

Analiza statystyczna

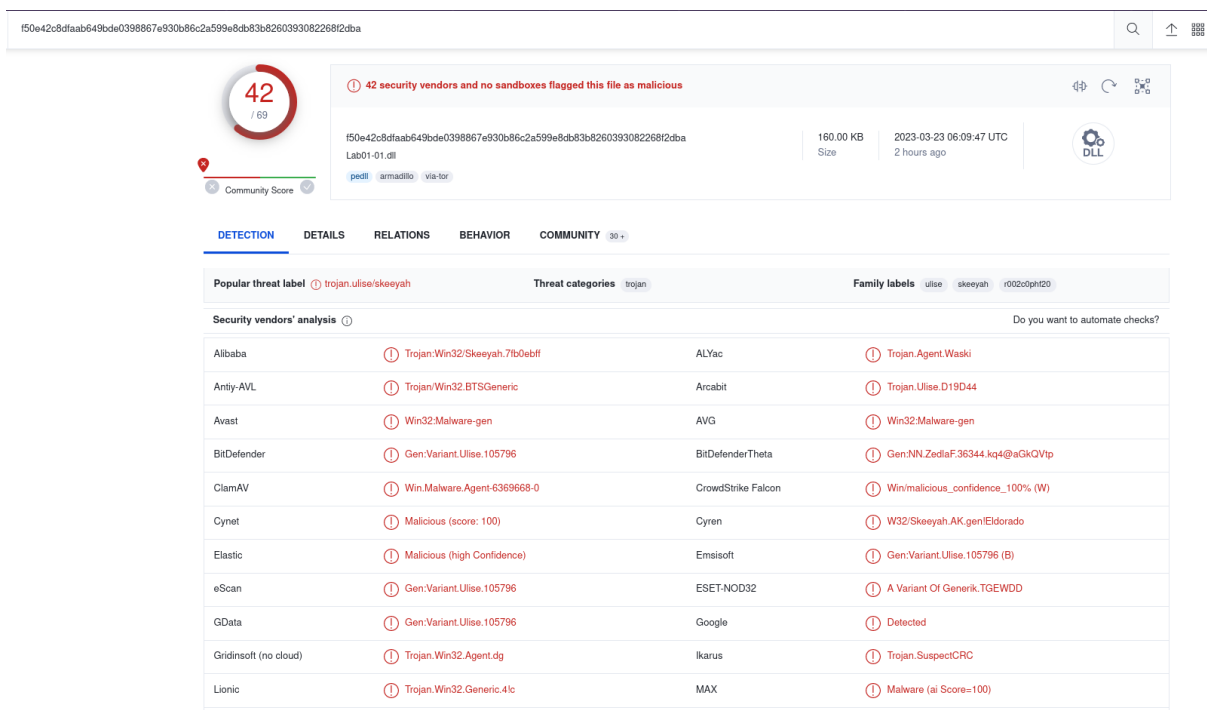
Laboratorium 1.1

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?

```
arek@Arek:~/studia/malware/lab2/binaries$ md5sum Lab02-01*
290934c61de9176ad682ffdd65f0a669  Lab02-01.dll
bb7425b82141a1c0f7d60e5106676bb1  Lab02-01.exe
```

Plik Lab02-01.dll



The screenshot shows the VirusTotal analysis page for the file Lab02-01.dll. The file's MD5 hash is f50e42c8dfaab649bde0398867e930b86c2a599e8db83b260393082268f2dba. The file size is 160.00 KB, and it was uploaded 2 hours ago. The analysis shows that 42 security vendors and no sandboxes flagged this file as malicious. The file is identified as a Trojan, specifically Trojan:Win32/Skeeyah.7fb0ebff. The analysis table below shows the results from various security vendors.

