

Windows Malware

Il malware garantisce la sua persistenza aggiungendo una voce nuova alla chiave di registro **Software\Microsoft\Windows\CurrentVersion\Run**, la quale contiene tutti i programmi avviati automaticamente all'avvio del sistema operativo. Per fare ciò, sfrutta le funzioni **RegOpenKey**, che apre la **chiave di registro** desiderata, con i parametri passati attraverso istruzioni **push** prima della chiamata di funzione, e **RegSetValueEx**, che consente al malware di inserire un nuovo valore nella **chiave di registro** appena aperta.

Come verificabile dall'immagine sottostante, il malware utilizza la libreria **WININET.dll**, identificandosi come **Internet Explorer**, versione **8**, per connettersi a Internet.

```
.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
```

Il malware tenta di stabilire una connessione all'URL **www.malware12.com**. Per farlo, utilizza la funzione di chiamata **InternetOpenURL**, passando l'URL come parametro tramite un'istruzione **push** nello **stack**.

```
.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
```

Bonus: Il comando assembly **lea** (**Load Effective Address**) serve per caricare l'indirizzo effettivo di un operando nella memoria di un registro specifico. Quindi, consente di ottenere l'indirizzo di una variabile, un array o un'etichetta e di memorizzarlo in un registro per un successivo utilizzo.

Il comando **lea** calcola l'indirizzo effettivo sommando i valori di **base**, **scala * indice** e **offset**. Il risultato viene quindi memorizzato nel registro di destinazione.

Vantaggi:

- Il comando **lea** è più efficiente rispetto all'utilizzo di istruzioni di spostamento e caricamento separate, in quanto richiede una singola istruzione per calcolare e memorizzare l'indirizzo.
- Può essere utilizzato per accedere a dati in modo flessibile utilizzando registri, indici e offset.

Svantaggi:

- Il comando **lea** non carica il valore effettivo dei dati nella memoria nell'indirizzo calcolato. Carica solo l'indirizzo stesso.
- Può essere più difficile da comprendere rispetto a istruzioni di spostamento e caricamento separate, soprattutto per i principianti.