automaticamente all'avvio del sistema operativo.

Apertura della chiave di registro



```
040286F push 2          ; samDesired
0402871 push eax        ; ulOptions
0402872 push offset SubKey ; "Software\Microsoft\Windows\CurrentVersion\Run"
0402877 push HKEY_LOCAL_MACHINE ; hKey
040287C call esi ; RegOpenKeyExW
```

A questo punto, il programma carica nello stack tutti gli attributi necessari da passare alla funzione dedicata all'acquisizione della persistenza nel sistema operativo, come illustrato nell'immagine sottostante. La funzione utilizzata per questo scopo è RegSetValueExW, assicurando così che venga eseguito ad ogni avvio del sistema.

## Scrittura sul registro per ottenere la persistenza



```
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
```

## 2.Client Software

Il client software utilizzato dal malware è Internet Explorer 8.0, come si può dedurre dal parametro passato alla funzione InternetOpenA all'interno della subroutine.

“A” al termine della funzione InternetOpenA sta per ANSI, che è l’acronimo di “American National Standards Institute”. Questa codifica utilizza un solo byte per rappresentare i caratteri, consentendo di convertire i dati binari in caratteri secondo lo standard ANSI. Questo significa che InternetOpenA lavora con stringhe in formato ANSI, a differenza della versione Unicode della funzione (InternetOpenW), che utilizza due byte per carattere.



```
.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
```



### 3.Identificazione URL

Il malware tenta di connettersi all'URL “<http://www.malware12.com>”. Questo è evidente nella linea di codice che contiene push offset szUrl, dove szUrl è un puntatore alla stringa contenente l'URL.

```
.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
.tout=00401180
```



Quindi, la sequenza che porta alla connessione è:

1. L'URL è caricato nello stack (push offset szUrl).
2. L'handle di Internet (push esi), che è stato precedentemente ottenuto con la chiamata a InternetOpenA, è anch'esso caricato nello stack.
3. Viene chiamata la funzione InternetOpenUrlA tramite il registro edi, che si occupa di stabilire la connessione all'URL specificato.

## 4.Significato e funzionamento del comando assembly “Lea”

L'istruzione LEA (Load Effective Address) viene utilizzata per calcolare e memorizzare l'indirizzo effettivo di una locazione di memoria, non il contenuto della memoria stessa. Questo significa che LEA carica l'indirizzo di una variabile o di un valore all'interno dello stack in un registro della CPU, senza accedere direttamente alla memoria per leggere o scrivere dati.

Ad esempio, nel caso di riga 00402882 lea ecx, [esp+424h+Data], il programma memorizza nel registro ecx il valore della locazione di memoria della variabile [esp+424h+Data].



Cybersecurity Specialist

# Cyber Security & Ethical Hacking

GRAZIE.