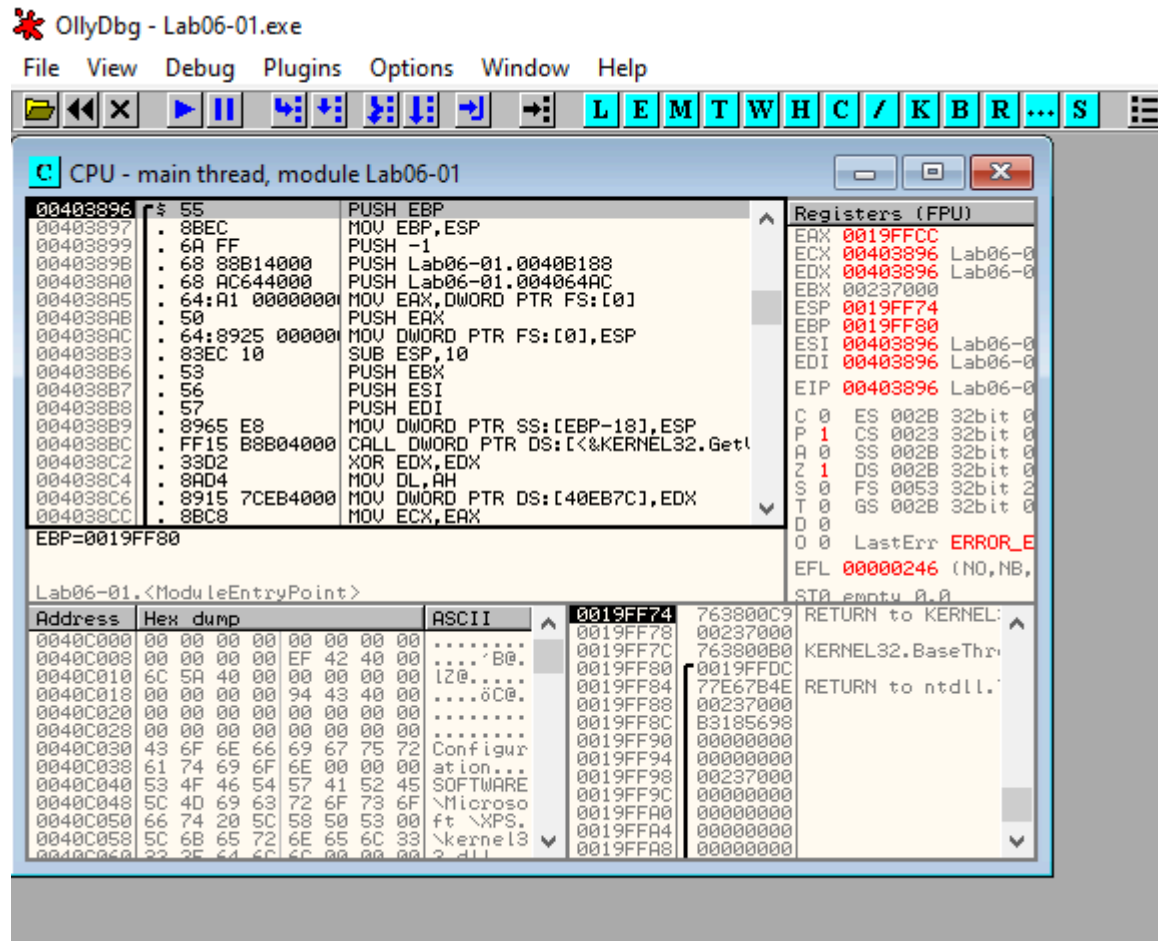


5.1

1)



Należy wpisać -in <hasło>, aby zmusić program do instalacji.

Malware zostanie uruchomiony po restarcie systemu.