Security vendors' analysis	Threat categories	Family labels
Alibaba	Trojan:Win32/Skeeyah.7fb0ebff	ALYac
Antiy-AVL	Trojan:Win32/BTSGeneric	Arcabit
Avast	Win32:Malware-gen	AVG
BitDefender	Gen:Variant.Ulise.105796	BitDefenderTheta
ClamAV	Win.Malware.Agent-6369668-0	CrowdStrike Falcon
Cyren	Malicious (score: 100)	Cyren
Elastic	Malicious (high Confidence)	Emsisoft
eScan	Gen:Variant.Ulise.105796	ESet-NOD32
GData	Gen:Variant.Ulise.105796	Google
Gridinsoft (no cloud)	Trojan.Win32.Agent.dg	Ikarus
Lionic	Trojan.Win32.Generic.4fc	MAX

Plik Lab02-01.exe

58898bd42c5bd3bf9b1389f0eee5b39cd59180e8370eb9ea838a0b327bd6fe47

48

69

Community Score

peexe checks-disk-space via-tor detect-debug-environment idie armadillo checks-user-input long-sleeps

48 security vendors and 1 sandbox flagged this file as malicious

58898bd42c5bd3bf9b1389f0eee5b39cd59180e8370eb9ea838a0b327bd6fe47

Lab01-01.exe

16.00 KB

Size

2023-03-23 06:14:11 UTC

2 hours ago

EXE

DETECTION

DETAILS

RELATIONS

BEHAVIOR

COMMUNITY 30+

Dynamic Analysis Sandbox Detections

The sandbox ReaQta-Hive flags this file as: MALWARE

Popular threat label

trojan.ulise/aenjaris

Threat categories

trojan

Family labels

ulise aenjaris r002c0d420

Security vendors' analysis

Do you want to automate checks?

AhnLab-V3	Trojan:Win32.Agent.C957604	Alibaba	Trojan:Win32/Aenjaris.2be749b4
ALYac	Trojan.Agent.163845S	Antiy-AVL	Trojan:Win32.TSGeneric
Arcabit	Trojan.Ulise.D1BC1E	Avast	Win32:Malware-gen
AVG	Win32:Malware-gen	Avira (no cloud)	HEUR/AGEN.1344261
BitDefender	Gen:Variant.Ulise.113694	ClamAV	Win.Malware.Agent-6342616-0
CrowdStrike Falcon	Win/malicious_confidence_100% (W)	Cynet	Malicious (score: 100)
Cyren	W32/Ulise.CK.genIEldorado	Elastic	Malicious (high Confidence)
Emsisoft	Gen:Variant.Ulise.113694 (B)	eScan	Gen.Variant.Ulise.113694
ESET-NOD32	A Variant Of Win32/Agent.WOM	Fortinet	W32/Agent.WOM!tr

Jak widać oba pliki zostały uznane przez VirusTotal za szczególnie złośliwe

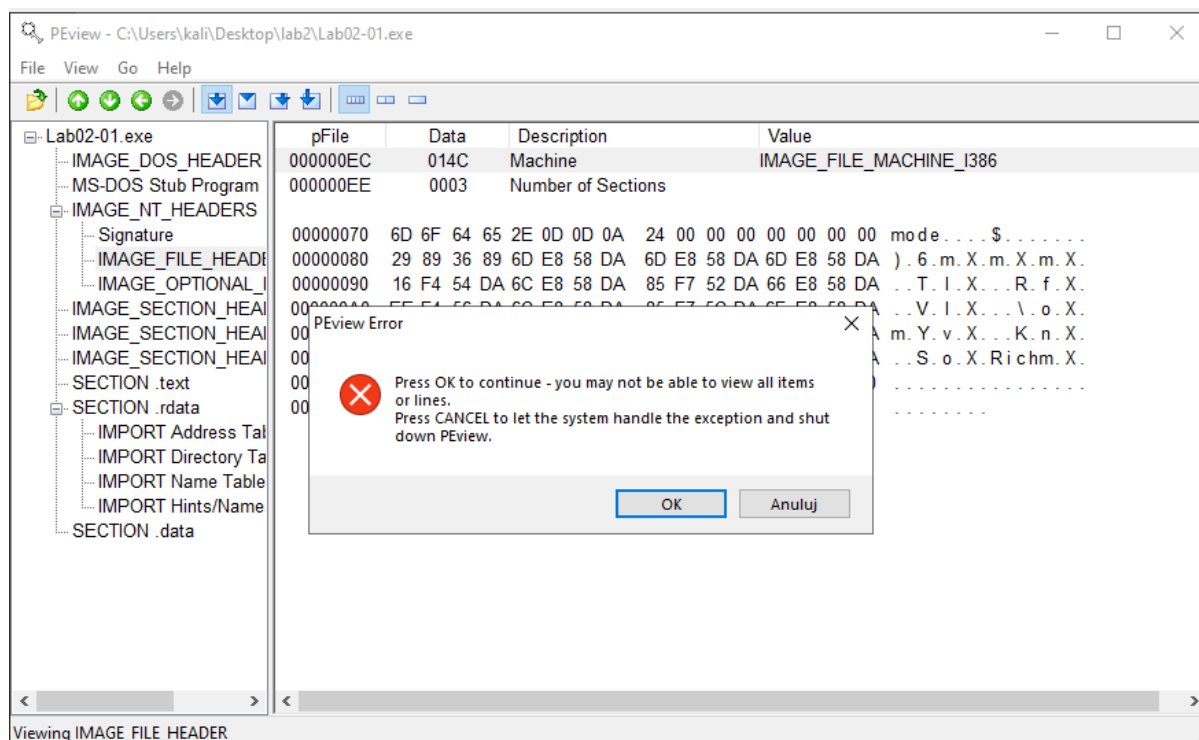
Najwcześniejszy rekord analizy plików pochodzi z 2018 roku



