

WINDOWS MALWARE

Con riferimento alle tracce presenti

- Descrivere come il malware ottiene persistenza
- Identificare il client software utilizzato
- Identificare l'URL al quale il malware tenta la connessione

Traccia:

```
X040286F push 2 ; samDesired
X0402871 push eax ; ulOptions
X0402872 push offset SubKey ; "Software\\Microsoft\\Windows\\CurrentVersion\\Run"
X0402877 push HKEY_LOCAL_MACHINE ; hKey
X040287C call esi ; RegOpenKeyExW
X040287E test eax, eax
X0402880 jnz short loc_4028C5
X0402882 loc_402882:
X0402882 lea ecx, [esp+424h+Data]
X0402886 push ecx ; lpString
X0402887 mov bl, 1
X0402889 call ds:strlenW
X040288F lea edx, [eax+eax+2]
X0402893 push edx ; cbData
X0402894 mov edx, [esp+428h+hKey]
X0402898 lea eax, [esp+428h+Data]
X040289C push eax ; lpData
X040289D push 1 ; dwType
X040289F push 0 ; Reserved
X04028A1 lea ecx, [esp+434h+ValueName]
X04028A8 push ecx ; lpValueName
X04028A9 push edx ; hKey
X04028AA call ds:RegSetValueExW
```

Parametri passati alla funzione RegOpenKeyEx, tramite push (sullo stack)

Notare quale chiave di registro viene utilizzata dal malware per ottenere persistenza

Funzione che il malware usa per accedere alla chiave prima di modificare il valore

Funzione che il malware usa per modificare il valore del registro e aggiungere una nuova entry

Spesso i Malware utilizzano le funzioni per modificare i valori delle chiavi di registro. Nel codice possiamo notare come apre una chiave di registro, per aggiungere un valore, in modo tale da ottenere persistenza. I Malware aggiungono se stessi alle entry dei programmi che devono essere eseguiti all'avvio del pc. In modo da essere automatici e permanenti senza azioni da parte dell'utente.

Traccia:

```
.text:00401150 ; SUBROUTINE
.text:00401150
.text:00401150 ; DWORD __stdcall StartAddress(LPVOID)
.text:00401150 StartAddress proc near ; DATA XREF: sub_401040+EC70
.text:00401150 push esi
.text:00401151 push edi
.text:00401152 push 0 ; dwFlags
.text:00401154 push 0 ; lpzProxyBypass
.text:00401156 push 0 ; lpzProxy
.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:00401168 mov esi, eax
.text:0040116D loc_40116D: ; CODE XREF: StartAddress+301j
.text:0040116D push 0 ; dwContext
.text:0040116F push 80000000h ; dwFlags
.text:00401174 push 0 ; dwHeadersLength
.text:00401176 push 0 ; lpzHeaders
.text:00401178 push offset szUrl ; "http://www.malware12.com"
.text:0040117D push esi ; hInternet
.text:0040117E call edi ; InternetOpenUrlA
.text:00401180 jmp short loc_40116D
.text:00401180 StartAddress endp
.text:00401180
```

CLIENT SOFTWARE

FUNZIONE PER INIZIALIZZARE CONNESSIONE VERSO INTERNET

URL

FUNZIONE PER CONNESSIONE VERSO UN DETERMINATO URL

BONUS - ISTRUZIONE "lea"

Questa istruzione copia l'effettivo valore esadecimale a 16 bit di una etichetta, passata come operando sorgente, nel registro di destinazione.

Il registro coinvolto per ricevere l'**offset*** del puntatore associato all'etichetta può essere uno qualunque dei registri a 16 bit.

Lea edx [eax+eax+2] – valore esadecimale **[aex+eax+2]** / registro di destinazione **edx**

["http://www.giobe2000.it/Tutorial/Schede/07-IstruzioniCpu/LEA.asp"](http://www.giobe2000.it/Tutorial/Schede/07-IstruzioniCpu/LEA.asp)

*L'offset di un'istruzione può essere usato per spostare il codice di una quantità specifica di byte e, come operatore del compilatore, può indicare che si sta facendo riferimento all'indirizzo di un valore e non al valore stesso. Offset è la prima istruzione che leggerà in fase di compilazione