2)

76130000	00024000	76137560	GDI32	10.0.19041.2913	C:\Windows\System32\GDI32.dll
761C0000	0019C000	761FD160	USER32	10.0.19041.1	C:\Windows\System32\USER32.dll
76360000	000F0000	7637FC70	KERNEL32	10.0.19041.2788	C:\Windows\System32\KERNEL32.DLL
765F0000	0023A000	7670D7F0	KERNELBA	10.0.19041.2788	C:\Windows\System32\KERNELBASE.dll
76830000	0007B000	76847800	msvcrt_wi	10.0.19041.789	C:\Windows\System32\msvcrt_win.dll
768B0000	000E5000	76911C80	adi32ful	10.0.19041.2965	C:\Windows\System32\adi32full.dll

Argumenty wiersza poleceń:

-in:

Tworzy usługę, Tworzy klucz rejestru, kopiuje plik

-re:

Odinstaluje usługę, usunie plik

-c:

Aktualizuje konfigurację. Tworzy klucz rejestru, jeśli go nie ma

-cc:

Drukuje konfigurację

Hasło musi być podane jako argument dla opcji -in w formie: -in <hasło>. Oznacza to, że aby zainstalować program, użytkownik musi znać poprawne hasło.

3)

Aby usunąć wymaganie hasła, można zmienić skok warunkowy na skok bezwarunkowy w adresie 00402B61.

Instrukcja dla użytkownika jak zrobić to w OllyDbg:

Znajdź adres 00402B61:

W głównym oknie OllyDbg użyj skrótu Ctrl+G (Go to expression) i wpisz 00402B61, aby przejść do konkretnego adresu.

Identyfikacja instrukcji skoku warunkowego:

Na adresie 00402B61 powinna znajdować się instrukcja skoku warunkowego.

00402B5F	. 85C0	TEST EAX
00402B61	. 75 64	JNZ SHOF
00402B63	. 837D 08 03	CMP DWORD
00402B65	. 75 64	JNZ SHOF

Zmiana instrukcji:

Kliknij prawym przyciskiem myszy na instrukcji skoku warunkowego i wybierz „Assemble”.

W oknie, które się pojawi, zamień bieżącą instrukcję na JMP, co odpowiada skokowi bezwarunkowemu.

Zatwierdź zmiany, klikając „Assemble” w oknie dialogowym.

Zapisz zmiany:

Po wprowadzeniu zmian, kliknij prawym przyciskiem myszy w głównym oknie OllyDbg i wybierz „Copy to executable” -> „All modifications”.

W nowym oknie wybierz „Copy all”.

Pojawi się okno z edytowanym plikiem, gdzie należy ponownie kliknąć prawym przyciskiem myszy i wybrać „Save file”.

4)

Klucz rejestru:

Ścieżka: HKLM\SOFTWARE\Microsoft\XPS

Obecność tego klucza rejestru może wskazywać na instalację złośliwego oprogramowania lub jego konfigurację.

5)

Działania jakie może wykonywać złośliwe oprogramowanie za pomocą sieci:

SLEEP: Uśpienie na X sekund.

UPLOAD: Przesyłanie pliku do hosta.

DOWNLOAD: Pobieranie pliku z hosta.

CMD: Wykonanie polecenia na hoście.

Po uruchomieniu usługi:

- Zapytanie DNS: www.practicalmalwareanalysis.com
- Połączenie przez port 80.
- Żądania GET

6)

Klasyczny dla badanych wcześniej wirusów URL -

<http://www.practicalmalwareanalysis.com>

5.2

1)

Przykładowe łańcuchy znaków w tym pliku:

	Offset	Type	Length	String
37	4288	A	43	R6016 - not enough space for thread data
38	42b6	A	30	abnormal program termination
39	42d8	A	43	R6009 - not enough space for environment
40	4304	A	41	R6008 - not enough space for arguments
41	4330	A	36	R6002 - floating point not loaded
42	4358	A	36	Microsoft Visual C++ Runtime Library
43	4384	A	25	Runtime Error! Program:
44	43a4	A	22	<program name unknown>
45	43bc	A	18	GetLastActivePopup
46	43d0	A	15	GetActiveWindow
47	43e0	A	11	MessageBoxA
48	43ec	A	10	user32.dll
49	451e	A	19	WaitForSingleObject

Łańcuchy w pliku wskazują na obecność komunikatów błędów wykonawczych i wywołań funkcji Windows API, co może sugerować, że plik jest aplikacją napisaną w języku C++ i korzystającą z Microsoft Visual C++ Runtime Library.