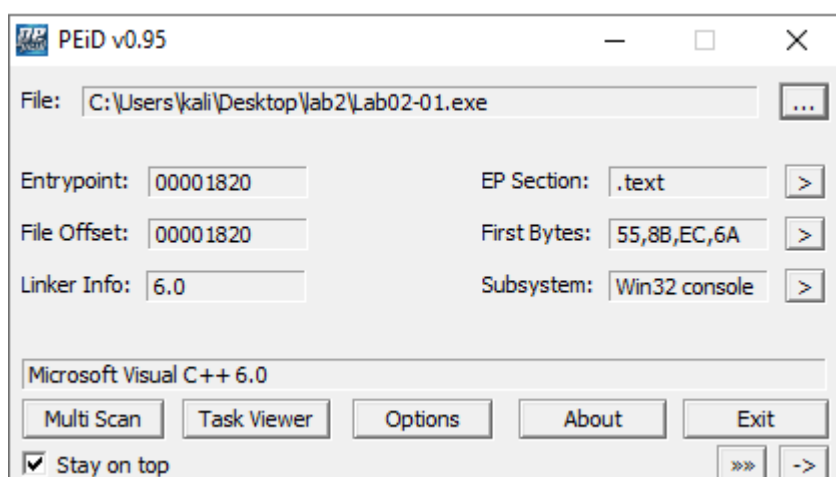
2. Wykorzystując narzędzie PExview odszukaj informacje o dacie skompilowania programu

Dla pliku Lab02-01.exe nie udało mi się znaleźć informacji o dacie skompilowania programu. Niestety jedna sekcja odmawiała posłuszeństwa i nie chciała się otworzyć ``IMAGE_NT_HEADERS` > `IMAGE_SECTION_HEADER.rdata``. A sam program PExview się crashował

Udało mi się znaleźć informację, że to właśnie tam byłaby przechowywana data skompilowania programu



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.



Plik Lab02-01.exe jest w formie umożliwiającej pełną analizę

PEiD v0.95

File: C:\Users\kali\Desktop\lab2\Lab02-01.dll

Entrypoint: 000012FA EP Section: .text

File Offset: 000012FA First Bytes: 55,8B,EC,53

Linker Info: 6.0 Subsystem: Win32 GUI

Microsoft Visual C++ 6.0 DLL

Multi Scan Task Viewer Options About Exit

☒ Stay on top

PE Disassembler v0.03 :: CADT

```

100012FA: 55          PUSH EBP
100012FB: 8BEC       MOV EBP, ESP
100012FD: 53          PUSH EBX
100012FE: 8B5D08     MOV EBX, [EBP+08H]
10001301: 56          PUSH ESI
10001302: 8B750C     MOV ESI, [EBP+0CH]
10001305: 57          PUSH EDI
10001306: 8B7D10     MOV EDI, [EBP+10H]
10001309: 85F6       TEST ESI, ESI
1000130B: 7509       JNZ 10001316H
1000130D: 833D5860021000 CMP [10026058H], 00000000H
10001314: EB26       JMP 1000133CH
10001316: 83FE01     CMP ESI, 00000001H
10001319: 7405       JZ 10001320H
1000131B: 83FE02     CMP ESI, 00000002H
1000131E: 7522       JNZ 10001342H
10001320: A168600210 MOV EAX, [10026068H]
10001325: 85C0       TEST EAX, EAX
10001327: 7409       JZ 10001332H
10001329: 57          PUSH EDI
1000132A: 56          PUSH ESI
1000132B: 53          PUSH EBX
1000132C: FFD0       CALL EAX

