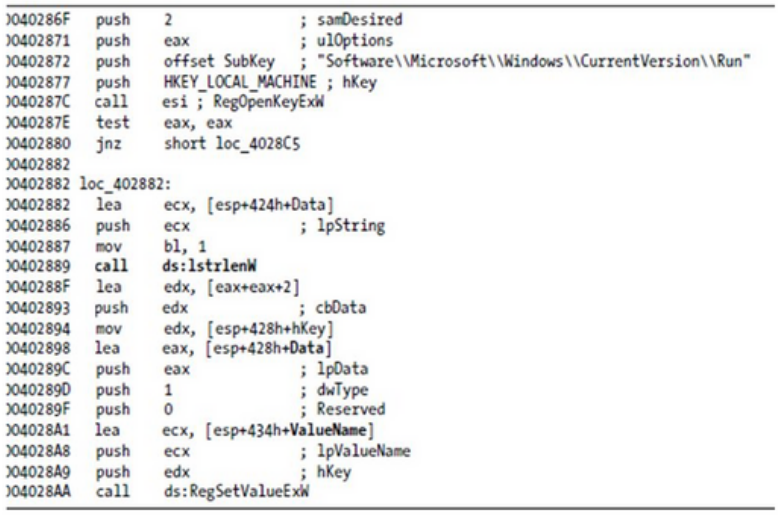
Esercizio S11 L1

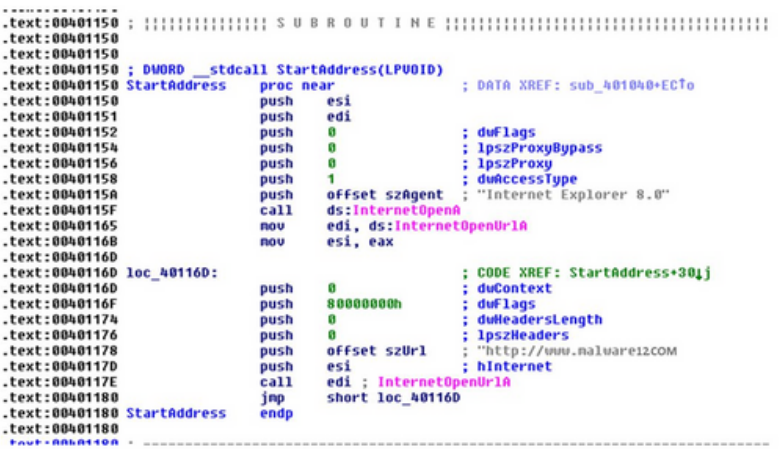
Traccia:

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.

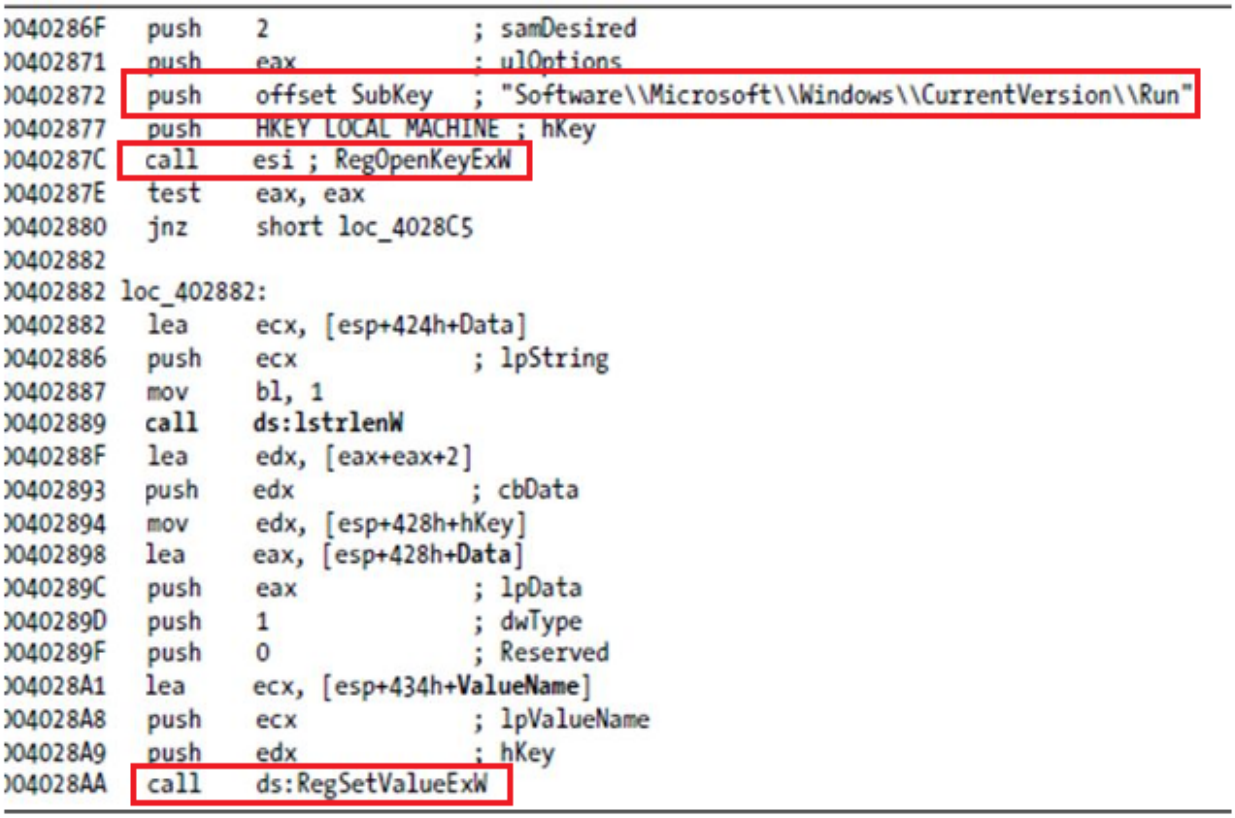


```
00040286F   push    2               ; samDesired
00402871    push    eax             ; ulOptions
00402872    push    offset SubKey   ; "Software\\Microsoft\\Windows\\CurrentVersion\\Run"
00402877    push    HKEY_LOCAL_MACHINE ; hKey
0040287C    call    esi ; RegOpenKeyExW
0040287E    test    eax, 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+ECTo
.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
```

