

# Laboratorium 2

---

Na laboratorium 2 przedstawione zostają proste programy do analizy statycznej.

Programy pokazują w plikach podstawowe informacje takie jak:

- Sumy kontrolne
- Importy
- Zaciemnianie
- Stringi

## Spis treści

- [Laboratorium 2](#)
  - [Spis treści](#)
  - [Sumy kontrolne](#)
  - [Laboratorium 1.1](#)
    - [Plik Lab02-01.dll](#)
    - [Plik Lab02-01.exe](#)
    - [Podsumowanie 01.exe](#)
  - [Laboratorium 1.2](#)
    - [Plik Lab02-02.exe](#)
  - [Laboratorium 1.3](#)
    - [Plik Lab02-03.exe](#)
  - [Laboratorium 1.4](#)
    - [Plik Lab02-04.exe](#)
    - [Podsumowanie 04.exe](#)

## Sumy kontrolne

File Name	MD5	SHA-256
Lab02-01.dll	290934C61DE9176AD682FFDD65F0A669	F50E42C8DFAAB649BDE0398867E930B86C2A599E8DB83B8260393082268F2DBA
Lab02-01.exe	BB7425B82141A1C0F7D60E5106676BB1	58898BD42C5BD3BF9B1389F0EEE5B39CD59180E8370EB9EA838A0B327BD6FE47
Lab02-02.exe	8363436878404DA0AE3E46991E355B83	C876A332D7DD8DA331CB8EEE7AB7BF32752834D4B2B54EAA362674A2A48F64A6
Lab02-03.exe	9C5C27494C28ED0B14853B346B113145	7983A582939924C70E3DA2DA80FD3352EBC90DE7B8C4C427D484FF4F050F0AEC
Lab02-04.exe	625AC05FD47ADC3C63700C3B30DE79AB	0FA1498340FCA6C562CFA389AD3E93395F44C72FD128D7BA08579A69AAF3B126

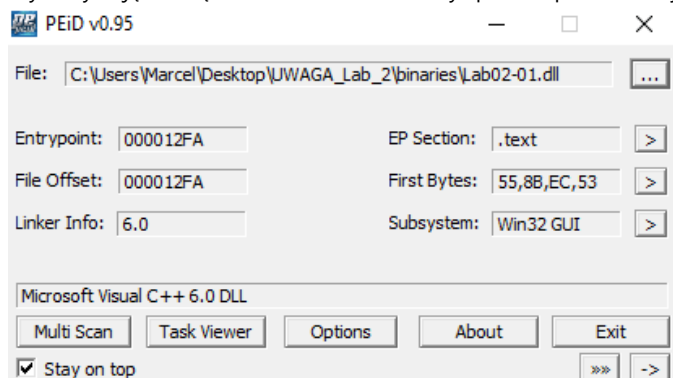
## Laboratorium 1.1

---

### Plik Lab02-01.dll

1. Na VirusShare plik został oznaczony 44 razy jako wirus i był już wcześniej analizowany
2. Plik został skompilowany 19 Grudnia 2010 roku

3. Wykorzystując narzędzie PEiD można w łatwy sposób sprawdzić czy program został zaciemniony

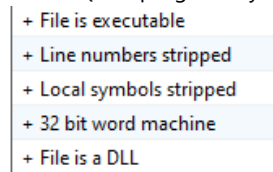


I jak widać nie został on zaciemniony, jest to zwykła biblioteka DLL.

Kiedy obejrzy się ten sam program w PPEE

znalezienie informacji o tym, że jest to biblioteka DLL nie jest takie oczywiste.

Na szczęście program wyświetla nam tę informację w File Header



Zrobiłem research, skąd program wie, że jest to biblioteka i natknąłem się na flagi *characteristics*

Ten plik ma characteristics równe 210E. Po przeczytaniu dokumentacji [Microsoft PE32 File](#) zauważyłem,

że flaga 0x2000 oznacza plik DLL.

Characteristics	210E
-----------------	------

4. Biblioteka importuje funkcje z 3 bibliotek:

- **KERNEL32.DLL**

Biblioteka odpowiada za zarządzanie pamięcią, I/O i tworzenie procesów.

Do ciekawych importów należą funkcje m. in.

- CreateProcessA (*tworzy proces*)
- CreateMutexA (*upewnia się, że tylko jedna instancja programu działa w danym momencie*)
- OpenMutexA (*podobnie jak poprzednia funkcja*)

- **WS2\_32.DLL**

Biblioteka odpowiada za połączenie sieciowe TCP/IP i wspiera przy tym m. in. IPv4, IPv6 oraz Bluetooth.