```

PE Details

Basic Information

EntryPoint: 000012FA	SubSystem: 0002
ImageBase: 10000000	NumberOfSections: 0004
SizeOfImage: 00028000	TimeDateStamp: 4D0E2FE6
BaseOfCode: 00001000	SizeOfHeaders: 00001000
BaseOfData: 00002000	Characteristics: 210E
SectionAlignment: 00001000	Checksum: 00000000
FileAlignment: 00001000	SizeOfOptionalHeader: 00E0
Magic: 010B	NumOfRvaAndSizes: 00000010

Directory Information

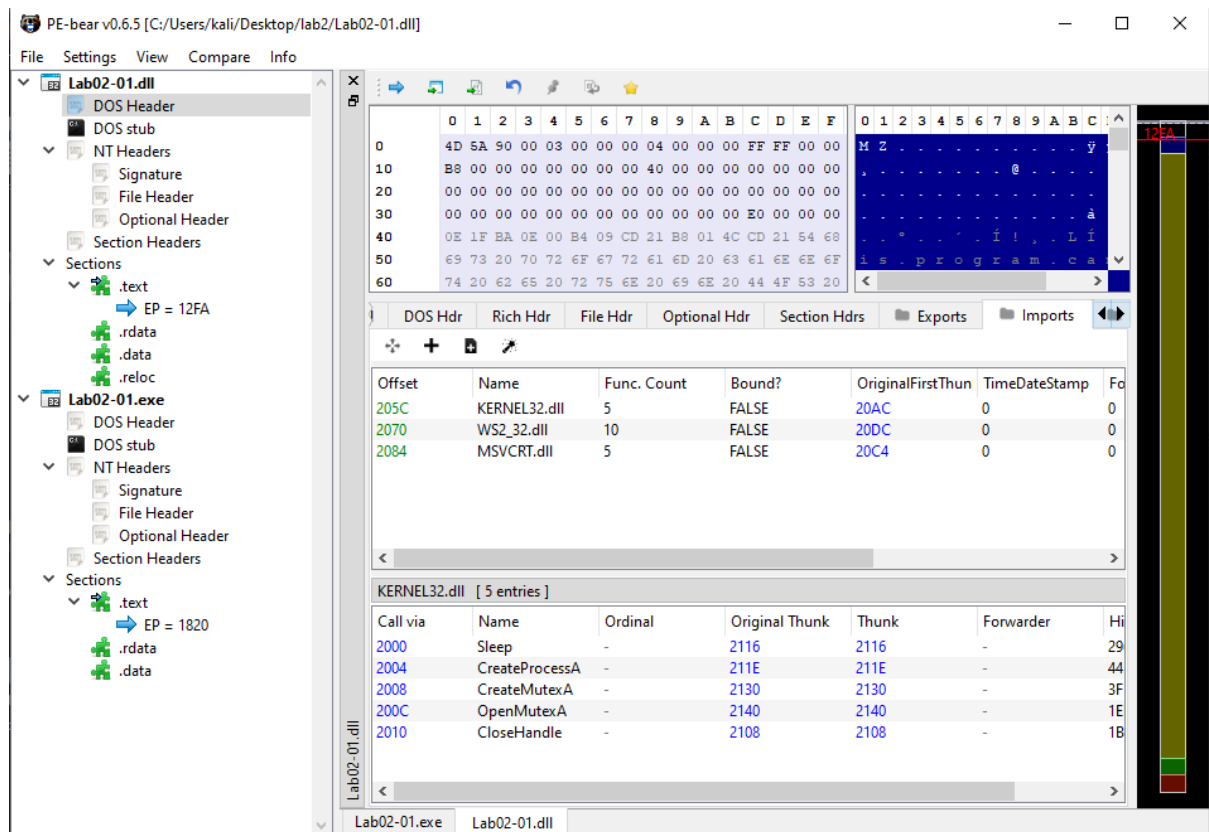
	RVA	SIZE
ExportTable:	000021B0	00023E16
ImportTable:	0000205C	00000050
Resource:	00000000	00000000
TLSTable:	00000000	00000000
Debug:	00000000	00000000

