Kraków, 8.06..23

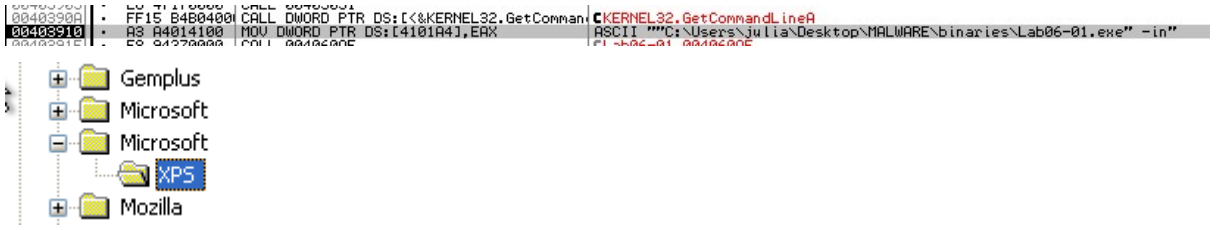
Laboratorium 5 - Zaawansowana analiza dynamiczna

Julia Sadecka, Cyberbezpieczeństwo

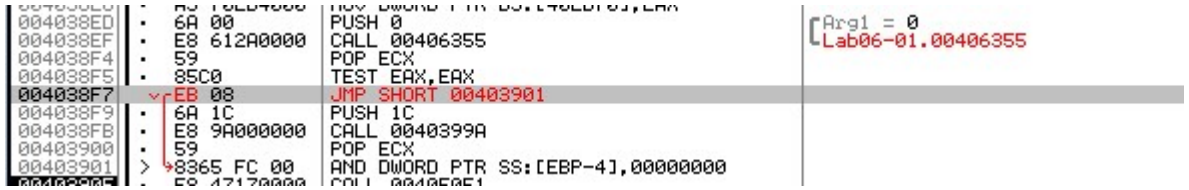
Laboratorium 5.1

1. Zmuszenie malware do instalacji

Aby malware się zainstalowało można albo wprowadzić argument '-in' i wpisać hasło. Takim sposobem możemy zobaczyć, że w rejestrach malware stworzyło nam w SOFTWARE folder Microsoft z folderem XPS.

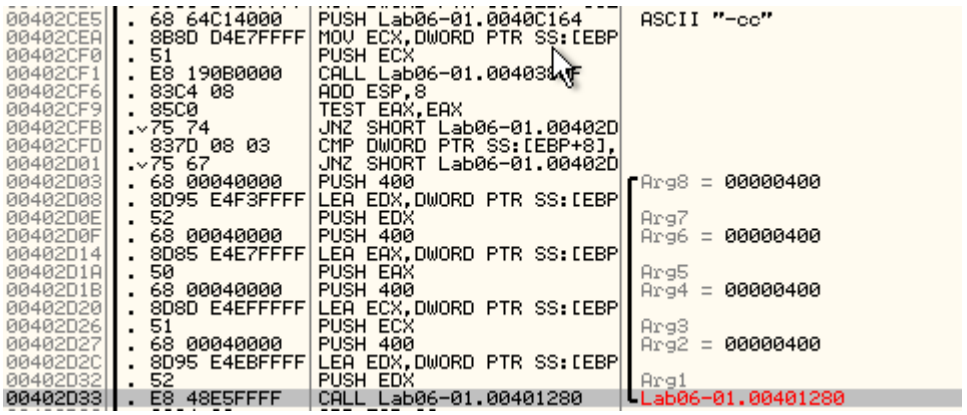


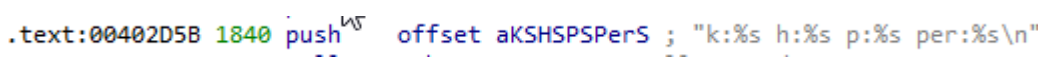
Można też gdy nie zna się hasła zmienić plik binarny np. zamiast JNZ wprowadzić JNB, aby przeskoczyło odrazu do wprowadzanego rejestru

