

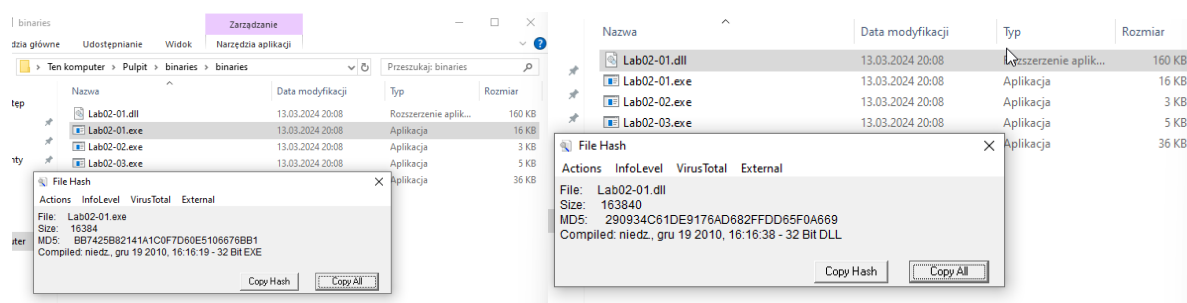
Analiza Malware Laboratorium nr 2

Raport – Nikodem Jakubowski

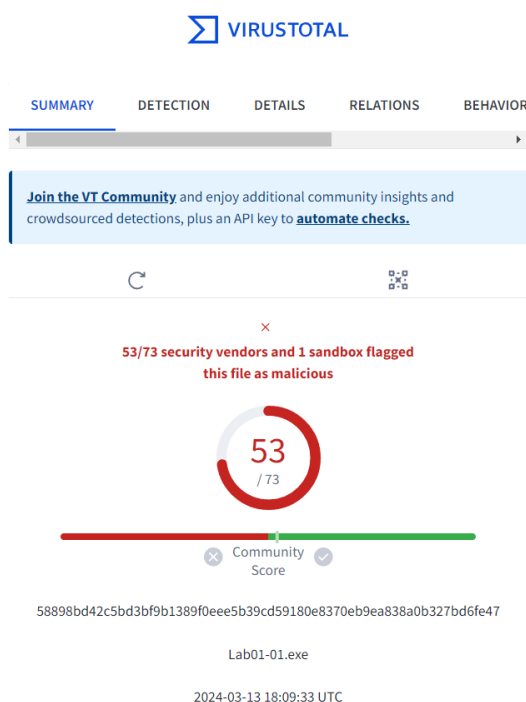
Laboratorium 1.1

W tym laboratorium wykorzystaj pliki Lab02-01.exe i Lab02-01.dll. Skorzystaj z narzędzi przeznaczonych do statycznej analizy i odpowiedz na poniższe pytania.

1. Wyciągnij hasha (np. md5 lub sha-1) z plików i sprawdź na stronie www.VirusTotal.com, czy pliki o tych samych sumach kontrolnych zostały wcześniej analizowane pod kątem szkodliwego oprogramowania?



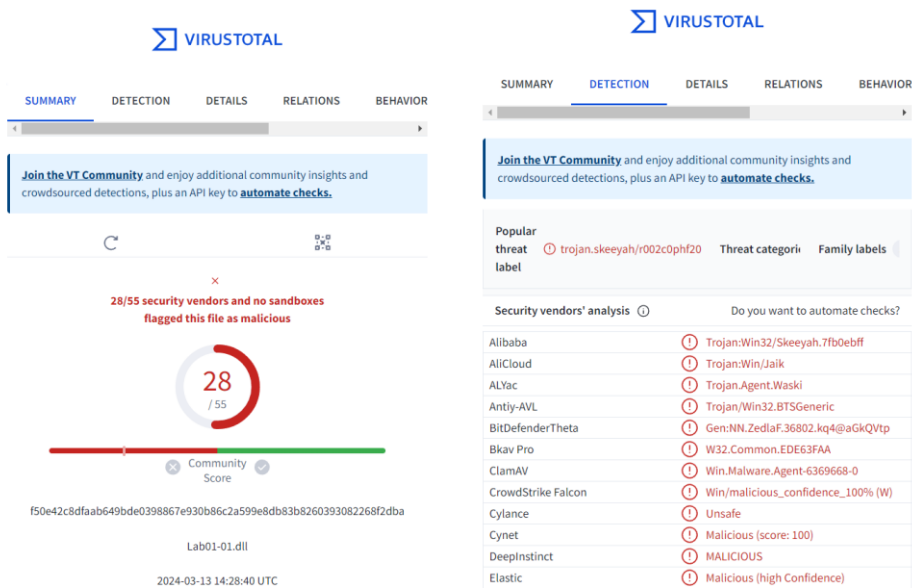
Wpis w VirusTotal dla pliku Lab02-01.exe.



The screenshot shows the VirusTotal analysis results for Lab02-01.exe. The detection status is 'Detected'.

Security Vendor	Detection
AhnLab-V3	Trojan.Win32.Agent.C957604
Alibaba	Trojan.Win32/Aenjaris.2be749b4
AliCloud	Backdoor
ALYac	Trojan.Agent.1638455
Antiy-AVL	Trojan.Win32.TSGeneric
Arcabit	Trojan.Ulise.D1BC1E
Avast	Win32:Malware-gen
AVG	Win32:Malware-gen
Avira (no cloud)	TR/Agent.kkbv
BitDefender	Gen:Variant.Ulise.113694
Bkav Pro	W32.Common.4C83E082
ClamAV	Win.Malware.Agent-6342616-0

Wpis w VirusTotal dla pliku Lab02-01.dll.



2. Wykorzystując narzędzie PView odszukaj informacje o dacie skompilowania programu.

Poniżej data stampy i ich tłumaczenie.

Member	Offset	Size	Value	Meaning
Machine	000000EC	Word	014C	Intel 386
NumberOfSections	000000EE	Word	0003	
TimeDateStamp	000000F0	Dword	4D0E2FD3	
PointerToSymbolTa...	000000F4	Dword	00000000	
NumberOfSymbols	000000F8	Dword	00000000	
SizeOfOptionalHea...	000000FC	Word	00E0	
Characteristics	000000FE	Word	010F	Click here

The current Unix epoch time is 1710513726

Convert epoch to human-readable date and vice versa

4D0E2FD3 Timestamp to Human date reset

Supports Unix timestamps in seconds, milliseconds, microseconds and nanoseconds.

Converting hexadecimal timestamp to decimal: 1292775379

Assuming that this timestamp is in seconds:

GMT: Sunday, 19 December 2010 16:16:19

Your time zone: niedziela, 19 grudnia 2010 17:16:19 GMT+01:00

Relative: 13 years ago

Member	Offset	Size	Value	Meaning
Machine	000000E4	Word	014C	Intel 386
NumberOfSections	000000E6	Word	0004	
TimeDateStamp	000000E8	Dword	4D0E2FE6	
PointerToSymbolTa...	000000EC	Dword	00000000	
NumberOfSymbols	000000F0	Dword	00000000	
SizeOfOptionalHea...	000000F4	Word	00E0	
Characteristics	000000F6	Word	210E	Click here

The current Unix epoch time is 1710513760

Convert epoch to human-readable date and vice versa

4D0E2FE6 Timestamp to Human date reset

Supports Unix timestamps in seconds, milliseconds, microseconds and nanoseconds.