[16Edit FX] - "memory buffer" [READONLY]

00002050:	74 00 00 80 09 00 00 80 00 00 00 00 AC 20 00 00	t...e...e...e...!
00002060:	00 00 00 00 00 00 00 00 4E 21 00 00 00 20 00 00	...N...N...N...
00002070:	DC 20 00 00 00 00 00 00 00 00 00 00 SC 21 00 00	...N...N...N...
00002080:	30 20 00 00 C4 20 00 00 00 00 00 00 00 00 00 00	...N...N...N...
00002090:	72 21 00 00 18 20 00 00 00 00 00 00 00 00 00 00	...N...N...N...
000020A0:	00 00 00 00 00 00 00 00 00 00 00 00 00 16 21 00 00	...N...N...N...
000020B0:	1E 21 00 00 30 21 00 00 40 21 00 00 08 21 00 00	...N...N...N...
000020C0:	00 00 00 00 9C 21 00 00 92 21 00 00 86 21 00 00	...N...N...N...
000020D0:	7E 21 00 00 68 21 00 00 00 00 00 00 17 00 00 80	...N...N...N...
000020E0:	73 00 00 80 0B 00 00 80 04 00 00 80 13 00 00 80	...N...N...N...
000020F0:	16 00 00 80 10 00 00 80 03 00 00 80 74 00 00 80	...N...N...N...
00002100:	09 00 00 80 00 00 00 00 1B 00 43 6F 73 65 48	...N...N...N...
00002110:	61 6E 64 6C 65 00 96 02 53 6C 65 65 70 00 44 00	...N...N...N...
00002120:	43 72 65 61 74 65 50 72 6F 63 65 73 73 41 00 00	...N...N...N...
00002130:	3F 00 43 72 65 61 74 65 4D 75 74 65 78 41 00 00	...N...N...N...
00002140:	ED 01 4F 70 65 6E 4D 75 74 65 78 41 00 00 4B 45	...N...N...N...
00002150:	52 4E 45 4C 33 32 2E 64 6C 6C 00 00 57 53 32 5F	...N...N...N...
00002160:	33 32 2E 64 6C 6C 00 00 C0 02 73 74 72 6E 63 6D	...N...N...N...
00002170:	70 00 4D 53 56 43 52 54 2E 64 6C 6C 00 00 52 02	...N...N...N...
00002180:	66 72 65 65 00 00 0F 01 5F 69 6E 69 74 74 65 72	...N...N...N...
00002190:	6D 00 91 02 6D 61 6C 6C 6F 63 00 00 9D 00 5F 61	...N...N...N...
000021A0:	64 6A 75 73 74 5F 66 64 69 76 00 00 00 00 00 00	...N...N...N...

