# S11/L1

Il lavoro di oggi prevede l'analisi di un malware in assembly.
La traccia seguente prevede:

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

La parte di codice nella quale il malware ottiene la persistenza è quella evidenziata in rosso nella figura:

```
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
```

# PERSISTENZA

La chiamata alla funzione RegOpenKeyEx. Notate come i parametri della funzione sono passati sullo stack tramite le istruzioni «push». Con questa funzione il malware accede alla chiave di registro prima di modificarne il valore.

Una cosa importante da notare, è che abbiamo visto qual è una delle chiavi di registro che viene utilizzata dai malware per ottenere persistenza su un sistema operativo Windows.
**Software\\Microsoft\\Windows\\CurrentVersion\\Run**

# CLIENT

Il client software utilizzato è Internet Explorer , versione 8.0.

```
push    offset szAgent  ; "Internet Explorer 8.0"
```

# URL E CHIAMATA DI FUNZIONE

```
push    0               ; dwContext
push    80000000h       ; dwFlags
push    0               ; dwHeadersLength
push    0               ; lpszHeaders
push    offset szUrl    ; "http://www.malware12COM
push    esi             ; hInternet
call    edi ; InternetOpenUrlA
```

Nell'immagine sono identificati rispettivamente l'url utilizzato "http://www.malware12.COM".
La chiamata di funzione è **InternetOpenUrl**.

# COMANDO LEA

Il comando LEA (Load Effective Address) in assembly viene utilizzato per caricare l'indirizzo effettivo di un operando in un registro, anziché caricare il valore di quell' operando stesso.
L'istruzione LEA calcola l'indirizzo effettivo dell'operando e lo memorizza nel registro di destinazione, senza eseguire un accesso effettivo alla memoria per recuperare il valore memorizzato all'indirizzo calcolato.

Ad esempio, l'istruzione LEA può essere utilizzata per calcolare l'indirizzo di una variabile o di un'area di memoria e memorizzarlo in un registro, in modo che il registro contenga l'indirizzo effettivo al quale è archiviato il dato desiderato.
Questo può essere utile per calcolare gli indirizzi dei dati senza doverli effettivamente leggere o scrivere in memoria, ma solo per eseguire calcoli basati su quegli indirizzi.

Un esempio di utilizzo di LEA potrebbe essere:

```
LEA EAX, [EBX + ECX*2]
```

Questo comando carica nel registro EAX l'indirizzo effettivo di EBX + ECX*2, senza effettuare alcun accesso effettivo in memoria.