Converting hexadecimal timestamp to decimal: 1292775398

Assuming that this timestamp is in seconds:

GMT: Sunday, 19 December 2010 16:16:38

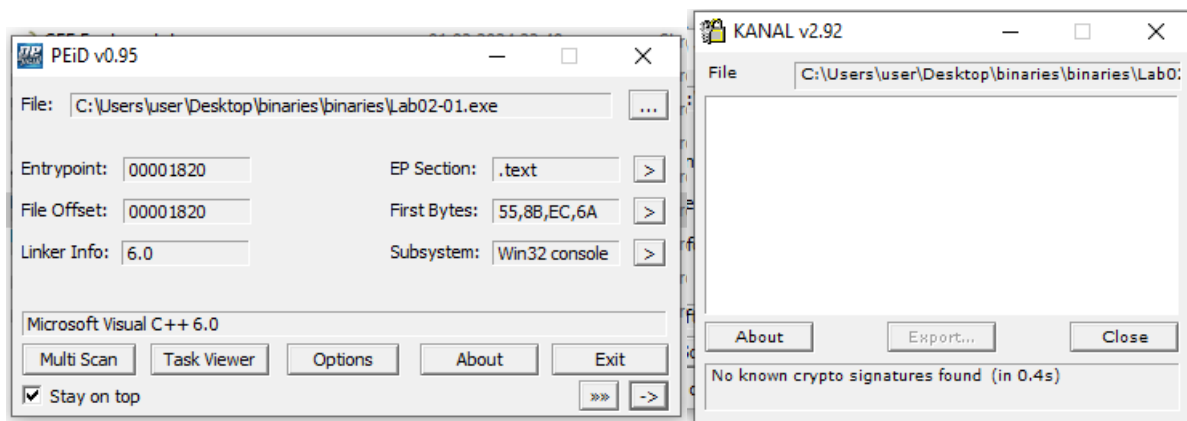
Your time zone: niedziela, 19 grudnia 2010 17:16:38 GMT+01:00

Relative: 13 years ago

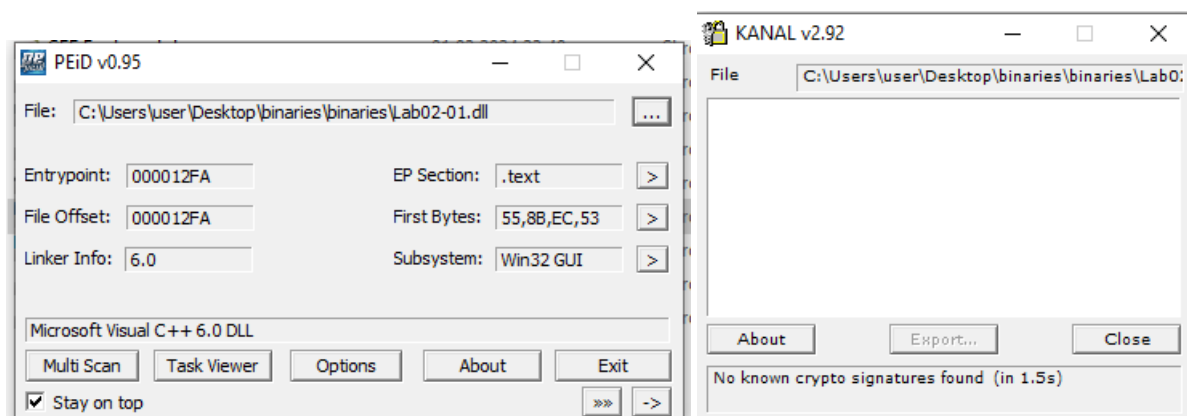
3. Często bywa tak, że złośliwe oprogramowanie znajduje się w formie spakowanej lub zaciemnionej utrudniając analizę. Wykorzystaj narzędzie PEiD lub PPEE do sprawdzenia, czy analizowane pliki znajdują się w formie umożliwiającej pełną analizę. Opisz uzyskany rezultat.

Znalazłem narzędzie peid. Przy użyciu pluginu sprawdziłem szyfrowanie i nie ma żadnego. Textbox podaje informacje o wersji packera: Microsoft Visual C++ 6.0.

Plik .exe.

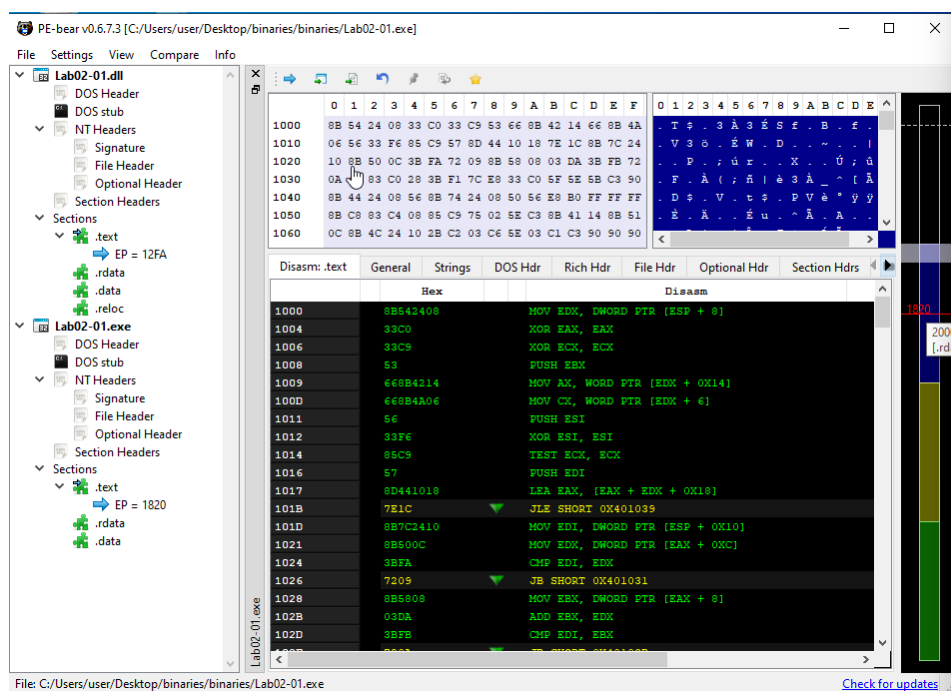


Plik .dll.

