Windows Malware

Obiettivo: Analizzare l’estratto di un codice Malware

1) Codice della Traccia

Il codice da analizzare proposto dalla traccia è il seguente

Traccia:

```
)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
)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
```

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+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: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
```

2) Persistenza

La prima richiesta della traccia è di trovare come il **Malware** riesca ad ottenere la ***Persistenza***, ovvero quando si aggiunge alle entry dei programmi che vengono eseguiti all’avvio del PC senza l’azione dell’utente.

Le funzioni utilizzate dai Malware per creare una Persistenza sono in genere:

- **RegOpenKeyEx**: questa funzione permette di aprire una chiave di registro per poi modificarla.

- **RegSetValueEx**: permette di aggiungere un nuovo valore all’interno del registro e di configurarne i dati.

- **RegGetValue**: a differenza del precedente restituisce il tipo e il dato per un determinato registro.

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:lstrlenW
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
```

Nel codice proposto dalla traccia sono presenti

- **RegOpenKeyExW**: questa funzione è utilizzata per aprire la **Chiave di Registro** situata in *"Software/Microsoft/Windows/CurrentVersion/Run"* (in SubKey bisogna infatti precisare il nome della subkey del Registro che si vuole aprire) dove sono presenti le istruzioni *test* e *jnz* dove probabilmente se il valore che viene testato è già settato salta direttamente alla locazione indicata;

- **RegSetValueExW**: la funzione modifica il valore del Registro una volta che il **Malware** è riuscito ad aprirlo.

In questo specifico caso la lettera ***W*** delle due funzioni indica che entrambe accettano argomenti di tipo **Wide Character** per i nomi delle **Chiavi di Registro** (***wchar_t***, sono caratteri che occupano 2 byte invece di 1 byte come i caratteri ***ANSI***).

3) Client

Successivamente viene chiesto di identificare il **Client** utilizzato dal **Malware** per la connessione ad Internet. Nella seconda slide possiamo avere maggiori informazioni a riguardo.

```
.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
```

Possiamo vedere che in questa parte di codice il Malware utilizza il parametro ***szAgent*** che indica come Agent il browser **Internet Explorer 8.0** per poi successivamente chiamare la funzione ***InternetOpenA*** (usata per aprire una sessione di accesso ad Internet).

4) URL

Viene anche chiesto di indicare l'**URL** dove il **Malware** mira a connettersi.

```
.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
```

In questo caso abbiamo come parametro ***szUrl*** che indica accanto ad esso *"http://www.malware12.com"*, sarà quindi l'indirizzo Internet che il **Malware** inserirà e che vorrà aprire con la chiamata di funzione **InternetOpenUrlA**; per finire il **Malware** esegue un salto incondizionato per ripetere la procedura.

InternetOpenA e **InternetOpenUrlA** fanno parte entrambe della libreria *Wininet.dll* di Windows.

```
HINTERNET InternetOpenA(
  [in] LPCSTR lpszAgent,
  [in] DWORD  dwAccessType,
  [in] LPCSTR lpszProxy,
  [in] LPCSTR lpszProxyBypass,
  [in] DWORD  dwFlags
);
```

```
HINTERNET InternetOpenUrlA(
  [in] HINTERNET hInternet,
  [in] LPCSTR    lpszUrl,
  [in] LPCSTR    lpszHeaders,
  [in] DWORD     dwHeadersLength,
  [in] DWORD     dwFlags,
  [in] DWORD_PTR dwContext
);
```

5) Comando lea

L'istruzione ***lea*** (**Load Effective Address**) è utilizzata in **Assembly** per caricare l'indirizzo effettivo di una **Chiave di Registro** ma non il suo valore (a differenza di ***mov***). La sintassi è

lea destinazione, origine

dove la **Destinazione** può essere un Registro o una locazione di Memoria e l'**Origine** può essere un Registro, una costante o un indirizzo di Memoria; è utilizzata per "risparmiare" istruzioni perché riesce a calcolare indirizzi di memoria complessi come nel caso della prima slide dell'esercizio.

```
00402882 loc_402882:
00402882  lea     ecx, [esp+424h+Data]
00402886  push    ecx             ; lpString
00402887  mov     bl, 1
00402889  call    ds:lstrlenW
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
```