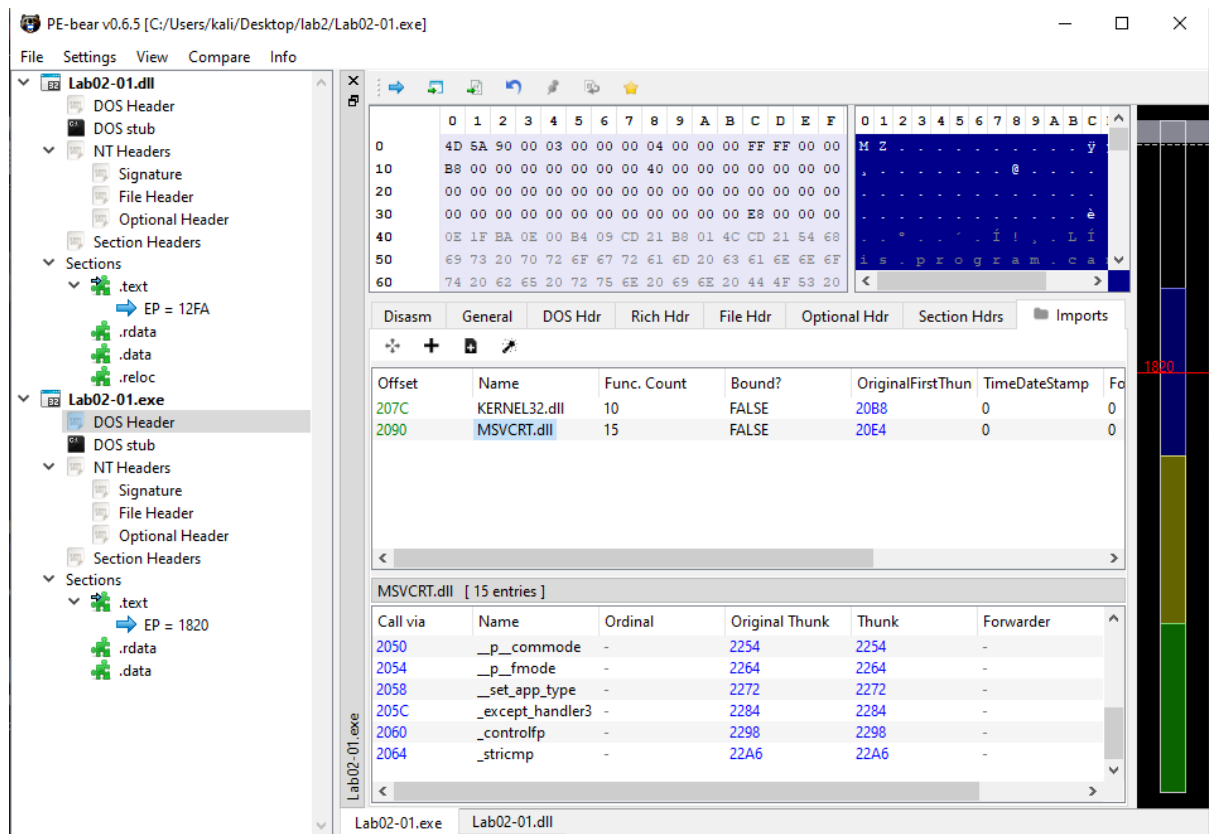
Offset: 0x0000205C - 0x000020AB Size: 0x00000050

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?).



Dla biblioteki *Lab02-01.dll* są to poniższe importy:

- KERNEL32.dll - wywołująca funkcje takie jak Sleep, CreateMutex, CloseHandle przy czym najciekawsza jest właśnie CreateProcess ponieważ program będzie prawdopodobnie próbował stworzyć potomne procesy
- WS2_32.dll
- MSVCRT.dll - do zarządzania pamięcią - malloc, free, strcmp



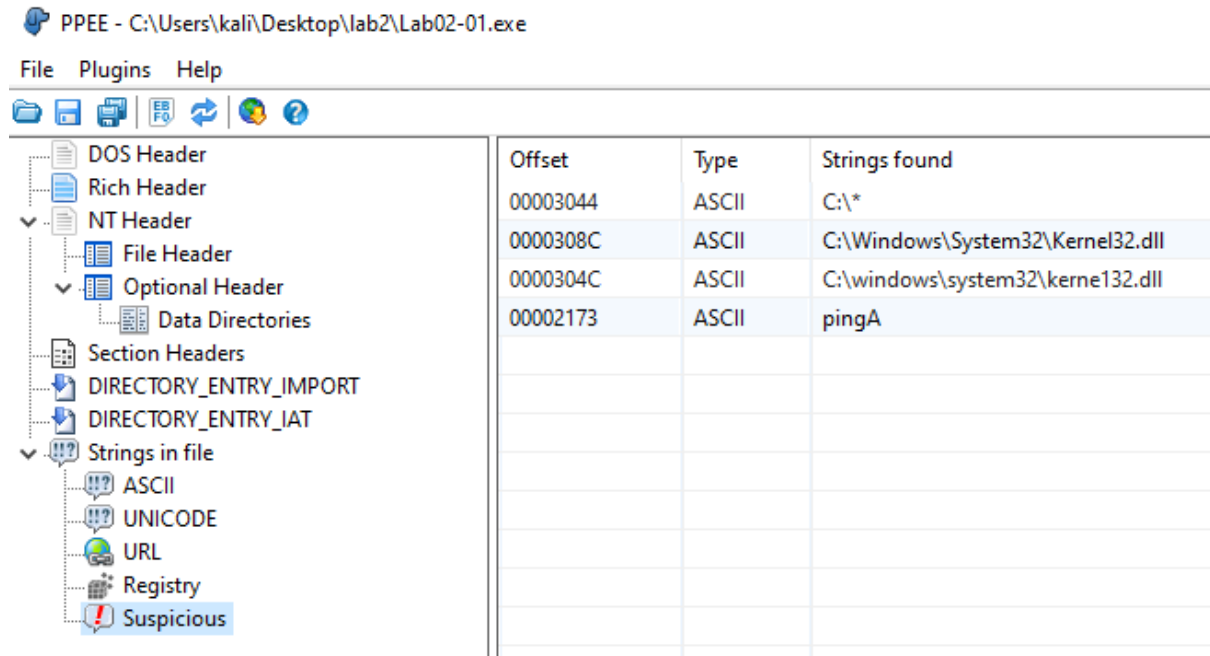
Dla pliku Lab02-01.exe importowane są 2 te same biblioteki, jednakże są one bardziej rozbudowane

