S11/L1

Malware Analysis

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

Come il malware ottiene la persistenza

Il malware ottiene la persistenza modificando il registro di Windows per eseguire il proprio codice ad ogni avvio del sistema.

In particolare:

- La chiamata **RegOpenKeyExW** apre il percorso del registro **HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run.**

- **RegSetValueExW** imposta un valore nel registro che contiene il percorso del malware, garantendo che venga eseguito ad ogni avvio del sistema.

Client software utilizzato per la connessione ad Internet

Il client software utilizzato dal malware per la connessione ad Internet è "Internet Explorer 8.0". Questo è evidenziato dalla stringa "Internet Explorer 8.0" nel codice:

```
.text:0040115A                push    offset szAgent  ; "Internet Explorer 8.0"
.text:0040115F                call    ds:InternetOpenA
```

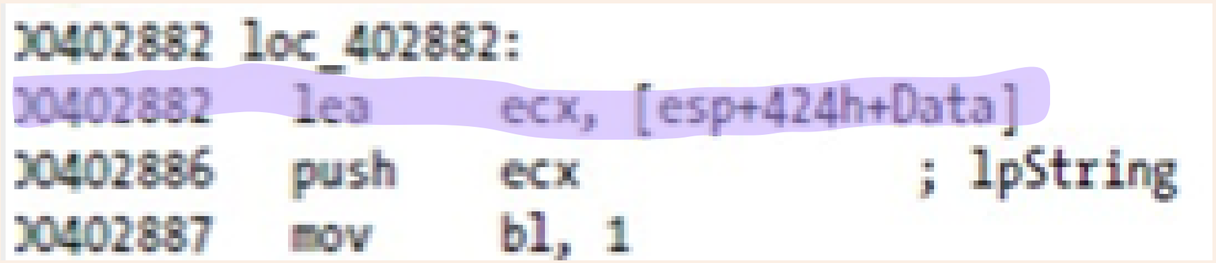
URL al quale il malware tenta di connettersi e chiamata di funzione

L'URL al quale il malware tenta di connettersi è http://www.malware12.com. La chiamata di funzione che permette al malware di connettersi a questo URL è la seguente:

```
.text:00401178                push    offset szUrl    ; "http://www.malware12COM
.text:0040117D                push    esi             ; hInternet
.text:0040117E                call    edi ; InternetOpenUrlA
```

BONUS: significato e funzionamento del comando assembly "lea”

Il comando lea (Load Effective Address) in assembly viene utilizzato per caricare l'indirizzo effettivo di un'operazione nella destinazione specificata. Non esegue l'operazione di dereferenziazione, ma calcola l'indirizzo e lo memorizza nel registro di destinazione. È comunemente utilizzato per operazioni di manipolazione di puntatori e calcolo di indirizzi.



Ad esempio, questo calcola l'indirizzo dell'operando [esp+424h+Data] e lo memorizza nel registro ecx senza dereferenziare l'indirizzo calcolato.