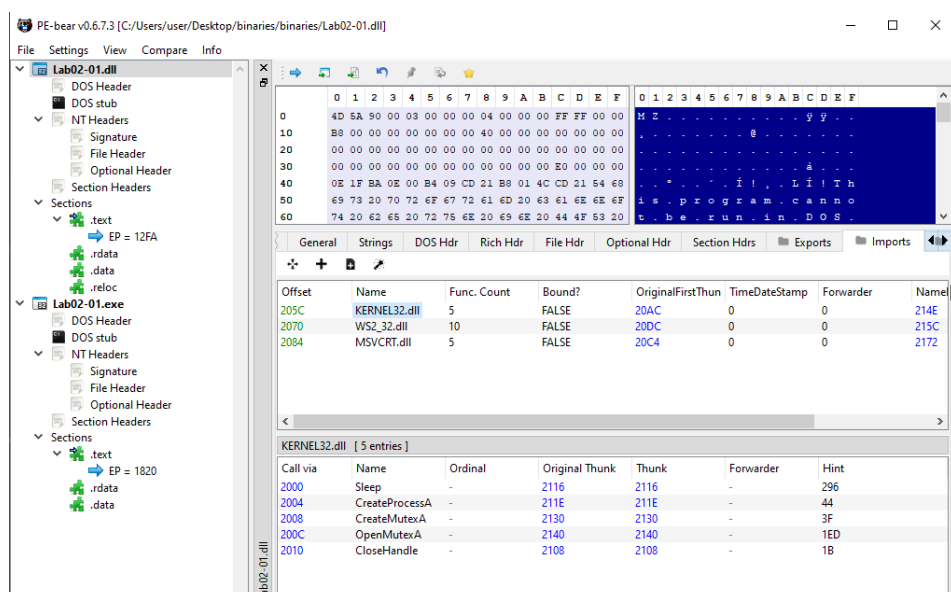


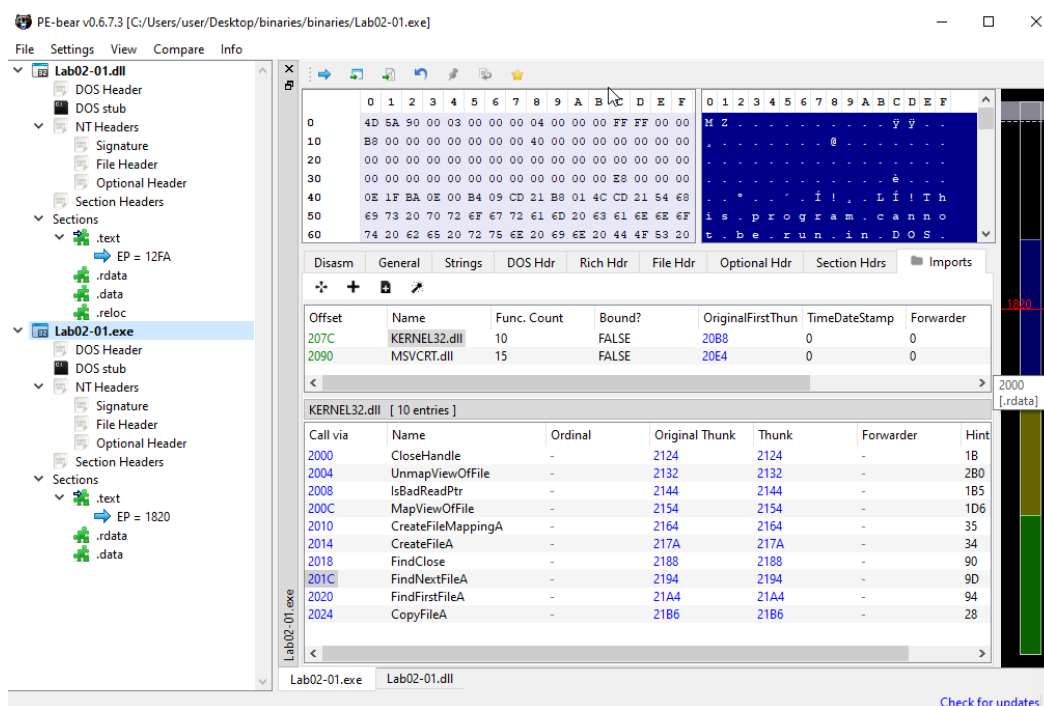
4. W celu statycznego sprawdzenia jak działa złośliwe oprogramowanie, możemy przeanalizować importy do bibliotek wykonywane przez analizowane pliki. Do tego możemy wykorzystać program PE-bear (program posiada funkcjonalność jednoczesnego analizowania dwóch plików). Przeanalizuj wykorzystywane importy do określenia sposobu działania pliku exe oraz dll (Lab02-01.exe i Lab02-01.dll). Opisz wybrane przez siebie najciekawsze importy (za co odpowiadają?).

Uruchomiłem PE-bear.



Według mnie dla obu plików najciekawszym importem jest KERNEL32.dll. oraz wykonywane tam operacje.



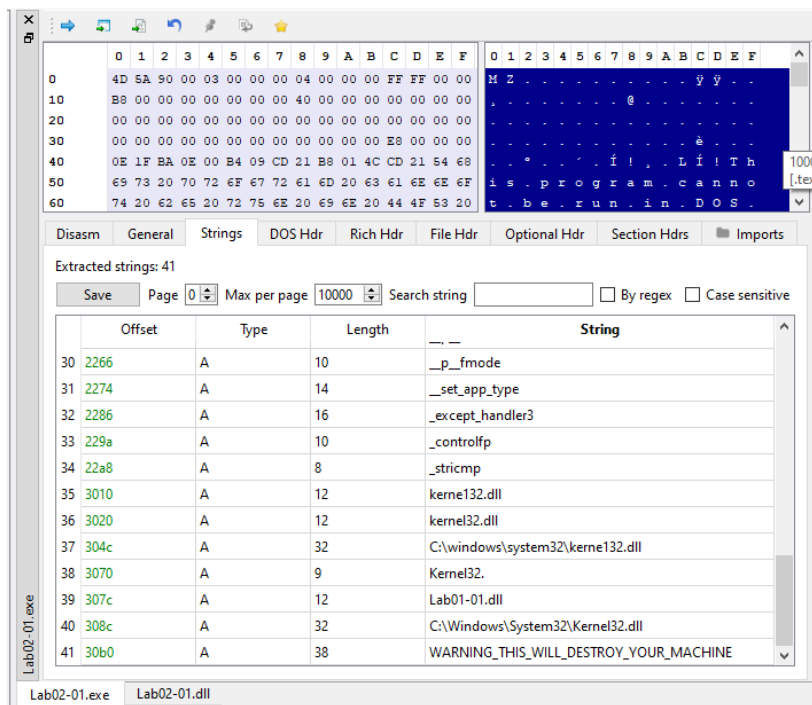


Kernel32.dll jest kluczową biblioteką systemową systemu Windows, umożliwiającą aplikacjom interakcję z jądrem systemu. Importy do niej w pliku malware sugerują wykorzystanie funkcji systemowych do działań złośliwych, takich jak uruchamianie procesów czy zarządzanie zasobami systemowymi.

5. Za co odpowiedzialna jest biblioteka WS2_32.dll (Lab02-01.dll)?

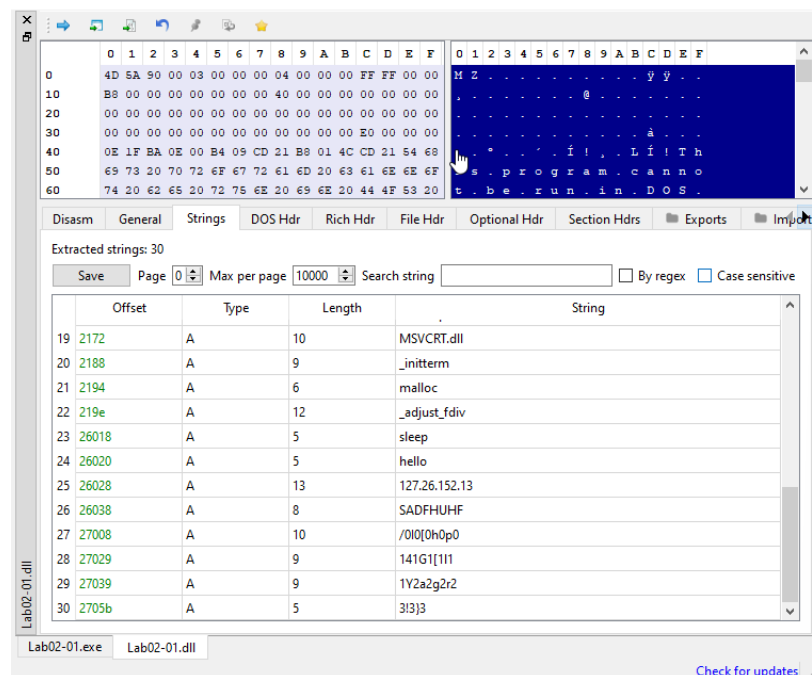
Biblioteka WS2_32.dll jest częścią systemu Windows i zapewnia obsługę funkcji związanych z komunikacją sieciową, takich jak nawiązywanie połączeń, przesyłanie danych oraz obsługa różnych protokołów sieciowych. Złośliwe oprogramowanie może wykorzystać tę bibliotekę do komunikacji z serwerami kontrolnymi, przesyłania skradzionych danych, prowadzenia ataków sieciowych oraz ukrywania swojej aktywności sieciowej, co pozwala na zdalne sterowanie infekowanym systemem i przeprowadzanie różnych działań związanych z atakami na sieć.

6. Wyświetl informacje strings z programu PPEE dla pliku Lab02-01.exe. Zwróć uwagę na ścieżki dostępne do biblioteki „C:\Windows\System32\Kernel32.dll” i jego odpowiednika. O czym mogą świadczyć dwa osobne podobne rekordy?



Moje pierwsze podejrzenie to, że może to być próba spoofa, gdzie złośliwe oprogramowanie próbuje ukryć swoją aktywność, udając działanie znanych i zaufanych procesów lub bibliotek systemowych.

7. Przeanalizuj tym samym sposobem plik Lab02-01.dll i odpowiedz, czy posiada on jakieś informacje mogące świadczyć o komunikacji internetowej?



Przy okazji tego pliku widzimy rekord z adresem IP.