Niestety PE-bear nie pokazuje nazw importów.

WS2_32.dll [ 10 entries ]		
Call via	Name	Ordinal
2030	-	17
2034	-	73
2038	-	B
203C	-	4
2040	-	13
2044	-	16

Widać jednak Ordinal, czyli numer połączony z nazwą. Nie znalazłem tych liczb w dokumentacji,

ale natknąłem się na [skrypt w pythonie](#), który ma wypisane i Ordinals jak i nazwy funkcji.

Po przekonwertowaniu liczb z hex na dec uzyskałem nazwy importów.

- socket
- inet\_addr (*konwertuje adres IP na format do użycia przez funkcję connect*)
- connect (*łączy z adresem IP*)
- send (*wysyła pakiet*)
- recv (*odbiera pakiet*)

Po fakcie zauważyłem, że PPEE automatycznie wyświetla nazwy importowanych funkcji

OFT	FT	Hint	Name	Ordinal
80000017	80000017	N/A	N/A	00000017 (socket)
80000073	80000073	N/A	N/A	00000073 (WSAStartup)
8000000B	8000000B	N/A	N/A	0000000B (inet_addr)
80000004	80000004	N/A	N/A	00000004 (connect)
80000013	80000013	N/A	N/A	00000013 (send)
80000016	80000016	N/A	N/A	00000016 (shutdown)
80000010	80000010	N/A	N/A	00000010 (recv)
80000003	80000003	N/A	N/A	00000003 (closesocket)
80000074	80000074	N/A	N/A	00000074 (WSACleanup)
80000009	80000009	N/A	N/A	00000009 (htons)

- **MSVCRT.DLL**

Jest to C Standard Library dla Visual C++. Niegroźne

5. Jak wyżej

6. Opisane poniżej

7. Po obejrzeniu pliku w PPEE, znalazłem String, który jest adresem IPv4

```
00026028      127.26.152.13
```

8. Podsumowanie

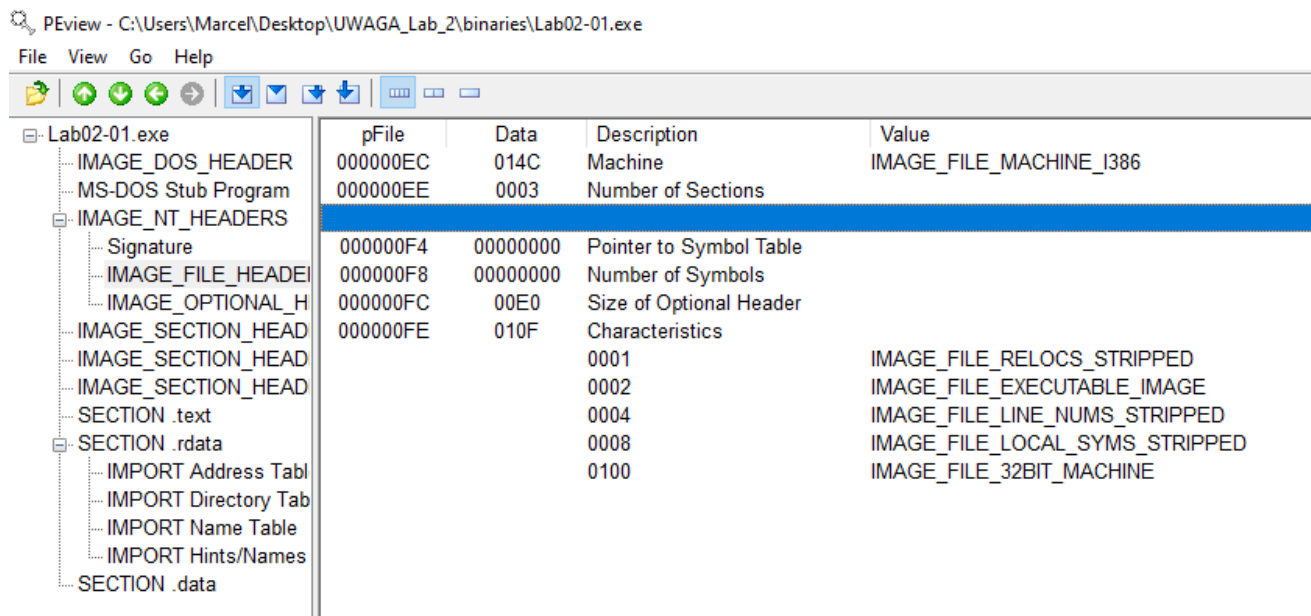
Plik Lab02-01.exe

1. Na VirusShare plik został oznaczony 50 razy jako wirus i był już wcześniej analizowany

2. Nie udało mi się znaleźć daty skompilowania. W jednym miejscu wyskakiwał błąd. Po sprawdzeniu dla innego pliku znalazłem datę