1. **Komunikaty błędów (np. R6016, R6009, R6008, R6002):** Typowe błędy środowiska uruchomieniowego Visual C++, które mogą wskazywać na problemy z alokacją zasobów lub argumentami.
2. **Biblioteka runtime - Microsoft Visual C++ Runtime Library** – biblioteka niezbędna do działania aplikacji napisanych w C++.
3. **Funkcje Windows API:**
 - GetLastActivePopup, GetActiveWindow, MessageBoxA, WaitForSingleObject - funkcje związane z obsługą okien i synchronizacją w systemie Windows.
4. **Plik DLL: user32.dll** – biblioteka zawierająca funkcje do zarządzania interfejsem użytkownika.

Obecność tych łańcuchów może sugerować, że plik to aplikacja wykorzystująca Windows API i runtime Visual C++.

2)

Na początku otwiera się proces cmd.exe, który umożliwia zdalne wykonywanie poleceń poprzez sieć.

Malware wysyła zapytanie DNS do domeny **practicalmalwareanalysis.com** w celu uzyskania adresu IP.

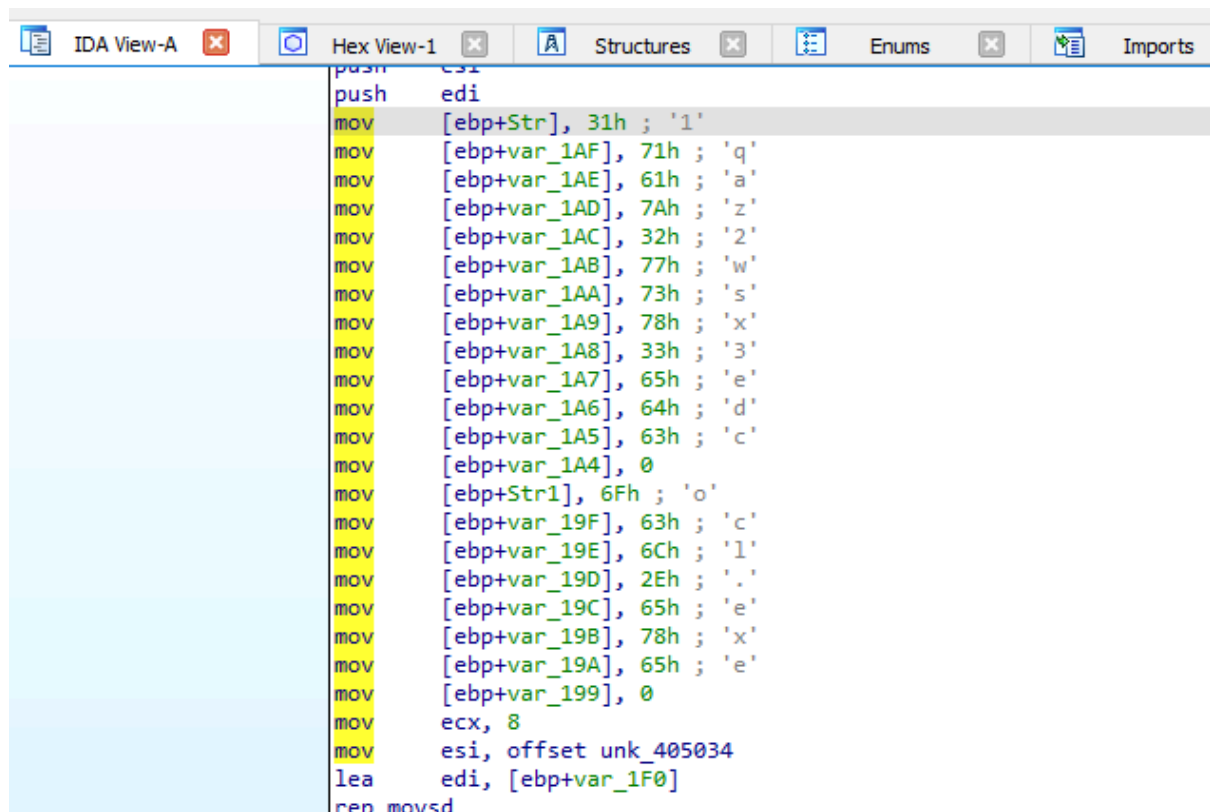
Funkcja **gethostbyname** zwraca strukturę **hostent**, z której oprogramowanie pobiera adres IP.