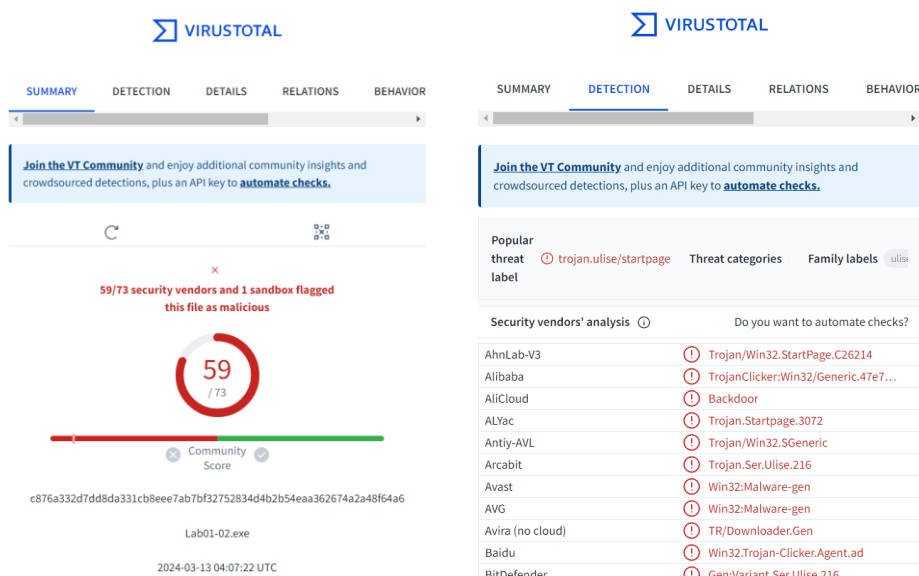
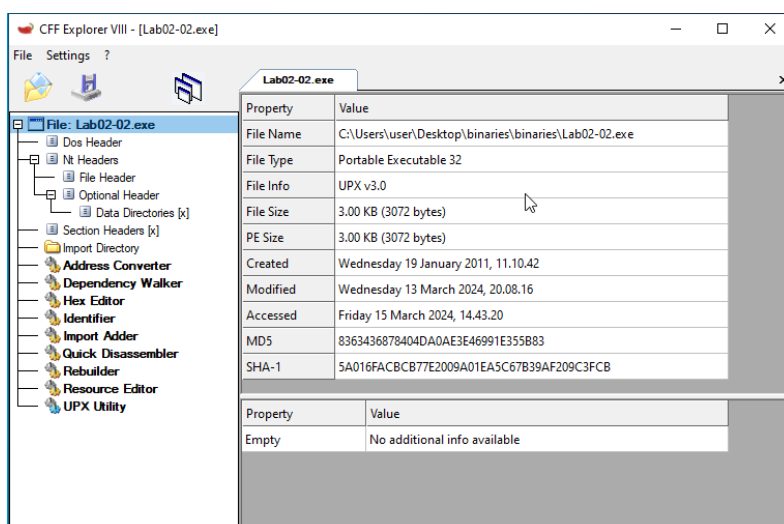
8. Posiadając aktualne informacje, czy jesteś w stanie określić w jaki sposób działają analizowane pliki oraz opisać zależność między plikami (exe i dll)?

Na podstawie znalezionych artefaktów myślę, że powyższa próbka działa jak typowy trojan. Ukrywa się jako wiarygodny i znany program, a później przesyła dane atakującemu. Podejrzewam, że program .exe wykorzystuje .dll to wyświetlenia użytkownikowi tego spoofowanego interfejsu, który doprowadza do tragedii.

Laboratorium 1.2

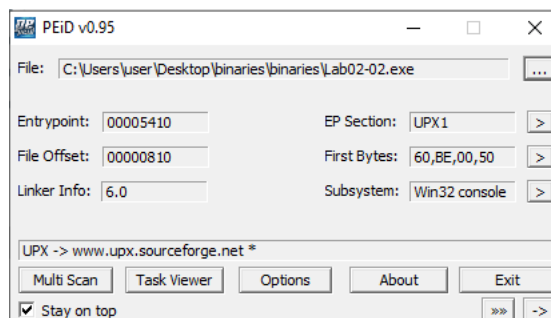
Wykonaj analizę pliku Lab02-02.exe i odpowiedz na pytania.

1. Czy sygnatura analizowanego pliku była już wcześniej analizowana w VirusTotal? Jeśli tak, to podaj wynik skanowania.



2. Sprawdź, czy coś świadczy o tym, że plik jest spakowany lub zaciemniony? Spróbuj go rozpakować.

Po użyciu PEiD okazuje się, że jest spakowany przy pomocy UPX.



Przedstawiam poniżej kolejne kroki w drodze do odpakowania.

Najpierw otwarłem cmd, wpisałem UPX (w celu przeczytania instrukcji) i znalazłem folder z plikami malware.

```
C:\Users\user>UPX
Ultimate Packer For executables
Copyright (C) 1996 - 2024
UPX 4.2.2 Markus Oberhumer, Laszlo Molnar & John Reiser Jan 3rd 2024

Usage: upx [-123456789dlthv] [-qvk] [-o file] file..

Commands:
  -1 compress faster          -9 compress better
  -d decompress              -l list compressed file
  -t test compressed file    -V display version number
  -h give more help          -L display software license

Options:
  -q be quiet                 -v be verbose
  -oFILE write output to 'FILE'
  -f force compression of suspicious files
  -k keep backup files
  file.. executables to (de)compress

Type 'upx --help' for more detailed help.

UPX comes with ABSOLUTELY NO WARRANTY; for details visit https://upx.github.io
```

```
C:\Users\user\Desktop\binaries\binaries>dir
Volume in drive C has no label.
Volume Serial Number is D061-B19F

Directory of C:\Users\user\Desktop\binaries\binaries

13.03.2024  20:08    <DIR>          .
13.03.2024  20:08    <DIR>          ..
13.03.2024  20:08             163 840 Lab02-01.dll
13.03.2024  20:08             16 384 Lab02-01.exe
13.03.2024  20:08              3 072 Lab02-02.exe
13.03.2024  20:08              4 752 Lab02-03.exe
13.03.2024  20:08              36 864 Lab02-04.exe
               5 File(s)          224 912 bytes
               2 Dir(s)  29 979 451 392 bytes free
```

W następnym kroku wypakowałem plik. Okazało się, że rozmiar spakowanego pliku stanowił tylko 18.75% rozpakowanego.

```
C:\Users\user\Desktop\binaries\binaries>UPX -d -o Lab01-02-unpacked.exe Lab01-02.exe
Ultimate Packer for eXecutables
Copyright (C) 1996 - 2024
UPX 4.2.2 Markus Oberhumer, Laszlo Molnar & John Reiser Jan 3rd 2024

File size      Ratio      Format      Name
-----
upx: Lab01-02.exe: FileNotFoundException: Lab01-02.exe: No such file or directory

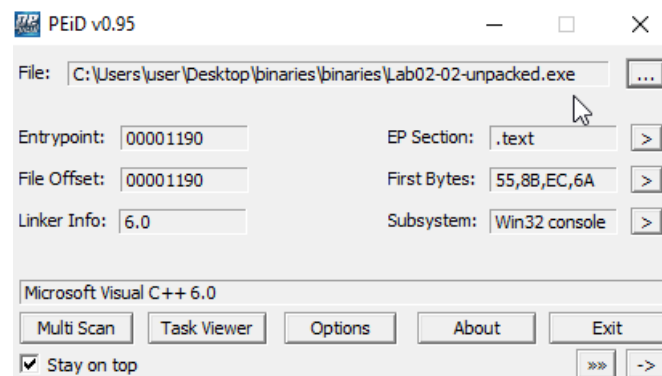
Unpacked 0 files.

C:\Users\user\Desktop\binaries\binaries>UPX -d -o Lab02-02-unpacked.exe Lab02-02.exe
Ultimate Packer for eXecutables
Copyright (C) 1996 - 2024
UPX 4.2.2 Markus Oberhumer, Laszlo Molnar & John Reiser Jan 3rd 2024

File size      Ratio      Format      Name
-----
16384 <-      3072      18.75%      win32/pe      Lab02-02-unpacked.exe

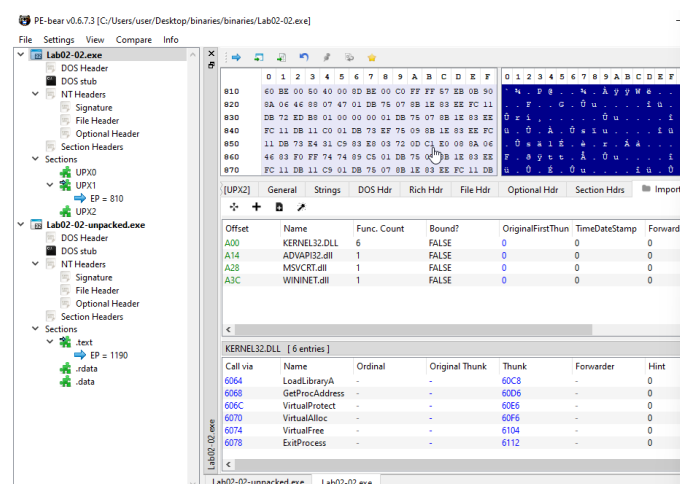
Unpacked 1 file.
```