PEview - C:\Users\Marcel\Desktop\UWAGA\_Lab\_2\binaries\Lab02-02.exe

	pFile	Data	Description	Value
Lab02-02.exe				
IMAGE_DOS_HEADER	000000E4	014C	Machine	IMAGE_FILE_MACHINE_I386
MS-DOS Stub Program	000000E6	0003	Number of Sections	
IMAGE_NT_HEADERS	000000E8	4D370D01	Time Date Stamp	2011/01/19 oer. 16:10:41 UTC
Signature	000000EC	00000000	Pointer to Symbol Table	
IMAGE_FILE_HEADER	000000F0	00000000	Number of Symbols	
IMAGE_OPTIONAL_HEADER	000000F4	00E0	Size of Optional Header	
IMAGE_SECTION_HEADER	000000F6	010F	Characteristics	
IMAGE_SECTION_HEADER		0001		IMAGE_FILE_RELOCS_STRIPPED
IMAGE_SECTION_HEADER		0002		IMAGE_FILE_EXECUTABLE_IMAGE
SECTION UPX0		0004		IMAGE_FILE_LINE_NUMS_STRIPPED
SECTION UPX1		0008		IMAGE_FILE_LOCAL_SYMS_STRIPPED
SECTION UPX2		0100		IMAGE_FILE_32BIT_MACHINE



Powinna być ona w tym miejscu

Jednak tu się niestety nie wyświetla

3. PEE pokazuje, że ten program również nie został zaciemniony.

Jest to zwykły plik exe, gdyż flaga characteristics 0x2000 nie została ustawiona (0x010F)

Characteristics	010F
+ Relocation info stripped	
+ File is executable	
+ Line numbers stripped	
+ Local symbols stripped	
+ 32 bit word machine	

4. Biblioteka importuje funkcje z 2 bibliotek:

- **KERNEL32.DLL**

Biblioteka odpowiada za zarządzanie pamięcią, I/O i tworzenie procesów.

Do ciekawych importów należą funkcje m. in.

- CreateFileA (*tworzy lub otwiera plik*)
- CopyFileA (*kopiuje plik*)
- MapViewOfFile (*malware unika wtedy użycia WriteFile do zmiany zawartości pliku*)

- **MSVCRT.DLL**

Jest to C Standard Library dla Visual C++. Niegroźne

5. Opisane wcześniej

6. PEE w zakładce strings wyświetla 2 ścieżki do biblioteki,

ale jedna jest podana niepoprawnie (*małe literki oraz 'l' zamienione na '1'*)

0000304C	C:\windows\system32\kerne132.dll
00003070	Kernel32.
0000307C	Lab01-01.dll
0000308C	C:\Windows\System32\Kernel32.dll

Trzeba jednak zauważyć, że litera 'C' jest wielka, a to oznacza, że ścieżka jest poprawna

Podejrzewam, że malware może próbować zamaskować się pod podaną ścieżką

7. Opisane wyżej

## Podsumowanie 01.exe

Po przeprowadzonej analizie na obu plikach, przypuszczam, że malware próbuje się zamaskować (dziwna ścieżka, import CreateFile) oraz połączyć się ze stroną internetową, bądź zdalnym hostem.

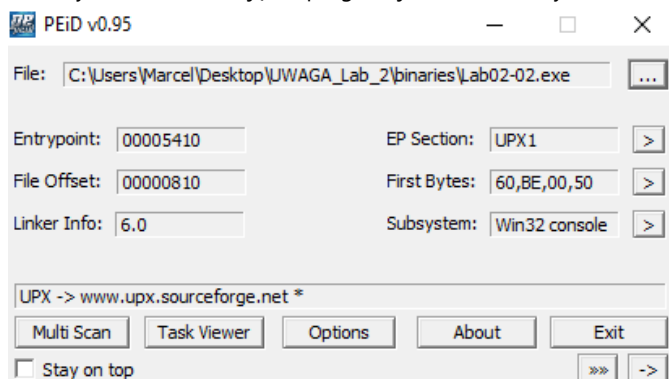
Może to być w celu pobrania pliku, bądź wymiany informacji (send, recv).