- KERNEL32.dll - różni się funkcjami odpowiedzialnymi za znajdowanie plików
- MSVCRT.dll - dodatkowo sprawdza wiele wyjątków

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

Służy do nawiązywania i zarządzania dynamicznymi połączeniami sieciowymi - Windows Sockets. Aplikacja zawsze działa w RAM'ie

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?



Program wskazał podane 4 rekordy jako podejrzane

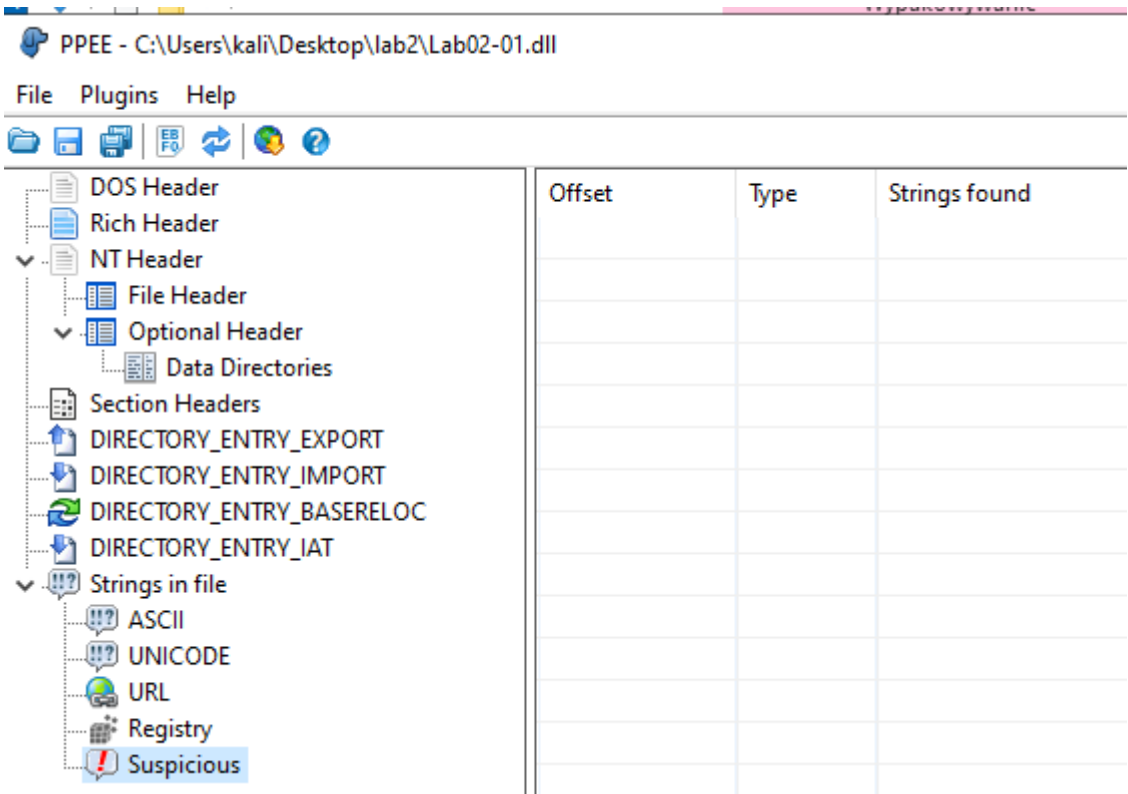
W zakładce String in file -> ASCII można również znaleźć wywołania funkcji bibliotek oraz z pozoru losowe ciągi znaków.