Na sam koniec widzimy efekt – plik rozpakowany.

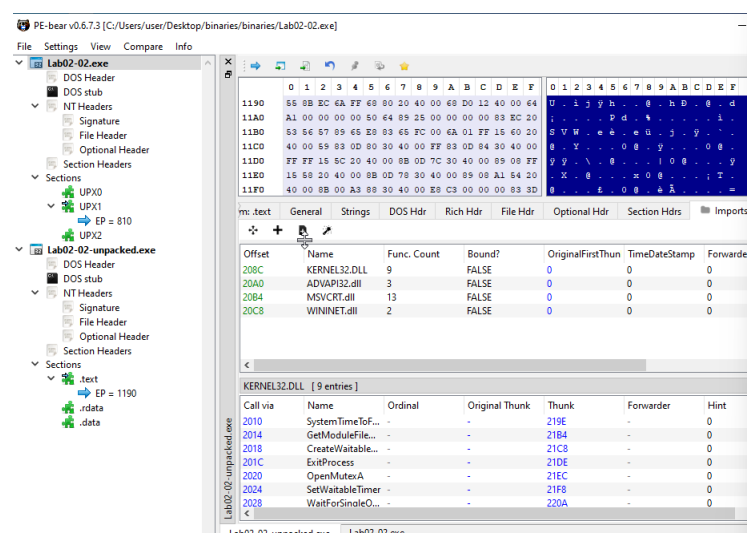


3. Wykorzystuj poznane narzędzia do porównania importów pliku spakowanego z rozpakowanym. Podaj jakie są różnice pomiędzy nimi oraz wymień najciekawsze importy z rozpakowanego pliku.

Plik spakowany, ukrywa niektóre odwołania.



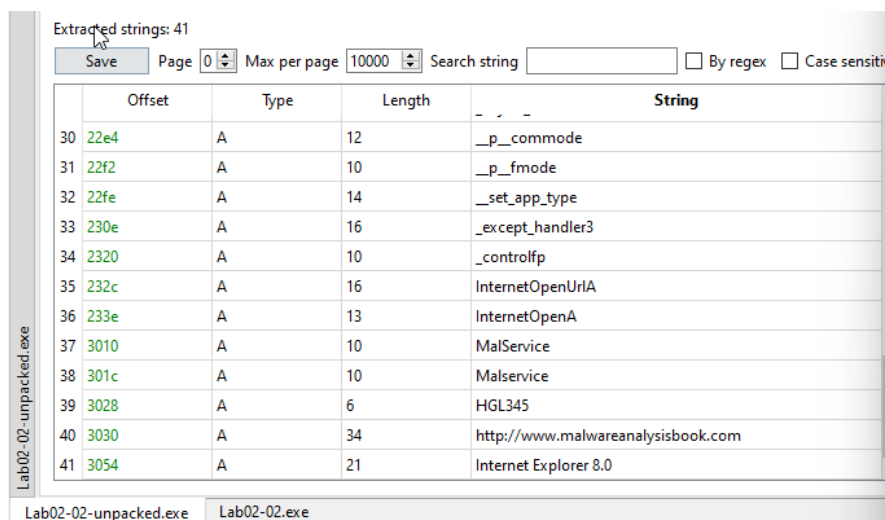
Plik rozpakowany. W nim widzimy większą ilość odwołań niż w spakowanym.



Najciekawsze importy z pliku rozpakowanego to: ADVAPI32.dll oraz WININET.dll. Pierwszy dotyczy otwarcia dispatchera a drugi otwarcia URL.

4. Odszukaj w strings informacje świadczące o połączeniach programu z siecią Internet.

Najciekawsze informacje to oczywiście OpenURL, MalService (prawdopodobnie Malicious Service), Internet Explorer i adres strony malwareanalysis.



Extracted strings: 41

Save Page 0 Max per page 10000 Search string ☐ By regex ☐ Case sensitive

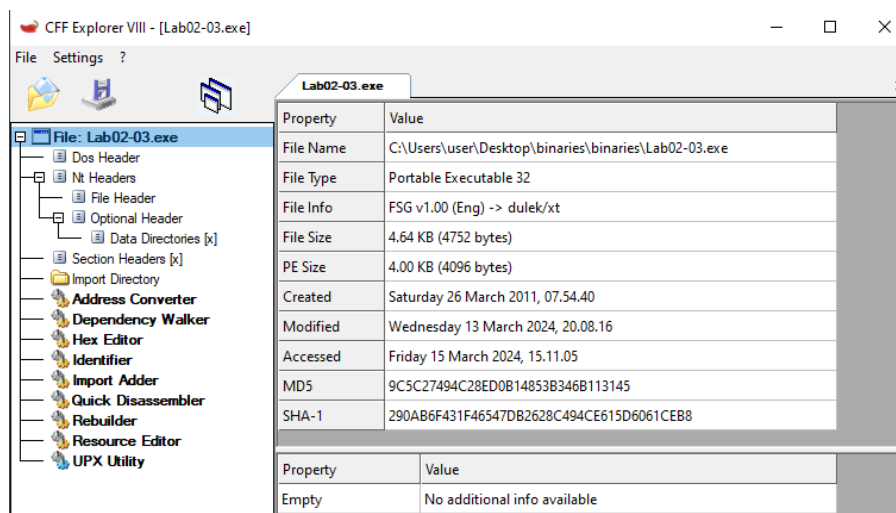
	Offset	Type	Length	String
30	22e4	A	12	__p_commode
31	22f2	A	10	__p_fmode
32	22fe	A	14	__set_app_type
33	230e	A	16	__except_handler3
34	2320	A	10	__controlfp
35	232c	A	16	InternetOpenUrlA
36	233e	A	13	InternetOpenA
37	3010	A	10	MalService
38	301c	A	10	Malservice
39	3028	A	6	HGL345
40	3030	A	34	http://www.malwareanalysisbook.com
41	3054	A	21	Internet Explorer 8.0

