# Windows malware

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

Identificazione del client software utilizzato dal malware per la connessione a internet

Identificazione dell'URL al quale il malware tenta di connettersi ed evidenziazione della chiamata di funzione che permette al malware di connettersi ad un URL

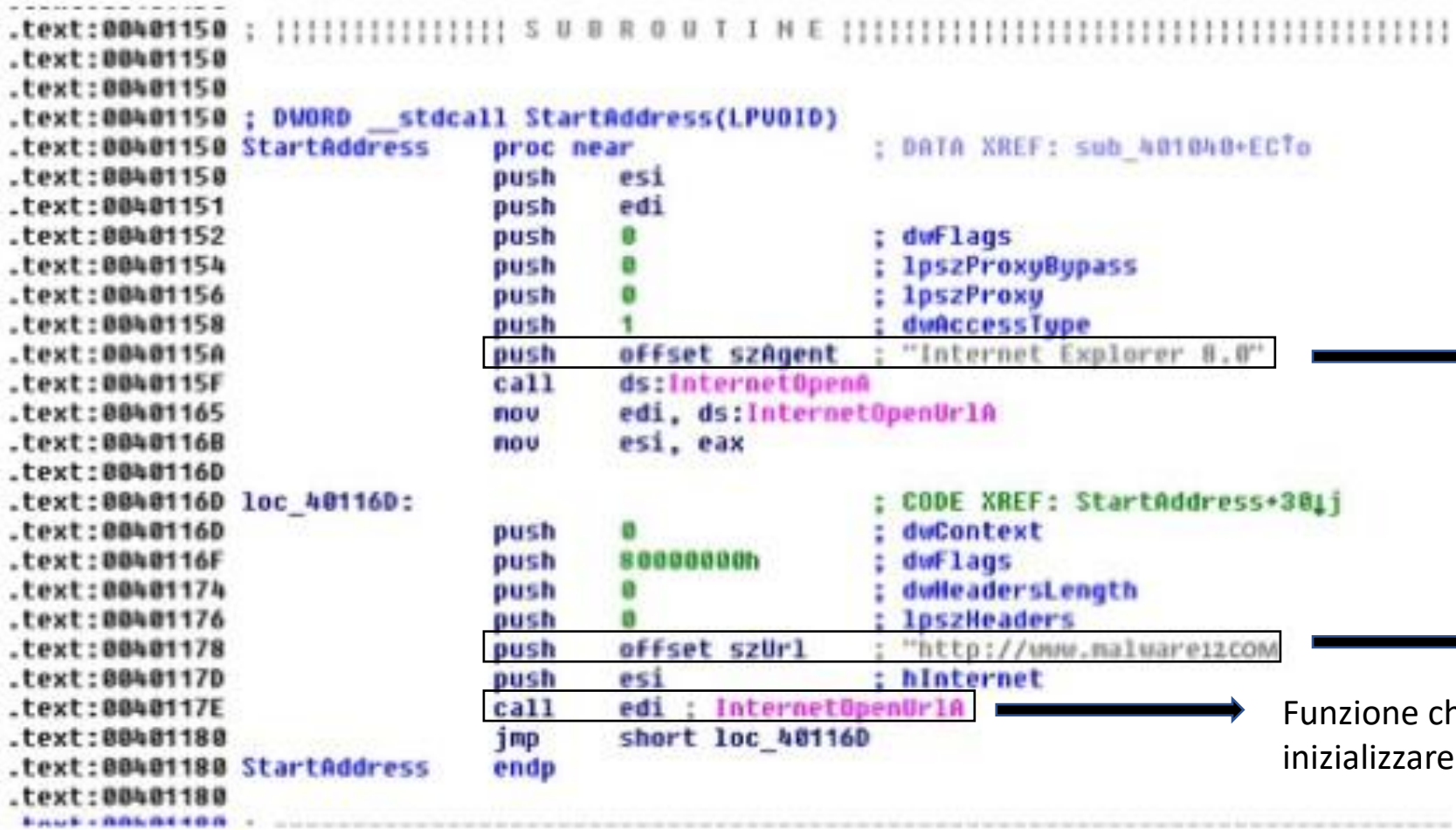
Significato e funzionamento del comando *lea*

Orlando Tangari

```
)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
```

Funzione che permette di aprire una chiave di registro e modificarla

Funzione che permette di aggiungere un nuovo valore all'interno dell'registro e di settare i rispettivi dati




Client software utilizzato dal malware

URL al quale il malware tenta di connettersi

Funzione che viene utilizzata per inizializzare una connessione verso internet

# Load Effective Address

Alcune architetture di set di istruzioni, come Intel x86 e IBM/360 e i suoi successori, hanno un'istruzione di indirizzo efficace del carico.

Ciò esegue un calcolo dell'indirizzo operativo efficace, ma invece di agire su quella posizione di memoria, carica l'indirizzo che sarebbe stato accessibile in un registro.
Questo può essere utile quando si passa l'indirizzo di un elemento di matrice a una subroutine.

Può anche essere un modo intelligente di fare più calcoli del normale in una istruzione; ad esempio, l'utilizzo di tali istruzioni con la modalità di indirizzamento
"base + indice + offset" consente di aggiungere due registri e una costante insieme in un'istruzione e memorizzare il risultato in un terzo registro.

fonte: https://en.wikipedia.org/wiki/Addressing_mode#Useful_side_effect