PRATICA S11\L1

SCOPO: rispondere alle domande che seguono ,faccendo riferimento all'estratti di malware sotto.

1-Descrivere come il malware ottiene la persistenza,ed evidenziare il codice assembly dove le relative istruzioni e chiamate di funzioni vengono eseguite.

2-indentificare 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.

4-Dire 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
```

PRATICA S11\L1

```
.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,ed evidenziare il codice assembly dove le relative istruzioni e chiamate di funzioni vengono eseguite.

La parola **Persistenza** quando si tratta di malware; si riferisce al processo attraverso il quale un malware cerca di mantenere la sua presenza sul sistema infetto nel tempo. Questo può avvenire attraverso varie tecniche, come l'installazione di file o registri di avvio automatico, l'utilizzo di servizi di sistema, l'aggiunta di voci al registro di sistema o altre modalità che consentono al malware di avviarsi automaticamente ad ogni avvio del sistema o di rimanere attivo anche dopo un riavvio.

Il codice assembly fornito mostra come il malware ottiene la persistenza nel sistema. In particolare, il malware sta eseguendo una serie di operazioni per aggiungere una voce di registro che fa riferimento a un percorso eseguibile nel registro di avvio del sistema operativo. Questo assicura che il malware venga avviato ogni volta che il sistema si avvia.

- Evidenziazione de codice

PRATICA S11\L1

Il codice assembly sotto; evidenzia le relative istruzioni e chiamate di funzioni che vengono eseguite.

```
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
```

2-indentificare il client software utilizzato dal malware per la connessione ad internet.

Nel secondo codice assembly fornito, il malware utilizza la funzione InternetOpenA per aprire una connessione all'Internet. Questa funzione fa parte delle API di WinINet di Windows e viene comunemente utilizzata per creare un "handle" di sessione per l'accesso a 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.

Nel secondo codice assembly, l'URL a cui il malware tenta di connettersi è "http://www.malware12.com". Questo viene evidenziato dalla stringa "szUr1" nella porzione di codice assembly fornita. La chiamata di funzione che consente al malware di connettersi a questo URL è InternetOpenUrlA, che

PRATICA S11\L1

viene utilizzata per aprire un URL specifico all'interno della sessione Internet creata con InternetOpenA.

BONUS:

4-Dire qual è il significato e il funzionamento del comando assembly “lea”.

Il comando assembly "lea" (Load Effective Address) calcola l'indirizzo effettivo di una destinazione e lo carica in un registro senza eseguire l'accesso alla memoria. In altre parole, calcola l'indirizzo della destinazione specificata e lo assegna a un registro senza leggere il valore dalla memoria. Viene spesso utilizzato per calcolare gli indirizzi di memoria per l'accesso a variabili o dati senza effettuare effettivamente l'accesso ai dati stessi.