



02/04/2024

# Windows malware

Prepared by:  
Manuel Buonanno

Organized by:




# Indice

1) Traccia.....	3
1.1) Configurazione macchina base.....	4
1.2) Configurazione macchina base usb.....	5
1.3) Configurazione macchina base cartelle condivise.....	6
1.4) Configurazione macchina base istantanee.....	7
2) Avvio malware.....	8
2.1) practicalmalwareanalysis.....	9
2.2) svchost.exe.....	10
3) Conclusioni.....	11

# 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".
- 

# Codice

```
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: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

.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↑to
.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+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
.text:00401180
```

# Persistenza

```
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
```

La chiamata alla funzione **RegOpenKeyEx**.

I parametri della funzione sono passati sullo stack tramite le istruzioni «push». Con questa funzione il malware accede alla chiave di registro <<Software\\Microsoft\\Windows\\CurrentVersion\\Run>> prima di modificarne il valore.

```
004028A8  push    ecx              ; lpValueName
004028A9  push    edx              ; hKey
004028AA  call    ds:RegSetValueExW
```

La chiamata alla funzione **RegSetValueEx**.

Anche in questo caso i valori sono passati sullo stack tramite le istruzioni «pushecx» e «pushedx». Tra le altre cose potete vedere come IDA identifichi il nome dei parametri passati alle funzioni. La funzione viene utilizzata dal malware per modificare il valore del registro ed aggiungere una nuova entry in modo tale da ottenere la persistenza all'avvio del sistema operativo.

# Client software

Il client utilizzato dal malware per connettersi ad internet è Internet Explorer, più precisamente la versione 8.

```
.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
```

# URL

Il malware cerca di connettersi all'URL [www.malware12.com](http://www.malware12.com). La chiamata di funzione che consente al malware la connessione verso un URL è «InternetOpenURL». L'URL è passato come parametro di questa funzione sullo stack, tramite l'istruzione push.

```
.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
```

# LEA

Il comando "lea" in assembly, che sta per "Load Effective Address", è utilizzato per caricare l'indirizzo di memoria di una variabile o di una locazione di memoria in un registro. Tuttavia, a differenza di "mov", che copia il valore effettivo, "lea" carica solo l'indirizzo, senza accedere al contenuto effettivo della memoria. Questo rende "lea" utile per calcolare indirizzi di memoria per operazioni successive. Ad esempio, il comando "lea" può essere utilizzato per calcolare l'indirizzo di una variabile e poi usare tale indirizzo in operazioni di accesso alla memoria o di calcolo.