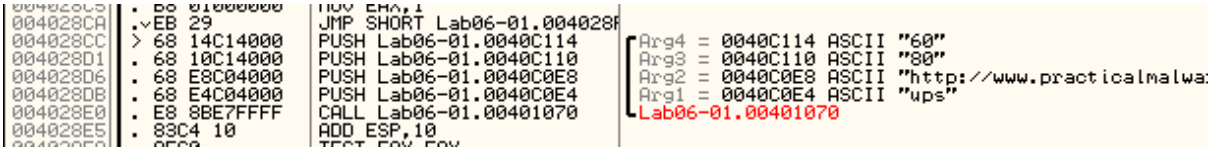


2. Argumenty wiersza poleceń

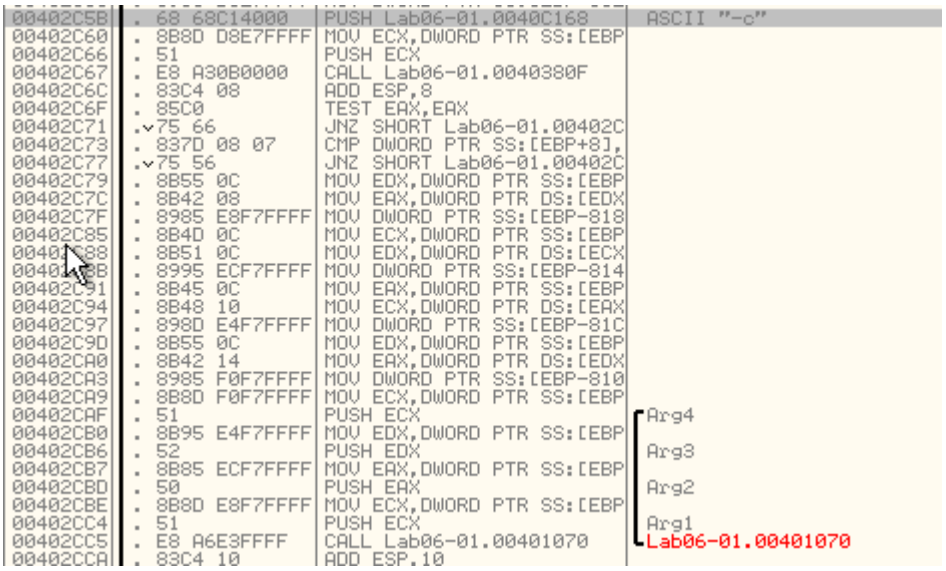
Funkcja -cc posiada 8 argumentów, które następnie wypisuje w postaci: "k: h: p: pre: ". Są to informacje o konfiguracji:
k=ups
h=host:http://www.practicalmalwareanalysis.com
p=port:80
per=60



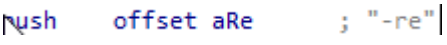




Funkcja -c pobiera 4 argumenty. Jeżeli nie istnieje to funkcja tworzy klucz rejestru.



Funkcja -re usuwa usługę, a także wpis w rejestrze



3. Zmiany w malware

Tak jak w punkcie pierwszym można zmienić funkcję JNB (jump if not equal) na JMP (jump)

4. Indykatory hostowe

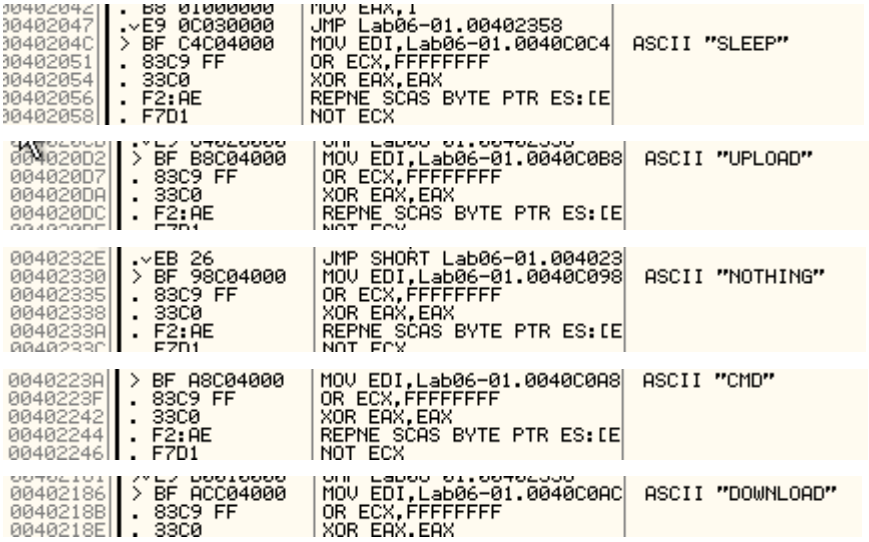
Klucz rejestru który jest tworzony:

0000C040	ASCII	SOFTWARE\Microsoft \XPS

5. Wykorzystanie przy pomocy Internetu

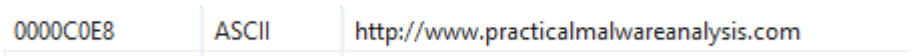
Oprogramowanie może dostać instrukcję przez sieć, aby zrobić którąś z poniższych funkcji:

- SLEEP - przez określony czas malware nic nie wykonuje
- UPLOAD - odczytuje plik z sieci
- NOTHING - malware dostaje informacje, że ma nic nie robić
- CMD - otwiera terminal wiersza poleceń
- DOWNLOAD - wysyła przez sieć plik do lokalnego hosta



6. Wskaźniki sieciowe

- Strona www.practicalmalwareanalysis.com



- Malware łączy się przez port 80

Laboratorium 5.2

1. Łańcuchy znaków

Jesteśmy w stanie odnaleźć dwa zaciemnione łańcuchy znaków. Znajduje się także string cmd

```
cmd
```

```
31h ; '1'
71h ; 'q'
61h ; 'a'
7Ah ; 'z'
32h ; '2'
77h ; 'w'
73h ; 's'
78h ; 'x'
33h ; '3'
65h ; 'e'
64h ; 'd'
63h ; 'c'
0
; 'o'
63h ; 'c'
6Ch ; 'l'
2Eh ; '.'
65h ; 'e'
78h ; 'x'
65h ; 'e'
0
```

1qaz2wsx3edc
ocl.exe

2. Wyniki programu

Funkcja najpierw używa funkcji GetVersion. Następnie pobiera ścieżkę do uruchomionego pliku i zapisuje ją w rejestrach.

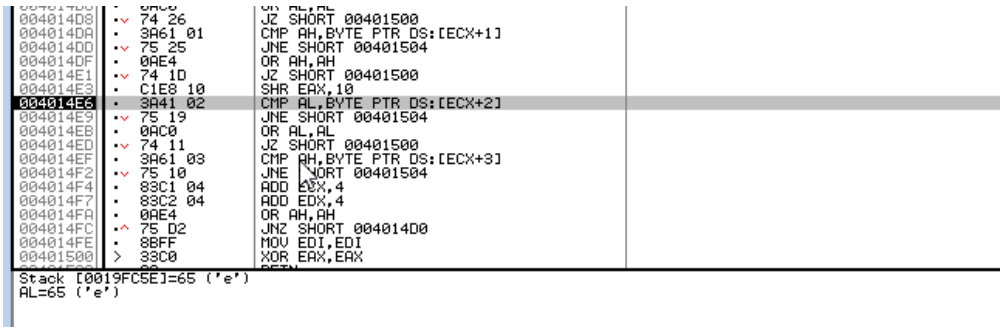
```
Registers (FPU)
EAX 005F40F8 ASCII ""C:\Users\julia\Desktop\MALWARE\binaries\La
```

Odczytuje końcówkę ścieżki i porównuje ją z ocl.exe. Jeżeli nie są takie same kończy swoje działanie.

```
EAX 00000001
ECX 0019FC58 ASCII "Lab06-02.exe"
EDX 0019FD90 ASCII "ocl.exe"
EBX 0039F000
```

3. Uruchomienie szkodliwej zawartości

Aby plik uruchomił szkodliwą zawartość należy zatem zmienić jego nazwę na ocl.exe. Program porówna nazwę, a następnie zacznie wykonywać dalej program

