Floriana Feminò

16/01/2023

MALWARE ANALYSIS - WINDOWS MALWARE

Tasks:

1. **Individuazione delle istruzioni Assembly finalizzate all'ottenimento della persistenza da parte del malware**
2. **Identificazione del client software utilizzato per l'accesso ad Internet**
3. **Identificazione della chiamata di funzione all'URL al quale il malware tenta di connettersi tramite chiamata di funzione**
4. **Illustrazione 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
)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(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
```

---

1. Individuazione delle istruzioni Assembly finalizzate all'ottenimento della persistenza da parte del malware

Le attività odierne si incentrano sull'analisi di codice Assembly estratto da un malware di test destinato a sistemi operativi Windows.

I malware utilizzano molto spesso il registro per ottenere la cosiddetta **persistenza**: il malware aggiunge se stesso alle entry dei programmi che devono essere eseguiti all'avvio del PC, in modo tale da essere **avviato in maniera automatica** e permanente senza che sia necessaria un'azione dell'utente. Una delle chiavi di registro che vengono utilizzate dai malware per ottenere persistenza su un sistema operativo Windows è **Software\\Microsoft\\Windows\\CurrentVersion\\Run**.

Per ottenere la persistenza, il malware esegue due chiamate di funzione principali: **RegOpenKeyEx** per accedere alla key e **RegSetValueEx** per modificarla.

RegOpenKeyEx – i parametri della funzione sono passati allo stack tramite istruzioni push. Con questa funzione il malware accede alla chiave di registro prima di modificarne il valore:

```
)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
```

RegSetValueEx: vengono passati allo stack alcuni valori tramite istruzioni *push ecx* e *push edx*. Questa 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.

```
)04028A8   push    ecx             ; lpValueName
)04028A9   push    edx             ; hKey
)04028AA   call    ds:RegSetValueExW
```

2. Identificazione del client software utilizzato per l'accesso ad Internet

Il malware in oggetto tenta di inizializzare una connessione ad Internet. Il client software utilizzato per l'accesso ad Internet è **Internet Explorer 8.0**, come illustrato nell'istruzione seguente:

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

3. Identificazione della chiamata di funzione all'URL al quale il malware tenta di connettersi tramite chiamata di funzione

Microsoft mette a disposizione delle APIs per la gestione del networking ad ampio raggio, che prendono il nome di **WinInet APIs**; sono incluse nella libreria WinInet.dll. Le funzioni di questa libreria includono funzioni per **l'implementazione di protocolli di rete come HTTP ed FTP**. Tra le più usate ci sono

InternetOpen – questa funzione viene utilizzata per inizializzare una connessione ad Internet.

InternetOpenUrl – viene utilizzata per la **connessione ad un determinato URL**. Accetta, tra i vari parametri, un oggetto handler ad una connessione inizializzata con InternetOpen e l'URL per la connessione.

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

4. Illustrazione del comando Assembly "*lea*"

L'istruzione **lea** (**L**oad **E**ffective **A**ddress) viene utilizzata per posizionare un indirizzo di memoria nella destinazione indicata. La sintassi è la seguente:

lea destinazione, sorgente → es. lea eax, [esp+428h+Data]

```
)0402898   lea     eax, [esp+428h+Data]
```