Po takiej wstępnej analizie nie jestem w stanie przypisać innych zachowań potencjalnego malware'u

## Laboratorium 1.2

Plik Lab02-02.exe

1. Na VirusShare plik został oznaczony 51 razy jako wirus i był już wcześniej analizowany
2. PEiD wyświetla informację, że program jest zaciemniony UPX



PPEE również wyświetla informacje o spakowaniu przez UPX

Name	VirtualAd...	VirtualSize	RawAddre...
UPX0	00001000	00004000	00000400
UPX1	00005000	00001000	00000400
UPX2	00006000	00001000	00000A00

3. Rozpakowałem program za pomocą UPX.

Następnie porównałem pliki przed i po rozpakowaniu w celu zobaczenia importów.

Offset	Name	Func. Count
A00	KERNEL32.DLL	6
A14	ADVAPI32.dll	1
A28	MSVCRT.dll	1
A3C	WININET.dll	1

Offset	Name	Func. Count
208C	KERNEL32.DLL	9
20A0	ADVAPI32.dll	3
20B4	MSVCRT.dll	13
20C8	WININET.dll	2

Po rozpakowaniu PE-bear pokazuje znacznie więcej importów:

- **KERNEL32.DLL**
  - OpenMutex (upewnienie się, że tylko jeden proces tego programu istnieje)
  - CreateWaitableTimer
  - GetModuleFileName (zwraca nazwę modułu, który obecnie działa, w celu zmodyfikowania lub skopiowania plików tego procesu)
- **ADVAPI32.DLL**
  - OpenSCManager (komunikuje się z Service Control Manager, konieczne przed stworzeniem usługi)
  - CreateService (tworzy usługę)
  - StartServiceCtrlDispatcher (łączy główny wątek z usługą, konieczne do działania usługi)
- **MSVCRT.DLL**
  - Nic ciekawego
- **WININET.DLL**
  - InternetOpenUrl (łączy ze stroną internetową)

4. W PPEE znalazłem stringi, które wskazują na łączenie się ze stroną internetową

<http://www.malwareanalysisbook.com>

Offset	Type	Strings recognized URL
00003030	ASCII	<a href="http://www.malwareanalysisbook.com">http://www.malwareanalysisbook.com</a>

## Laboratorium 1.3

## Plik Lab02-03.exe

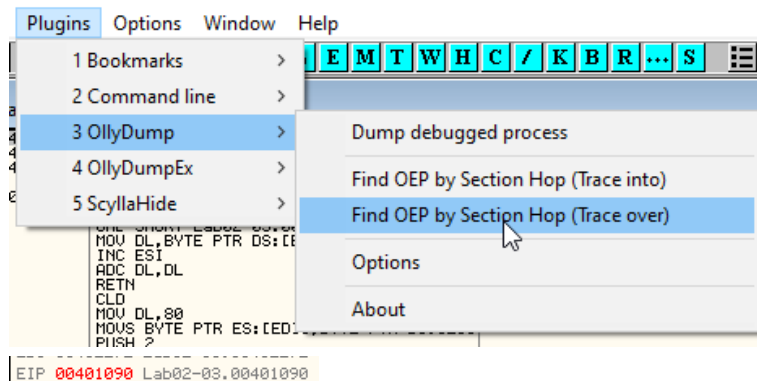
1. Na VirusShare plik został oznaczony 58 razy jako wirus i był już wcześniej analizowany
2. Program został zaciemniony za pomocą FSG 1.0.  
Programu nie da się rozpakować za pomocą UPX, ponieważ nie jest on spakowany przez UPX

FSG 1.0 -> dulek/xtl

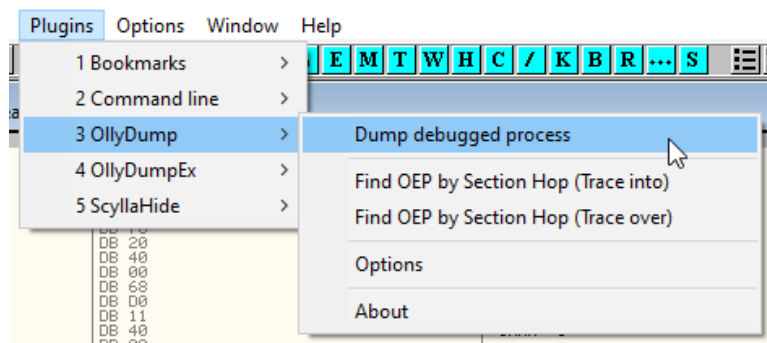
3. Daty kompilacji nie da się sprawdzić. Wyświetlana jest wartość 0x0 czyli 1 stycznia 1970
4. Rozpakowanie programu było trochę bardziej skomplikowane niż przy UPX.