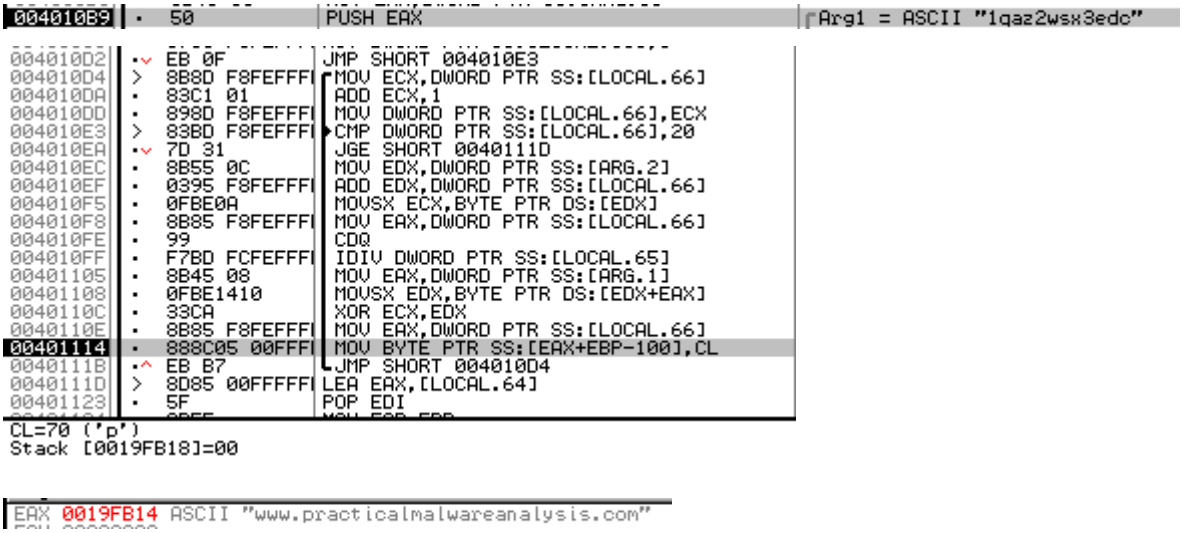


4. Działanie pod adresem 0x00401133

   Znajduje się tu zaciemniony string 1qaz2wsx3edc. Zaciemniony jest ponieważ jest zapisany znak po znaku (screen z podpunktu 1).

5. Argumenty 0x00401089

Jest wprowadzony ciąg znaków z adresu 0x00401133 i wskaźnik do bufora danych. Następnie ten ciąg jest XORowany kilka razy. Po tym powstaje domena www.practicalmalwareanalysis.com



6. Domena

   Malware wykorzystuje domenę www.practicalmalwareanalysis.com

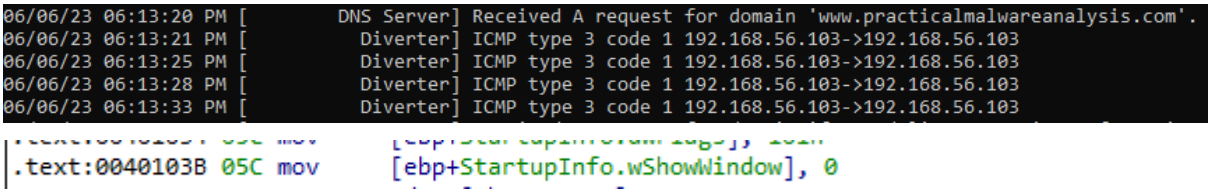
7. Procedura kodowania

   Do zaciemnienia domeny została zastosowana operacja XOR z łańcuchem 1qaz2wsx3edc.

8. CreateProcessA

   Proces ten najpierw szuka adresu IP domeny (www.practicalmalwareanalysis.com). Włącza command line (nie pokazując

okienka na monitorze) i następnie pobierając instrukcje ze strony wykonuje je w cmd.

