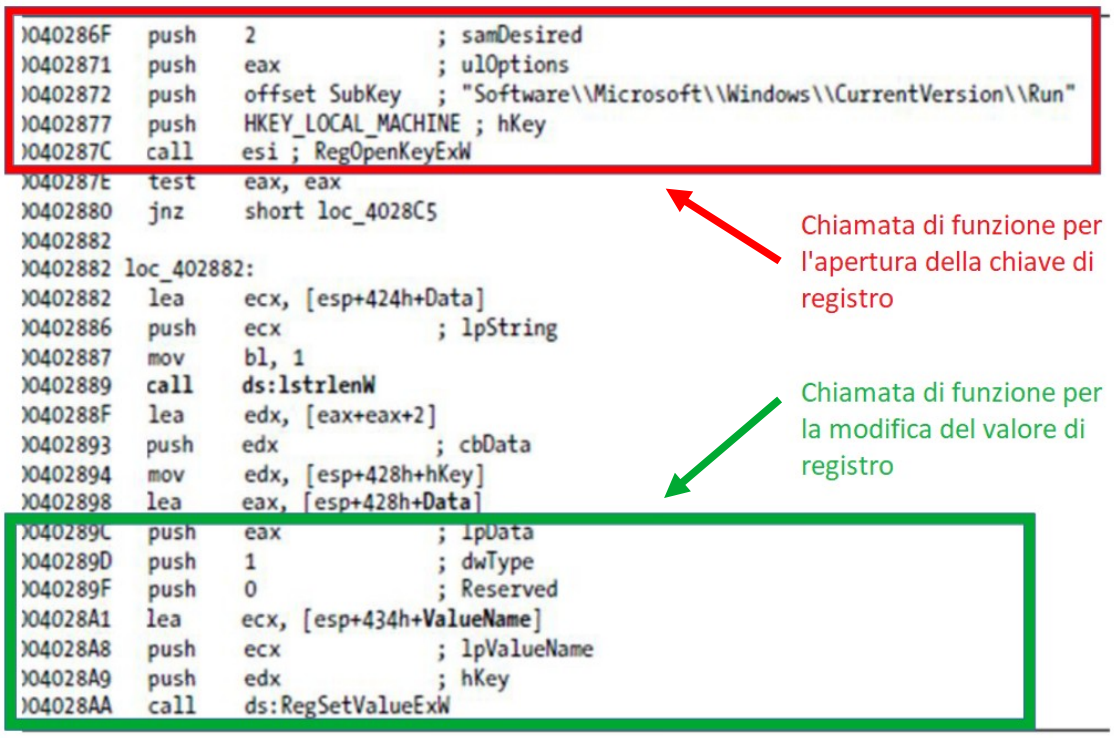
REPORT 16/01/2023 **LORENZO SCORBAIOLI**

TASK:

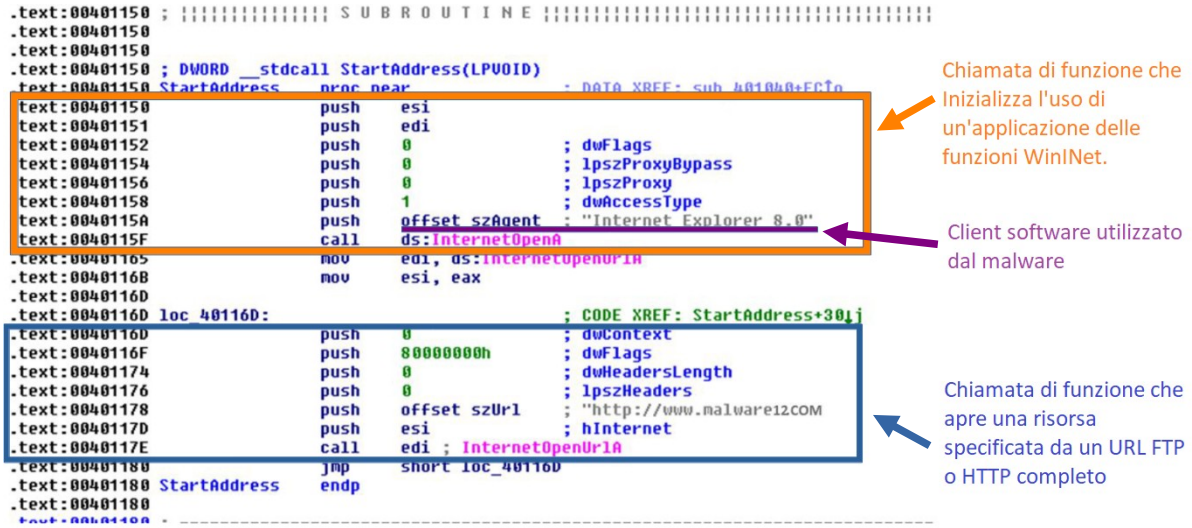
- Descrivere come il malware ottiene la persistenza, evidenziando il codice assembly dove le relative istruzioni e chiamate di funzioni vengono eseguite
- Identificare il client sofware utilizzato dal malware per la connessione ad internet
- Identificare l'URL al quale il malware tenta di connettersi ed evidenziare la chiamata di funzione che permette al malware di connettersi ad un URL
- BONUS: qual'è il significato ed il funzionamento del comando assembly "lea"



Nel riquadro rosso abbiamo la chiamata di funzione che apre la chiave del Registro di sistema specificata. Si noti che i nomi delle chiavi non sono distinzione tra maiuscole e minuscole[1].

Nel riquadro verde abbiamo invece la funzione che imposta i dati e il tipo di un valore specificato in una chiave del Registro di sistema[2]. Qua viene modificata la chiave di registro in modo da ottenere la persistenza.

1 https://learn.microsoft.com/it-it/windows/win32/api/winreg/nf-winreg-regopenkeyexw
2 https://learn.microsoft.com/it-it/windows/win32/api/winreg/nf-winreg-regsetvalueexw



Nel riquadro arancione abbiamo la funzione di chiamata che si occupa di inizializzare l'uso di un'applicazione delle funzioni WinINet[3]. La funzione “push offset szAgent”, sottolineata in viola, specifica il client sofware utilizzato dal malware per la connessione ad internet, che dal commento sappiamo essere Internet Explorer 8.0.

Nel riquadro blu invece abbiamo la chiamata di funzione che apre una risorsa specificata da un URL FTP o HTTP completo[4]. In questo caso la risorsa URL è indicata dall’ “offset szUrl ; http://www.malware12.com” che viene pushato sulla cima dello stack.

L'istruzione LEA (load effective address) inserisce l'indirizzo specificato dal suo primo operando nel registro specificato dal suo secondo operando. Si noti che il contenuto della posizione di memoria non viene caricato, solo l'indirizzo effettivo viene calcolato e inserito nel registro. Questo è utile per ottenere un puntatore in una regione di memoria o per eseguire semplici operazioni aritmetiche e LEA non altera le flag[5].

Questa istruzione copia l'effettivo valore esadecimale a 16 bit di una etichetta, passata come operando sorgente, nel registro di Offset indicato dall'operando destinazione.
Il registro coinvolto per ricevere l'offset del puntatore associato all'etichetta può essere uno qualunque dei registri a 16 bit (naturalmente esclusi quelli di segmento...)[6]

3 https://learn.microsoft.com/it-it/windows/win32/api/wininet/nf-wininet-internetopena
4 https://learn.microsoft.com/it-it/windows/win32/api/wininet/nf-wininet-internetopenurla
5 https://flint.cs.yale.edu/cs421/papers/x86-asm/asm.html
6 http://www.giobe2000.it/Tutorial/Schede/07-IstruzioniCpu/LEA.asp