Do rozpakowania użyłem OllyDbg z pluginem OllyDump.

Należało najpierw znaleźć OEP (Original Entry Point),

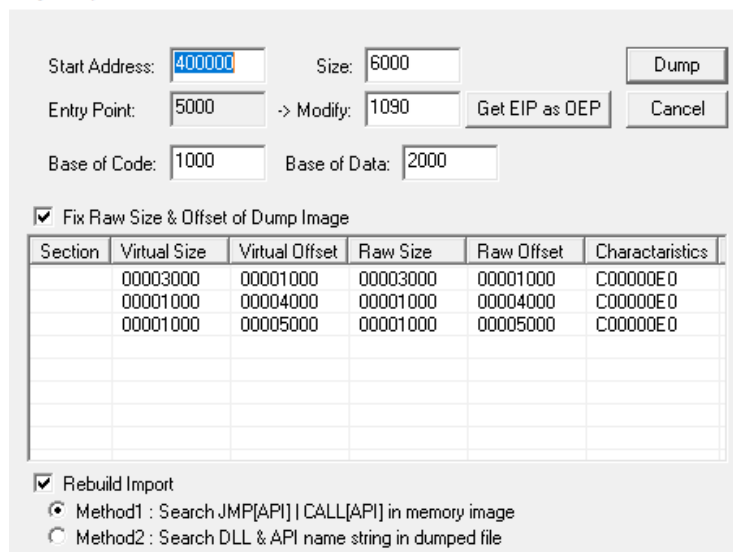


a następnie rzucić program z tego miejsca.



W tym przypadku EP został zamieniony z 5000 na 1090.

OllyDump - Lab02-03.exe



Jak widać mamy 2 programy: jeden przed, a drugi po rozpakowaniu

Lab02-03.exe

Lab02-03-fsgless.exe

W PE-bear zauważyłem różnicę w importowanych bibliotekach.

Przed rozpakowaniem była tylko biblioteka Kernel32.dll, prawdopodobnie używana przez FSG.

Po rozpakowaniu są widoczne tylko 2 biblioteki, lecz nie jestem pewny do czego te biblioteki służą:

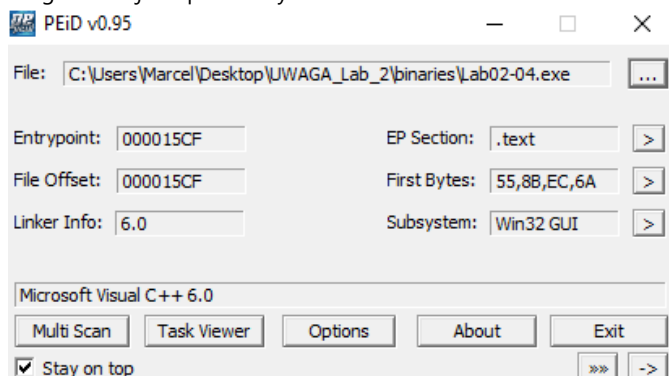
- **OLE32.DLL**
  - OleInitialize
- **OLEAUT32.DLL**
  - VariantInit
  - SysAllocString
  - SysFreeString

5. Podczas przeszukiwania stringów znalazłem adres strony,  
z którą malware może próbować się połączyć [malwareanalysisbook.com/ad.html](http://malwareanalysisbook.com/ad.html)

