


ESERCIZIO 1 SETTIMANA 11

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

**Esercizio**
Windows malware

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: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
```

4

**Esercizio**
Windows malware

Traccia:

```
.text:00401150 ; !!!!!!!!!!!!!!! SUBROUTINE !!!!!!!!!!!!!!!
.text:00401150
.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:00401168 mov esi, eax
.text:0040116D
.text:0040116D loc_40116D:
.text:0040116D push 0 ; CODE XREF: StartAddress+304j
.text:0040116F push 80000000h ; dwContext
.text:00401174 push 0 ; dwFlags
.text:00401176 push 0 ; dwHeadersLength
.text:00401178 push 0 ; lpszHeaders
.text:0040117D push offset szUrl ; "http://www.malware12.COM"
.text:0040117E call edi ; InternetOpenUrlA
.text:00401180 jmp short loc_40116D
.text:00401180 StartAddress endp
.text:00401180
.text:00401180
```

5

Persistenza del Malware:

Il malware cerca di ottenere persistenza nel sistema manipolando la chiave del Registro di sistema. Il segmento di codice coinvolto è il seguente:

Codice assembly

```
push 2 ; samDesired
push eax ; ulOptions
push offset Subkey ; "Software\Microsoft\Windows\CurrentVersion\Run"
push HKEY_LOCAL_MACHINE ; hKey
call esi ; RegOpenKeyExW
```

Qui, il malware apre la chiave del Registro di sistema "Software\Microsoft\Windows\CurrentVersion\Run" con privilegi appropriati (samDesired), cercando così di inserire se stesso tra i programmi in esecuzione all'avvio del sistema.

Client Software per la Connessione a Internet:

Il malware utilizza la libreria WinINet di Windows per effettuare una connessione a Internet. Il segmento di codice coinvolto è il seguente:

Codice assembly

```
push 0 ; dwFlags
push 0 ; lpszProxyBypass
push 0 ; lpszProxy
push 1 ; dwAccessType
push offset szAgent ; "Internet Explorer 8.0"
call ds:InternetOpenA
```

Qui, il malware apre una connessione a Internet utilizzando InternetOpenA e specifica "Internet Explorer 8.0" come agente utente.

URL di Connessione del Malware:

Codice assembly

```
push 0          ; dwContent
push 80000000h   ; dwFlags
push 0          ; dwHeadersLength
push 0          ; lpszHeaders
push offset szUrl ; "http://www.malware12com/"
push esi        ; hInternet
call edi        ; InternetOpenUrlA
```

Il malware tenta di connettersi all'URL "http://www.malware12com/" utilizzando InternetOpenUrlA. Questo segmento indica una possibile attività di download o comunicazione con un server remoto.

BONUS: spiegare il significato e i funzionamenti del comando "lea" assembly

Link di riferimento per il ricavo delle informazioni: <https://www.aldeid.com/wiki/X86-assembly/Instructions/lea>

L'istruzione lea (Load Effective Address) in assembly x86/x86-64 calcola e carica l'indirizzo effettivo di un operando nella destinazione specificata, senza accedere direttamente alla memoria. È spesso utilizzata per eseguire operazioni di calcolo degli indirizzi senza leggere o scrivere dati.