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



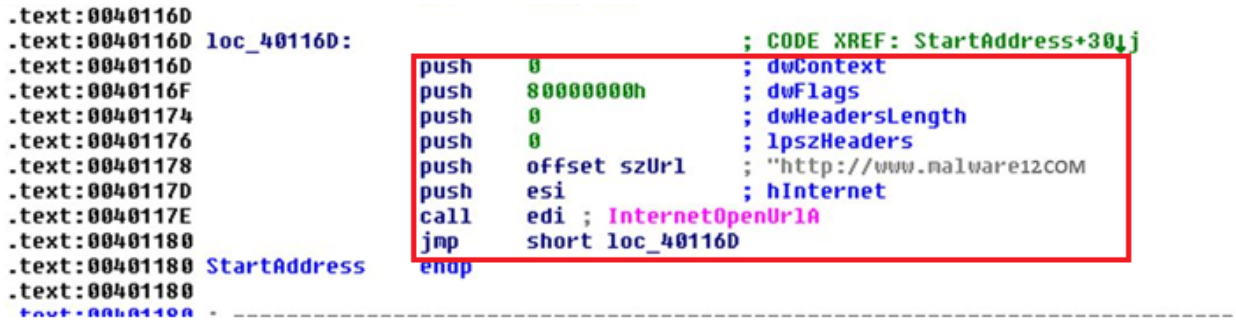
```
)040286F   push    2               ; samDesired
)0402871   push    eax             ; ulOptions
)0402872   push    offset SubKey   ; "Software\\Microsoft\\Windows\\CurrentVersion\\Run"
)0402877   push    HKEY_LOCAL_MACHINE ; hKey
)040287C   call    esi ; RegOpenKeyExW
)040287E   test    eax, eax
)0402880   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
```

Il malware inserisce un nuovo valore all'nterno della chiave di registro che contiene tutti i programmi avviati all'apertura del sistema operativo. Poi per ottenere la persistenza utilizza due funzioni, **RegOpenKeyExW** e **RegSetValueExW.** La prima funzione permette di aprire la chiave selezionata, la seconda imposta i dati e il tipo di un valore specificato in una chiave di registro.

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

Il client software utilizzato dal malware per la connessione ad internet è “**Internet Exploer 8.0**”.

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.



L’URL al quale tenta di connettersi è “**http://www.malware12.com**”. Possiamo notare evidenziata in figura la chiamata alla funzione “**InternetOpenUrlA**“ che permette al malware di connettersi all’URL passato precendentemente con il push.

BONUS

Qual è il significato e il funzionamento del comando assembly "lea".

Il comando assembly "**lea**" (Load Effective Address) è utilizzato per caricare in un registro l'indirizzo effettivo di una certa variabile. In altre parole, l'istruzione "lea" calcola l'indirizzo di un'operando e lo carica in un registro senza accedere effettivamente alla memoria per ottenere il valore.
La sintassi generale dell'istruzione "lea" è la seguente:
lea destinazione, operando
Dove "destinazione" è il registro in cui l'indirizzo sarà caricato e "operando" è l'operando per il quale si vuole ottenere l'indirizzo effettivo.

L'utilità principale di "lea" è nell'effettuare calcoli di indirizzi in modo efficiente, senza dover accedere alla memoria per ottenere il valore reale. Ciò può essere utile in situazioni in cui si desidera solo l'indirizzo di un'operando per eseguire successivamente operazioni come l'accesso a una struttura dati o il calcolo di indirizzi di salti condizionali.