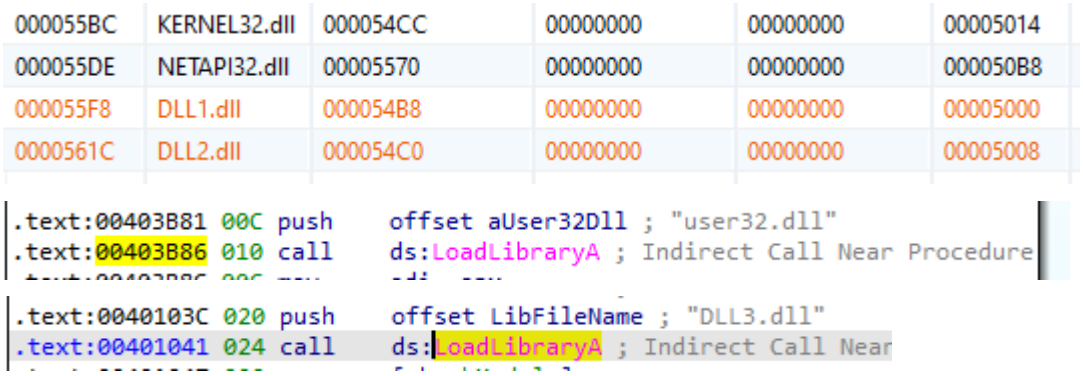


8. Opisz znaczenie wywołania CreateProcessA znajdującego się pod adresem 0x0040106E w nawiązaniu do tego malware?

Laboratorium 5.3

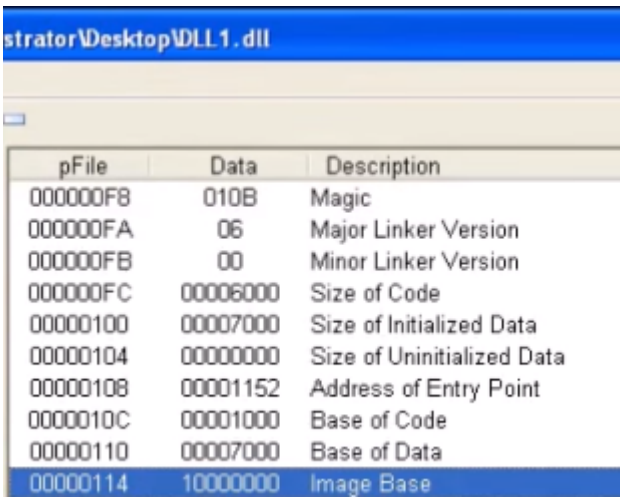
1. Biblioteki dll

   Ładują się 4 biblioteki dll (program PPEE). Dynamicznie ładuje się biblioteka user32.dll



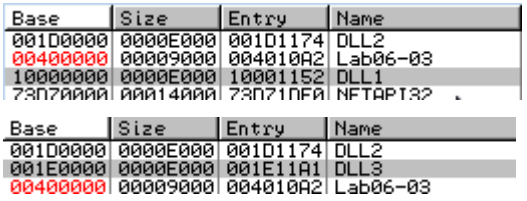
2. Wymagany adres bazowy

   Adres bazowy wymagany przez biblioteki DLL1.dll, DLL2.dll i DLL3.dll to 0x10000000



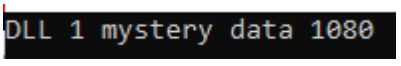
3. Przypisany adres bazowy

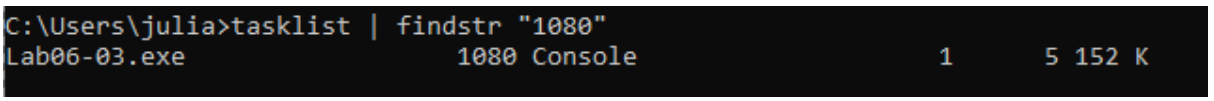
DLL1 - 10000000
DLL2 - 001D0000
DLL3 - 001E0000



4. Działanie funkcji z DLL1.dll

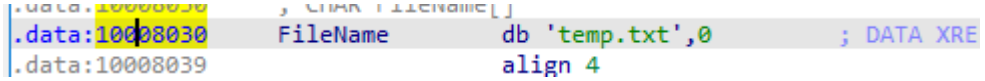
Biblioteka DD1.dll ma dwie funkcje: DLL1Print i DLLMain. Ta pierwsza pokaże napis “DLL 1 mystery data” a także numer procesu. Po sprawdzeniu rzeczywiście taki proces istnieje .





5. Nazwa pliku

Funkcja WriteFile wykorzystuję nazwę pliku temp.txt



6. Dane drugiego parametru NetScheduleJobAdd

Dane do drugiego parametru z funkcji NetScheduleJobAdd pobiera się z DLL3.DLL3GetStructure.

7. Mystery Data

Po wykonaniu program wyświetla w terminalu 3 informacje Mystery Data. Razem z nimi są identyfikatory PID procesów.

```
DLL 1 mystery data 3500
DLL 2 mystery data 232
DLL 3 mystery data 2011328
```

DLL1 jest związane z identyfikatorem bieżącego procesu

```
C:\Users\julia>tasklist | findstr "1080"
Lab06-03.exe                  1080 Console                    1      5 152 K
```

DLL 2 jest związany z plikiem temp.txt

```
.data:10008030 FileName        db 'temp.txt',0         ; DATA XRE
.data:10008039                 align 4
```

DLL 3 jest związany z łańcuhem znaków ping www.malwareanalysisbook.com

00008038	ASCII	ping www.malwareanalysisbook.com

8. Ładowanie DLL2.dll

Aby załadować DLL2.dll do IDA, aby było to zgodne z adresem ładowania zastosowanym przez OllyDbg nalaeży w IDA wybrać Manual Load, a następnie adres bazowy tej biblioteki. W moim przypadku adresem bazowym jest 001D0000.