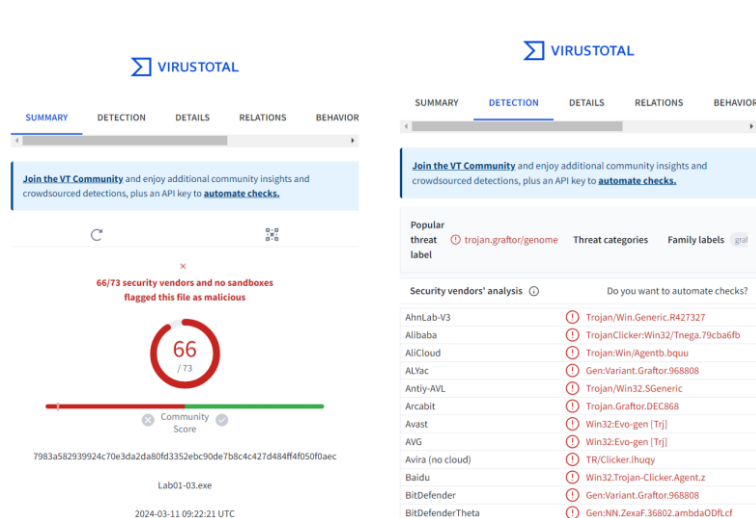
Lab02-02-unpacked.exe Lab02-02.exe

Laboratorium 1.3

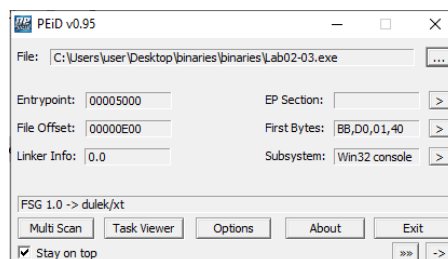
Przeprowadź analizę pliku Lab02-03.exe

1. Czy sygnatura analizowanego pliku była już wcześniej analizowana w VirusTotal? Jeśli tak, to podaj wynik skanowania.





2. Sprawdź, czy coś świadczy o tym, że plik jest spakowany lub zaciemniony? Czy będziesz w stanie rozpakować go przy pomocy UPX? Jeśli nie, to dlaczego?



Według PEiD plik jest spakowany przy pomocy innego packera niż UPX, z tego prostego powodu nie dam rady nic zrobić z UPX. Mogę to nawet sprawdzić.

```
C:\Users\user\Desktop\binaries\binaries>UPX -d -o Lab02-03-unpacked.exe Lab02-03.exe
Ultimate Packer for executables
Copyright (C) 1996 - 2024
UPX 4.2.2 Markus Oberhumer, Laszlo Molnar & John Reiser Jan 3rd 2024

File size      Ratio      Format      Name
-----
upx: Lab02-03.exe: NotPackedException: not packed by UPX

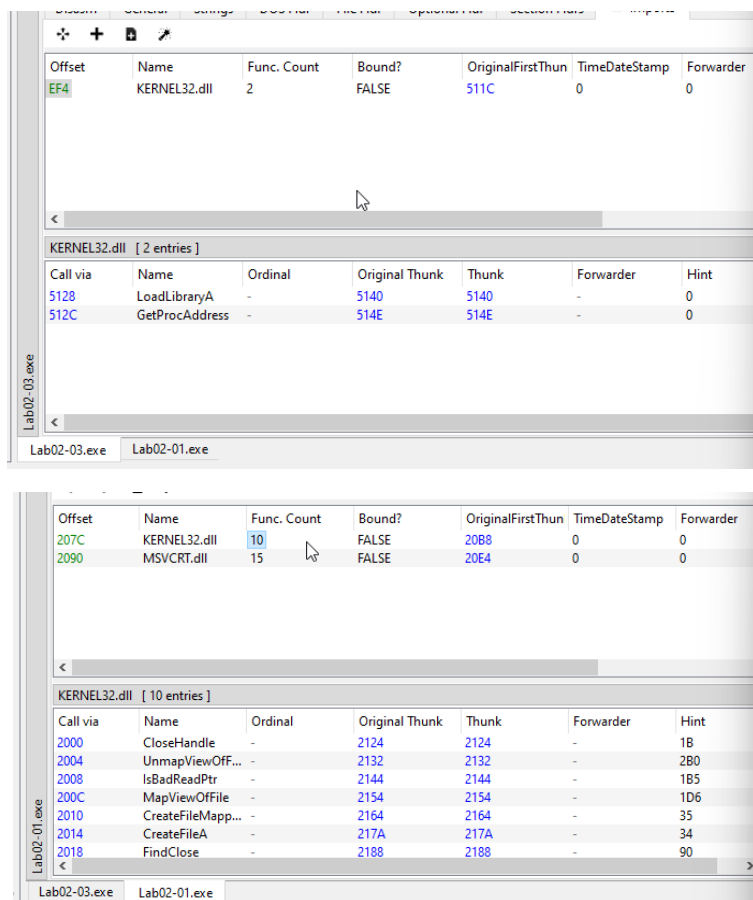
Unpacked 0 files.
```

3. Czy jesteś w stanie sprawdzić datę kompilacji pliku (Time Data Stamp)?
Niestety nie.

The CFF Explorer VIII application window shows the file 'Lab02-03.exe'. The 'File Header' tab is selected, displaying various fields including Machine (Intel 386), NumberOfSections (00000066), TimeDateStamp (00000000), and Characteristics (010F).

Member	Offset	Size	Value	Meaning
Machine	00000064	Word	014C	Intel 386
NumberOfSections	00000066	Word	0003	
TimeDateStamp	00000068	Dword	00000000	
PointerToSymbolTa...	0000006C	Dword	00000000	
NumberOfSymbols	00000070	Dword	00000000	
SizeOfOptionalHea...	00000074	Word	00E0	
Characteristics	00000076	Word	010F	Click here

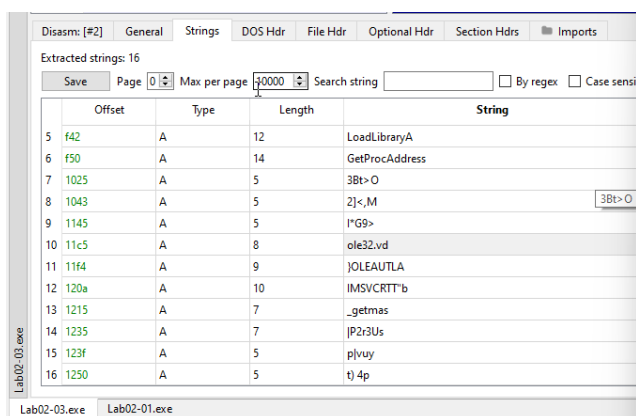
4. Wykorzystaj poznane narzędzia do porównania importów pliku, odpowiedz, czy jesteś w stanie sprawdzić funkcjonalność badanego pliku, w taki sam sposób jak w Laboratorium 1.1?



Są delikatne różnice, nie widać wierszy kolumny „Hint” oraz jest mniej importów (nie wiadomo wprost czy to wina spakowania).

5. Odszukaj w strings informacje świadczące o połączeniach programu z siecią Internet.

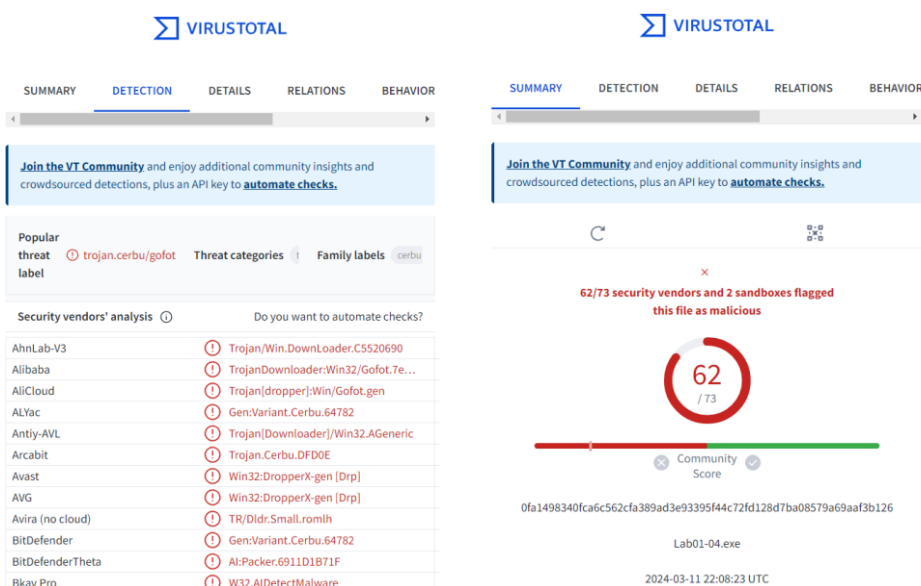
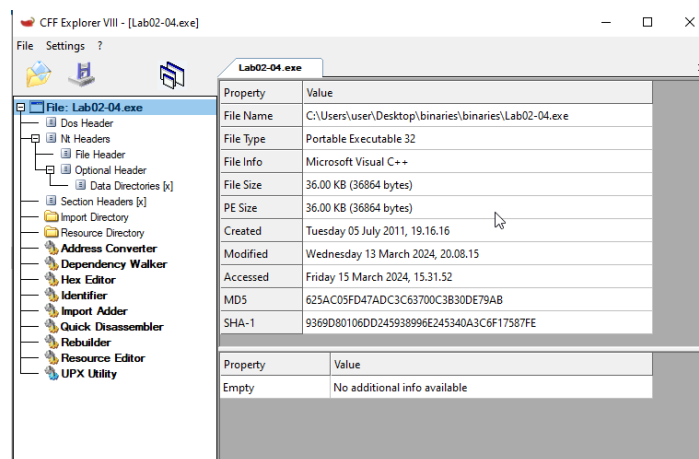
Szczerze mówiąc nie ma dużo ciekawych informacji związanych z Internetem.