Program próbuje połączyć się z **practicalmalwareanalysis.com** na porcie TCP.

3)

Trzeba zmienić nazwę pliku na ocl.exe.

4)



The screenshot shows the Hex View window of IDA Pro. The assembly code is as follows:

```
push    esi
push    edi
mov     [ebp+Str], 31h ; '1'
mov     [ebp+var_1AF], 71h ; 'q'
mov     [ebp+var_1AE], 61h ; 'a'
mov     [ebp+var_1AD], 7Ah ; 'z'
mov     [ebp+var_1AC], 32h ; '2'
mov     [ebp+var_1AB], 77h ; 'w'
mov     [ebp+var_1AA], 73h ; 's'
mov     [ebp+var_1A9], 78h ; 'x'
mov     [ebp+var_1A8], 33h ; '3'
mov     [ebp+var_1A7], 65h ; 'e'
mov     [ebp+var_1A6], 64h ; 'd'
mov     [ebp+var_1A5], 63h ; 'c'
mov     [ebp+var_1A4], 0
mov     [ebp+Str1], 6Fh ; 'o'
mov     [ebp+var_19F], 63h ; 'c'
mov     [ebp+var_19E], 6Ch ; 'l'
mov     [ebp+var_19D], 2Eh ; '.'
mov     [ebp+var_19C], 65h ; 'e'
mov     [ebp+var_19B], 78h ; 'x'
mov     [ebp+var_19A], 65h ; 'e'
mov     [ebp+var_199], 0
mov     ecx, 8
mov     esi, offset unk_405034
lea     edi, [ebp+var_1F0]
rep movsd
```

Przypisywany jest ciąg znaków **1qaz2wsx3edc** bajt po bajcie. Dane te są zapisywane w pamięci, a następnie prawdopodobnie używane do deobfuskacji nazwy domeny.

5)

Przekazywany jest string **1qaz2wsz3edc**.

6)

www.practicalmalwareanalysis.com

7)

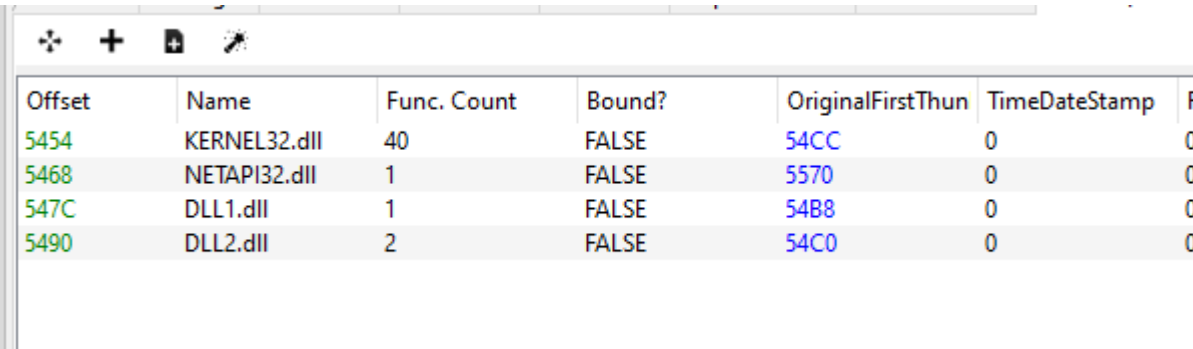
Jest to XOR.

