

# S11/L1

GIULIA FIACCHI

# TRACCIA

Con riferimento agli estratti di un malware reale presenti nelle prossime slide, rispondere alle seguenti domande.

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

```
0040286F 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
```

```
.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+ECD  
.text:00401150     push    esi  
.text:00401150     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:00401168     mov     esi, eax  
.text:0040116D  
.text:0040116D loc_40116D: ; CODE XREF: StartAddress+304j  
.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.malware12.COM"  
.text:0040117D     push    esi        ; hInternet  
.text:0040117E     call    edi ; InternetOpenUrlA  
.text:00401180     jmp     short loc_40116D  
.text:00401180 StartAddress endp  
.text:00401180  
.text:00401180 .
```

# PARTE 1

## COME OTTIENE LA PERSISTENZA

La persistenza è in genere ottenuta assicurandosi che il malware si avvii automaticamente all'avvio del sistema o quando l'utente effettua l'accesso.

Per identificare la persistenza, in genere cercheremmo chiamate a funzioni come RegCreateKeyEx, RegSetValueEx, o operazioni di file che suggeriscono che il malware si sta copiando in una posizione di avvio.

```
0040286F 01 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
```

questa funzione permette di aprire una chiave di registro al fine di modificarla

La chiave di registro che viene utilizzata è questa:



# PARTE 2

## CLIENT SOFTWARE

Il malware usa le funzioni **InternetOpenUrlA** questa funzione viene utilizzata per inizializzare una connessione verso Internet, che fanno parte della [WinINet API](#) ( includono funzioni per l'implementazione di protocolli di rete come HTTP ed FTP) . Ciò indica che il malware sta usando le librerie di [Internet Explorer](#) per connettersi a Internet. In particolare, *Internet Explorer 8.0* è referenziato nel codice con la stringa "Internet Explorer 8.0" passata come argomento a **InternetOpenA**.

```
.text:00401150 ; DWORD __stdcall StartAddress(LPUOID)
.text:00401150 StartAddress    proc near             ; DATA XREF: sub_401040+ECT
.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
```

# PARTE 3

## CHIAMATA DI FUNZIONE

- L'URL a cui il malware sta tentando di connettersi è "http://www.malware120.com". Questo viene inserito nello stack all'indirizzo 0x00401170 nel codice:

```
push offset szUrl ; "http://www.malware120.com"
```

- La funzione responsabile della connessione a questo URL è InternetOpenUrlA, che viene chiamata in 0x00401175:

```
call esi ; InternetOpenUrlA
```

Questa chiamata viene utilizzata per aprire l'URL con i parametri specificati, avviando la connessione al sito dannoso.

```
.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.malware120.COM"
.text:0040117D          push    esi        ; hInternet
.text:0040117E          call    edi        ; InternetOpenUrlA
.text:00401180          jmp     short   loc_401160
.text:00401180 StartAddress
.text:00401180 endp
```

# BONUS

## COMANDO “LEA”

L'istruzione LEA (Load Effective Address) nel linguaggio assembly viene utilizzata per calcolare l'indirizzo di una locazione di memoria e caricarlo in un registro senza accedere direttamente al contenuto della memoria stessa. È particolarmente utile per calcolare indirizzi o eseguire operazioni aritmetiche su indirizzi.

Usi comuni di LEA:



- Ottimizzazione:** Viene utilizzato per calcolare velocemente indirizzi o eseguire operazioni aritmetiche semplici, sfruttando la capacità dell'unità di indirizzamento del processore.
- Calcolo di indirizzi in strutture dati:** Utile per accedere a campi di strutture dati o array.
- Aritmetica senza influenzare flag:** A differenza delle istruzioni aritmetiche come ADD, SUB, ecc., LEA non modifica i flag condizionali del processore, rendendola utile in situazioni dove è necessario mantenere intatti i flag.