Laboratorium 1.4

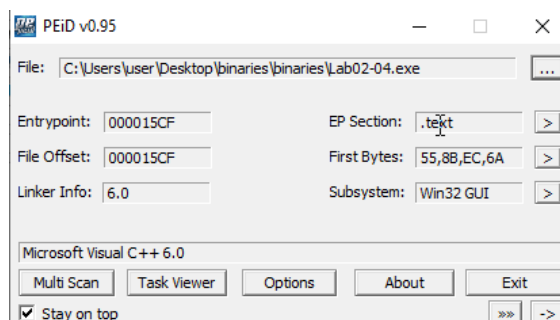
Przeprowadź analizę pliku Lab02-04.exe

1. Czy sygnatura analizowanego pliku była już wcześniej analizowana w VirusTotal? Jeśli tak, to podaj wynik skanowania.



2. Sprawdź, czy coś świadczy o tym, że plik jest spakowany lub zaciemniony?

Wstępnie nic nie świadczy o zaciemnieniu.



3. Kiedy ten plik został skompilowany?

Został skompilowany 30 sierpnia 2019r.

Member	Offset	Size	Value	Meaning
Machine	000000EC	Word	014C	Intel 386
NumberOfSections	000000EE	Word	0004	
TimeDateStamp	000000F0	Dword	5D69A2B3	
PointerToSymbolTa...	000000F4	Dword	00000000	
NumberOfSymbols	000000F8	Dword	00000000	
SizeOfOptionalHea...	000000FC	Word	000E	
Characteristics	000000FE	Word	010F	Click here

Epoch & Unix Timestamp Conversion Tools

The current Unix epoch time is 1710513541

Convert epoch to human-readable date and vice versa

5D69A2B3 [Timestamp to Human date](#) [reset](#)

Supports Unix timestamps in seconds, milliseconds, microseconds and nanoseconds.

Converting hexadecimal timestamp to decimal: 1567204019

Assuming that this timestamp is in **seconds**:

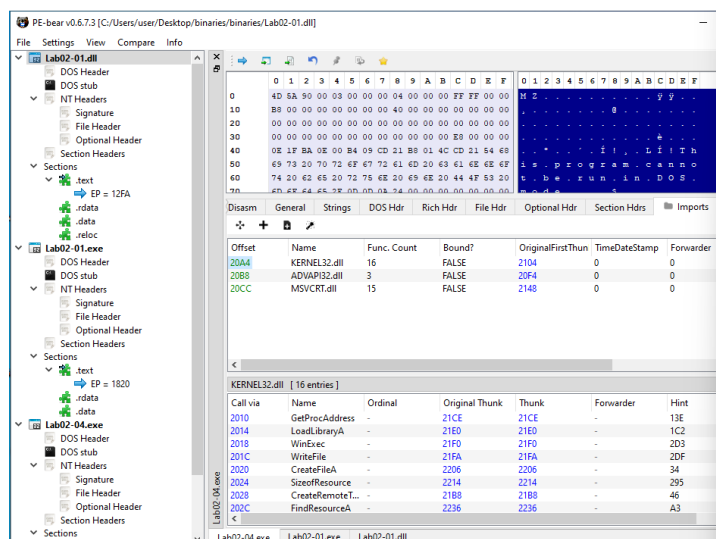
GMT: Friday, 30 August 2019 22:26:59

Your time zone: sobota, 31 sierpnia 2019 00:26:59 GMT+02:00 DST

Relative: 5 years ago

4. Wykorzystuj poznane narzędzia do porównania importów pliku, odpowiedz, czy jesteś w stanie sprawdzić funkcjonalność badanego pliku, w taki sam sposób jak w Laboratorium 1.1?

Tak, wszystkie są podobne.



5. Odszukaj w strings informacje świadczące o połączeniach programu z siecią Internet.

Standardowo widzimy informacje o URL.

Extracted strings: 84

Save Page 0 Max per page 10000 Search string ☐ By regex ☐ Case sensitive

	Offset	Type	Length	String
65	619a	A	12	KERNEL32.dll
66	61aa	A	18	URLDownloadToFileA
67	61be	A	10	urlmon.dll
68	61cc	A	9	_snprintf
69	61d6	A	10	MSVCRT.dll
70	61e4	A	5	_exit
71	61ec	A	11	_XcptFilter
72	6202	A	13	_p__initenv
73	6212	A	13	_getmainargs
74	6222	A	9	_initterm
75	622e	A	16	_setusermatherr
76	6242	A	12	_adjust_fdiv
77	6252	A	12	_p__commode
78	6262	A	10	p_fmode

Lab02-04.exe Lab02-01.exe Lab02-01.dll

6. Czy analizowany plik posiada importy świadczące o dostępie do funkcji sieciowych?

Tak jak na ostatnim zdjęciu.

7. Badany plik zawiera jeden zasób w sekcji zasobów. Użyj programu Resource Hacker, aby zbadać ten zasób, a następnie użyj go do jego wyodrębnienia. Wczytaj plik w programie a następnie użyj funkcji „Action->Save Resource to Bin File” Czego możesz się dowiedzieć analizując ten wyeksportowany zasób?

Strings DOS Hdr Rich Hdr File Hdr Optional Hdr Section Hdrs Imports Resources

Offset	Name	Value	Value	Meaning	Meaning	Type
4000	Characteristics	0				
4004	TimeDateStamp	0				
4008	MajorVersion	0				
400A	MinorVersion	0				
400C	NumberOfNam...	1				
400E	NumberOfEnt...	0				
4010	Name_0	80000058	80000018	4058	4018	BIN