8)

Wywołanie **CreateProcessA** pod adresem 0x0040106E w tym malware uruchamia instancję **cmd.exe**. Konfigurując wejście, wyjście i błędy standardowe do połączenia z gniazdem sieciowym (socket). W ten sposób tworzy odwróconą powłokę (reverse shell), która jest połączona z tym gniazdem, umożliwiając zdalne wykonywanie poleceń przez atakującego.

5.3

1)



Offset	Name	Func. Count	Bound?	OriginalFirstThun	TimeDateStamp	F
5454	KERNEL32.dll	40	FALSE	54CC	0	0
5468	NETAPI32.dll	1	FALSE	5570	0	0
547C	DLL1.dll	1	FALSE	54B8	0	0
5490	DLL2.dll	2	FALSE	54C0	0	0

Natomiast dynamicznie ładowane są biblioteki(za pomocą funkcji LoadLibraryA):

- USER32
- DLL3

2)

Wszystkie biblioteki DLL wymagają tego samego adresu bazowego: **0x10000000**.

3)

DLL1 - 10000000

DLL2 - 00330000

DLL3 - 00390000

4)

Importowana funkcja z **DLL1.dll** printuje identyfikator procesu (PID). Funkcja **DLLMain** w **DLL1.dll** pobiera PID bieżącego procesu i przechowuje go w pamięci DLL. Następnie funkcja **DLL1Print** printuje ten PID jako „mystery data” w konsoli. Każde uruchomienie programu generuje inny PID, dlatego liczba ta zmienia się przy każdym uruchomieniu.

5)

Jest to **temp.txt**

```
|push    offset FileName ; "temp.txt"
```

6)

Dane dla drugiego parametru funkcji **NetScheduleJobAdd** są pobierane z wyniku wywołanej funkcji **DLL3.DLL3GetStructure**.

7)

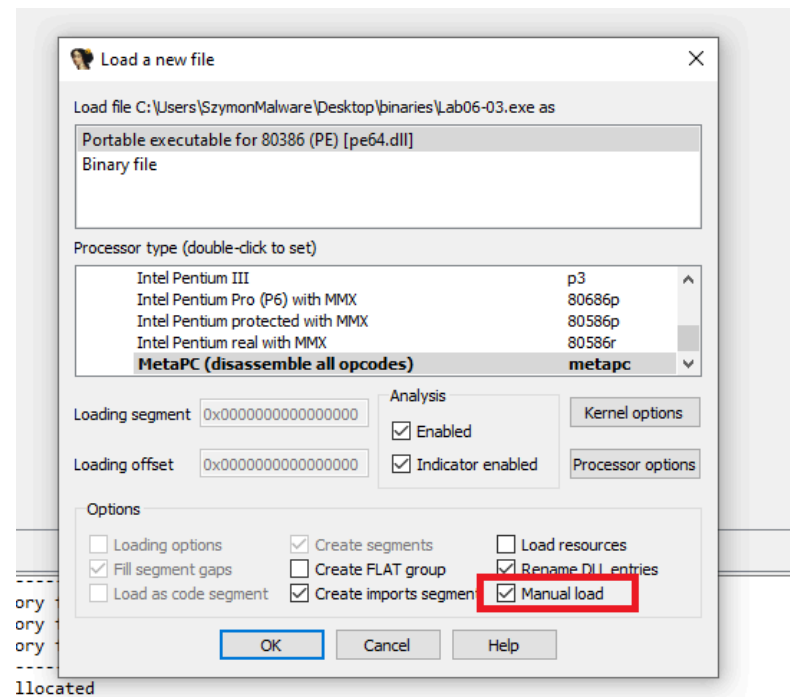
DLL1: PID

DLL2: uchwyt pliku

DLL3: Ciąg znaków Unicode jako **liczba całkowita**.

8)

Wybrać opcję Manual load.



Files\IDA Freeware 8.3\procs\pc64.dll for metapc...Initializing processor mod

Następnie należy podać ustalony adres bazowy.