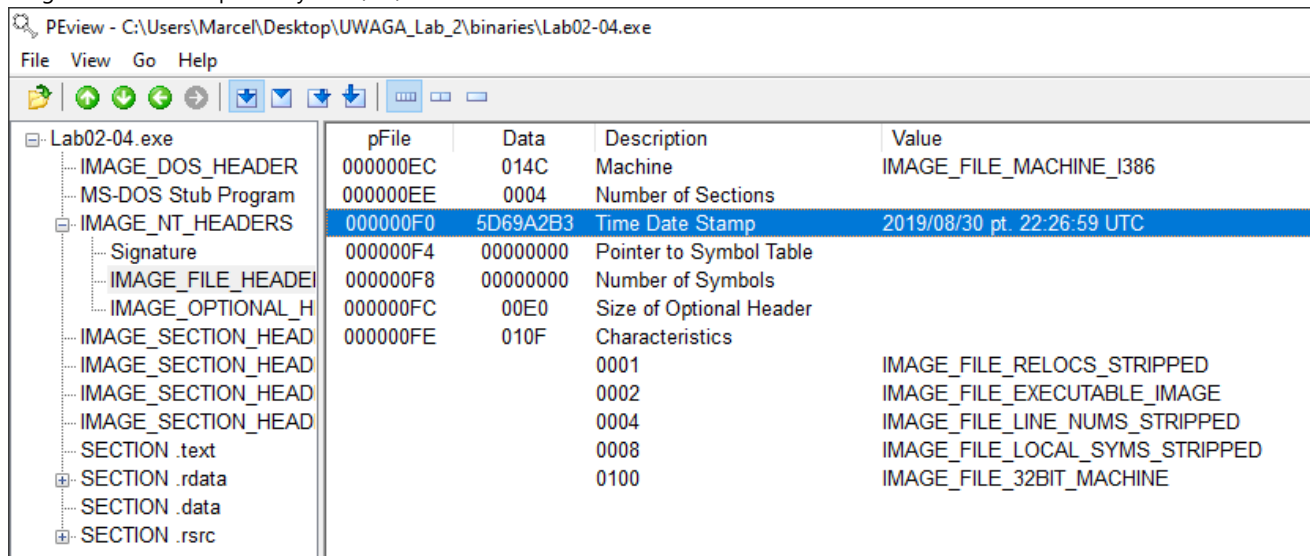
## Laboratorium 1.4

### Plik Lab02-04.exe

1. Na VirusShare plik został oznaczony 52 razy jako wirus i był już wcześniej analizowany
2. Program nie jest spakowany



3. Program został skompilowany 2019/08/30 o 22:26:59 UTC



4. Program ma sporo podejrzanych importów:

- **KERNEL32.DLL**
  - LoadLibrary (*ładuje bibliotekę*)
  - WinExec (**odpala inny program exe**)
  - WriteFile (*modyfikuje plik*)
  - CreateFile (*tworzy plik*)
  - CreateRemoteThread (*tworzy wątek w innym procesie*)
  - GetModuleHandle (*używane do modyfikowania bibliotek*)
  - GetProcAddress (*wyszukuje adres biblioteki załadowanej do pamięci*)
  - GetTempPath (*zwraca ścieżkę katalogu Temp*)
  - FindResource (*wyszukuje zasób w bibliotece*)
  - OpenProcess
- **ADVAPI32.DLL**
  - OpenProcessToken

- AdjustTokenPrivileges (*prawdopodobnie do uniesienia uprawnień*)
- **MSVCRT.DLL**
  - Nic ciekawego

5. W strings znajduje się jeszcze więcej informacji. Znajduje się tu więcej funkcji:

- SetDebugPrivilege
- URLDownloadToFile.

Widnieje też adres <http://www.practicalmalwareanalysis.com/updater.exe>

6. Program posiada importy z funkcjami sieciowymi, mimo że w PEE w sekcji import one nie widnieją

7. Po wyeksportowaniu i wrzuceniu pliku do PEE widać powyższy import URLDownloadToFile

## Podsumowanie 04.exe

Program jest zdecydowanie groźny. Importuje on wiele funkcji, które na to wskazują.

Po niezbyt podstawowej analizie statycznej mam pewne przypuszczenia co do działania wirusa.

Może stosować mechanizmy maskowania, przy wykorzystaniu code injection (FindResource, GetProcessAddress, getModuleHandle, WriteFile, CreateFile).

Wstrzykuje on groźny kod do istniejących już bibliotek, lub może tworzyć nowe.

Może robić swoje działanie na kilka plików, np. w katalogu Temp.

Z pewnością pobiera on plik ze strony (URLDownloadToFile) oraz go uruchamia (WinExec).

Podejrzewam, że wirus próbuje uzyskać uprawnienia administratora (AdjustTokenPrivileges), ale nie jestem tego pewien w 100%