< >

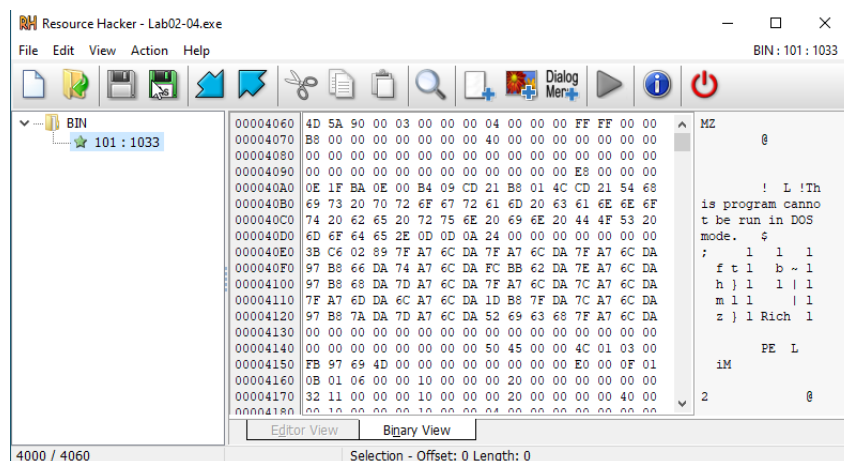
Table Content

Resources

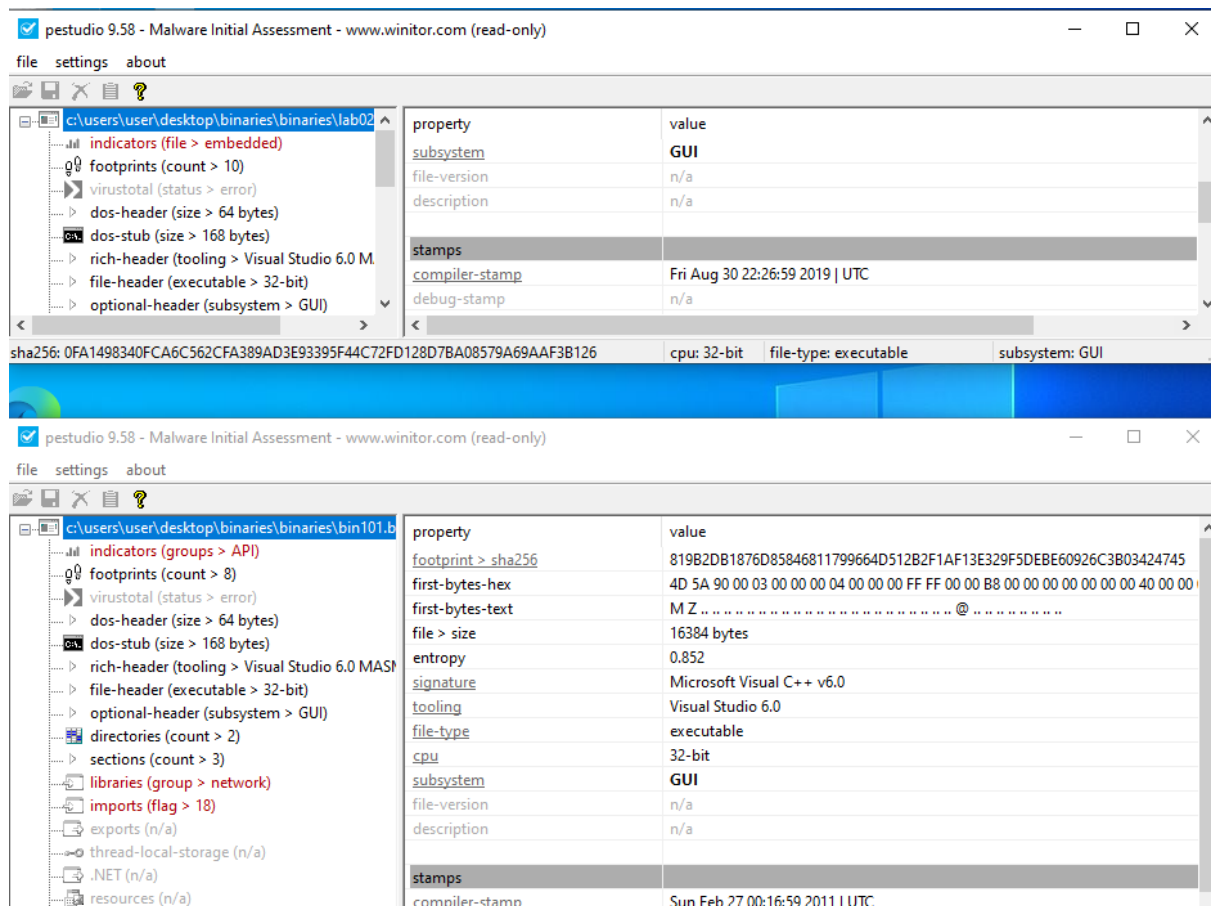
Offset	Name	Value
4048	OffsetToData	4060
404C	DataSize	4000
4050	CodePage	0
4054	Reserved	0

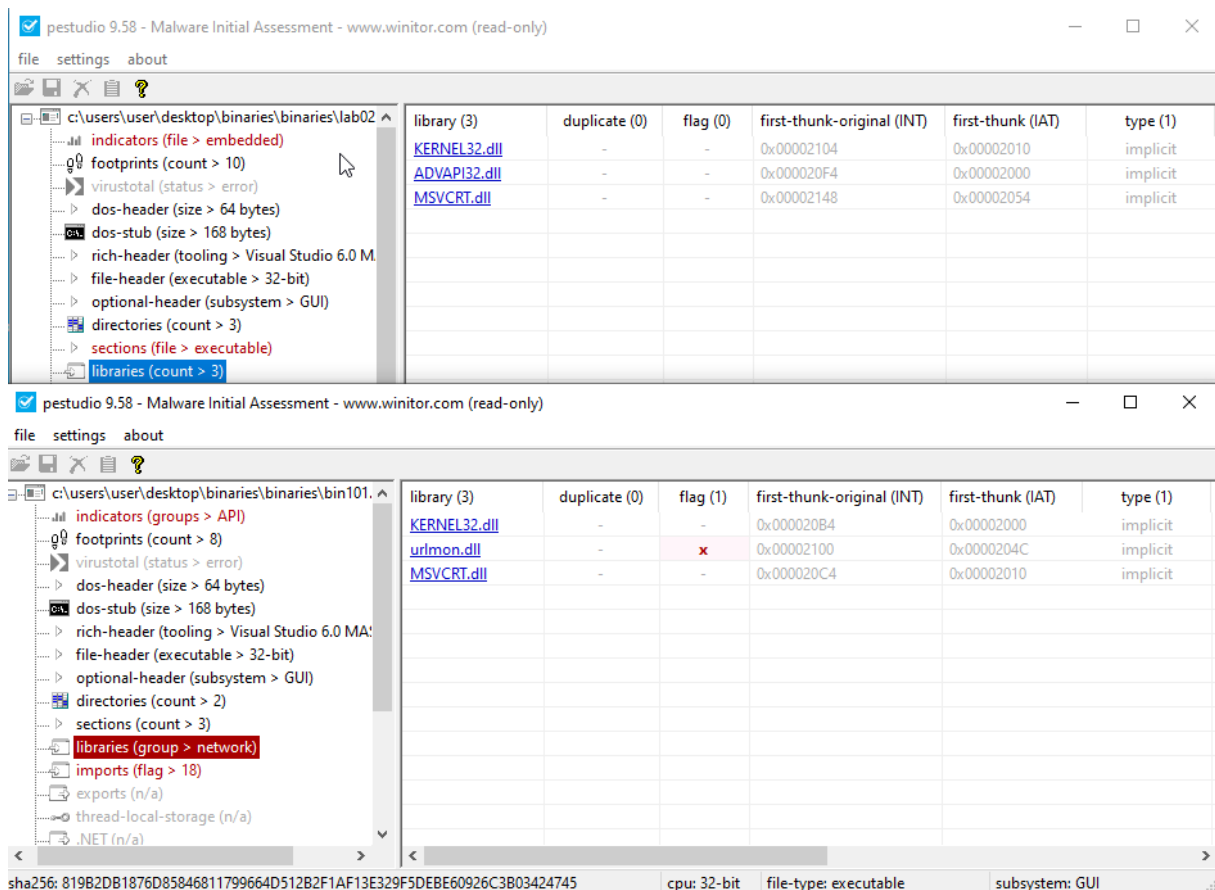
Lab02-04.exe Lab02-01.exe Lab02-01.dll

Niestety nie mam narzędzia Resource Hacker i muszę je doinstalować. Po udanej instalacji znowu wyłączyłem karty sieciowe, żeby było bezpiecznie.



Na pierwszy rzut oka widać różnicę, nawet w czasie kompilacji. Ten zasób binarny, który sobie pozyskałem z Resource Hacker można bardziej dokładnie analizować. To wygląda jak by ten zasób był osadzony wewnątrz całego pliku .exe.





Podaję, że jest możliwość używania tego zasobu do dalszych modyfikacji i „crackowania” malware. Podejrzałem również oba zasoby przy pomocy programu do deasemblacji - „IDA”, tam również są zauważalne różnice. Możemy sobie wszystko rozłożyć na czynniki pierwsze.