ESERCITAZIONE WEEK 23 DAY 1

EPICODE | W23D1 - Pratica (1) PDF | Esercizio Windows malware

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

Traccia:

```
00040286F   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
```

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+ECTo
.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 ; ---------------------------------------------------------------------------
```

Il registro di Windows è utilizzato per salvare informazioni sul sistema operativo come ad esempio configurazioni del sistema stesso o delle applicazioni che girano sul sistema. I malware utilizzano molto spesso il registro per ottenere quella che viene chiamata **persistenza**: il malware aggiunge sé stesso alle entry dei programmi che devono essere avviati all'avvio del PC in modo tale da essere eseguiti in maniera automatica e permanente senza l'azione dell'utente.

I malware utilizzano le funzioni per modificare i valori delle chiavi di registro che sono parte delle APIs di Windows per ottenere la persistenza e fare in modo che il sistema operativo stesso li avvii nelle fasi iniziali di start-up.

Nel codice proposto, sono evidenti le chiamate a queste funzioni che agiscono sulle chiavi di registro, in particolare:

- **RegOpenKeyExW:** questa funzione permette di aprire una chiave di registro per poi modificarla in seguito. In particolare, prende in ingresso la chiave di tipo handle hKey, **HKEY_LOCAL_MACHINE**, ovvero la **root key** dove sono contenuti i record e le configurazioni della macchina. Come sottochiave prende in ingresso la subkey **Software\\Microsoft\\Windows\\CurrentVersion\\Run** e altri due input ulOptions=eax (specifica le opzioni aggiuntive per l'apertura della chiave di registro) e samDesired=2. Tutti questi input vengono passati sullo stack tramite le istruzioni **push. SamDesired =2** potrebbe significare di mettere tale maschera al valore KEY_SET_VALUE (0x0002), che equivale a creare, eliminare o modificare un valore nel registro, oppure KEY_WRITE, indicando l'intenzione di ottenere l'autorizzazione per scrivere nella chiave di registro;
- **RegSetValueKeyW:** questa funzione permette invece di aggiungere un nuovo valore all'interno del registro e di settare i rispettivi dati. Accetta come parametri la chiave hKey edx, la sottochiave e il dato da inserire. Anche in questo caso gli input sono spinti sullo stack con la push.

In questo modo viene ottenuta la persistenza.

Segue una parte di codice in cui un malware tenta di connettersi a Internet.

- Inizialmente chiama la funzione **InternetOpenA** per inizializzare una connessione verso Internet, per mezzo del **client Internet Explorer 8.0**;
- Poi chiama la funzione **InternetOpenUrlA**,utilizzata invece per la connessione ad un determinato URL. Accetta, tra gli altri parametri, un oggetto handler ad una connessione inizializzata con InternetOpenA, e l'URL per la connessione, in questo caso **http://www.malware12com**.
  Segue un loop infinito di chiamate a tale funzione tramite l'istruzione jump incondizionale.

Si provano a dettagliare tutte le righe di codice:

Traccia:

```
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
```

push 2 ; samDesired	È uno degli input della funzione **RegOpenKeyExW**. In particolare potrebbe significare di mettere tale maschera al valore **KEY_SET_VALUE (0x0002)**, che equivale a creare, eliminare o modificare un valore nel registro, oppure **KEY_WRITE (0x20006)**, indicando l'intenzione di ottenere l'autorizzazione per scrivere nella chiave di registro. L'istruzione push serve a spingere tale valore nello stack della funzione che verrà chiamata.
push eax ;ulOptions	È uno degli input della funzione **RegOpenKeyExW**. Specifica le opzioni per l'apertura della chiave. L'istruzione push serve a spingere tale valore nello stack della funzione che verrà chiamata. Per il controllo del valore vedere riga istruzione **test.**
push offset SubKey ; "Software\\Microsoft\\Windows\\CurrentVersion\\Run"	È uno degli input della funzione **RegOpenKeyExW**. L'istruzione push serve a spingere tale valore nello stack della funzione che verrà chiamata. Questo parametro specifica il nome della sottochiave che si desidera aprire, rispetto alla root key (vedi sotto).
push HKEY_LOCAL_MACHINE ;hKey	È uno degli input della funzione **RegOpenKeyExW**. L'istruzione push serve a spingere tale valore nello stack della funzione che verrà chiamata. Si tratta del handle alla chiave che si vuole aprire, in questo caso **HKEY_LOCAL_MACHINE**, ovvero la **root key** dove sono contenuti i record e le configurazioni della macchina.
call esi ; RegOpenKeyExW	Chiamata della funzione **RegOpenKeyExW** usando l'indirizzo contenuto nel registro **esi.**
test eax,eax	AND bit a bit tra il valore contenuto nel registro accumulatore e se stesso. **Serve per verificare se il contenuto di eax è uguale o diverso da 0**. Poiché eax è correlato all'input ulOptions, il codice vuole controllare se **ulOptions = 0** ovvero **se la funzione RegOpenKeyEx verrà eseguita con le opzioni predefinite senza alcuna modifica al comportamento standard**, **o se invece seguirà delle modifiche (ulOptions = 1).**
jnz loc_4028C5	Conditional jump: il salto alla locazione 4028C5 è eseguito se **l'AND precedente è diverso da 0, quindi quando ZF=0**. In sostanza **salta se eax non è 0**, quindi se **ulOptions non è 0**, **volendo impostare un comportamento diverso delle opzioni della funzione**. Se

	invece **eax = ulOptions =0**, per **eseguire le opzioni predefinite si procederà** a eseguire le linee di codice successive, **non eseguendo il salto**.
lea ecx, [esp+424h+Data]	Calcola l'indirizzo effettivo della variabile o del dato memorizzato nell'indirizzo di **memoria [esp + 424h + Data]** e lo carica nel registro ECX senza effettuare alcuna operazione sui dati.
push ecx ; lpString	Spinge nello stack un puntatore a una stringa contenuto in ecx, che sarà l'input per la funzione chiamata.
mov bl, 1	Mette a 1 il valore di bl(registro) che potrebbe essere un flag o un parametro della funzione che viene chiamata dopo.
call ds ; lstrlenW	Chiamata alla funzione **lstrlenW** che calcola la lunghezza della stringa in input.
lea edx, [eax +eax+2]	L'istruzione calcola l'indirizzo effettivo di [eax + eax + 2] e lo carica nel registro edx.
push edx ; cbData	Spinge nello stack l'input della funzione **RegSetValueExW** che verrà chiamata in seguito, nel quale è contenuta la lunghezza in byte dei dati da scrivere. Questo valore è nel registro edx.
mov edx, [esp+428h+hKey]	Muove il contenuto della memoria presente in **[esp+428h+hKey]** nel registro **edx**. hKey dovrebbe essere un puntatore a una chiave di registro.
lea eax, [esp+428h+Data]	Carica l'indirizzo effettivo di **[esp+428h+Data]** nel registro **eax**. Data potrebbe essere un puntatore ai dati da scrivere nel registro.
push eax ; lpData	Mette il contenuto del registro **eax** nello stack. Si tratta di un input della funzione, che corrisponde al puntatore ad un buffer che contiene i dati da salvare.
push 1 ; dwType	Mette il valore 1 nello stack, liberando spazio per un altro input della funzione, ovvero il tipo di dati da scrivere. Essendo a 1 significa che vanno scritti dati di tipo REG_SZ (stringa zero-terminata).
push 0 ; Reserved	Questa istruzione mette il valore 0 nello stack, valore riservato. Si tratta di un input della funzione, che va messo sempre a 0.
lea ecx, [esp+434h+ValueName]	Carica l'indirizzo effettivo di **[esp+434h+ValueName]** nel registro **ecx**. ValueName potrebbe essere un puntatore al nome del valore da impostare.
push ecx ; lpValueName	Mette il contenuto del registro ecx nello stack per creare spazione per un altro input della funzione, ovvero un puntatore al nome del valore (lpValueName).
push edx; hKey	Mette il contenuto del registro edx nello stack per spingere l'input della funzione che rappresenta un handle alla chiave di registro (hKey).
call ds: RegSetValueExW	Chiamata alla funzione **RegSetValueExW** per impostare un valore in una chiave del Registro di sistema di Windows.