Offset	Strings recognized ASCII
000021E2	MSVCRT.dll
000021F0	_exit
000021F8	_XcptFilter
00002206	__p__initenv
00002216	__getmainargs
00002226	_initterm
00002232	__setusermatherr
00002246	_adjust_fdiv
00002256	__p__commode
00002266	__p__fmode
00002274	__set_app_type
00002286	_except_handler3
0000229A	_controlfp
000022A8	_stricmp
00003010	kerne132.dll
00003020	kernel32.dll
00003030	.exe
00003044	C:*
0000304C	C:\windows\system32\kerne132.dll
00003070	Kernel32.
0000307C	Lab01-01.dll
0000308C	C:\Windows\System32\Kernel32.dll
000030B0	WARNING_THIS_WILL_DESTROY_YOUR_MACHINE

Offset	Strings recognized ASCII
0000004D	!This program cannot be run in DOS
000000C8	Richm
000001E0	.text
00000207	`.rdata
0000022F	@.data
000010A8	UVWj
0000116C	ugh 0@
000011D5	_^[
000011ED	SUVW
000013A1	h00@
00001434	_^[
0000144A	SUVW
000014E7	h 0@
000014F6	D\$Pj
0000152C	I\$\'u
00001578	\$SQWR
000015AB	FxRVP
00001639	D\$\$3
000016B9	D\$8R
000016F2	t\$<f
000017E2	T\$PR
000017EA	hL0@
000017EF	h 0@

Dwa osobne rekordy biblioteki kernel32.dll i kerne132.dll najprawdopodobniej mają za zadanie podmianę oryginalnej biblioteki z zainfekowanym zamiennikiem

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



Program nie znalazł podejrzanych napisów, dlatego przejdę do próby ręcznego ich znalezienia

Offset	Strings recognized ASCII
00001390	_^[[]
0000210A	CloseHandle
00002118	Sleep
00002120	CreateProcessA
00002132	CreateMutexA
00002142	OpenMutexA
0000214E	KERNEL32.dll
0000215C	WS2_32.dll
0000216A	strncmp
00002172	MSVCRT.dll
00002180	free
00002188	_initterm
00002194	malloc
0000219E	_adjust_fdiv
00026010	exec
00026018	sleep
00026020	hello
00026028	127.26.152.13
00026038	SADFHUHF
00027008	/0I0[0h0p0
00027029	141G1[1I1
00027039	1Y2a2g2r2

Biblioteka ta powiela wiele napisów z poprzedniego pliku wykonywalnego (Lab02-01.exe) jednakże można znaleźć tutaj również pewien adres ip - **127.26.152.13**

Spróbowałem wyszukać powyższy adres w serwisie VirusTotal, jednakże nie znalazł on niczego podejrzanego

0

/ 84

10+ detected files embedding this IP address

127.26.152.13

loopback private

Community Score

DETECTION

DETAILS

RELATIONS

COMMUNITY

Security vendors' analysis

Do you want to automate checks?

Abusix	✓ Clean	Acronis	✓ Clean
ADMINUSLabs	✓ Clean	AICC (MONITORAPP)	✓ Clean
AlienVault	✓ Clean	alphaMountain.ai	✓ Clean
Antiy-AVL	✓ Clean	ArcSight Galaxy	✓ Clean
Avira	✓ Clean	benkow.cc	✓ Clean
Bfore.Ai PreCrime	✓ Clean	BitDefender	✓ Clean
Bkav	✓ Clean	Blueliv	✓ Clean
Certego	✓ Clean	Chong Lua Dao	✓ Clean

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

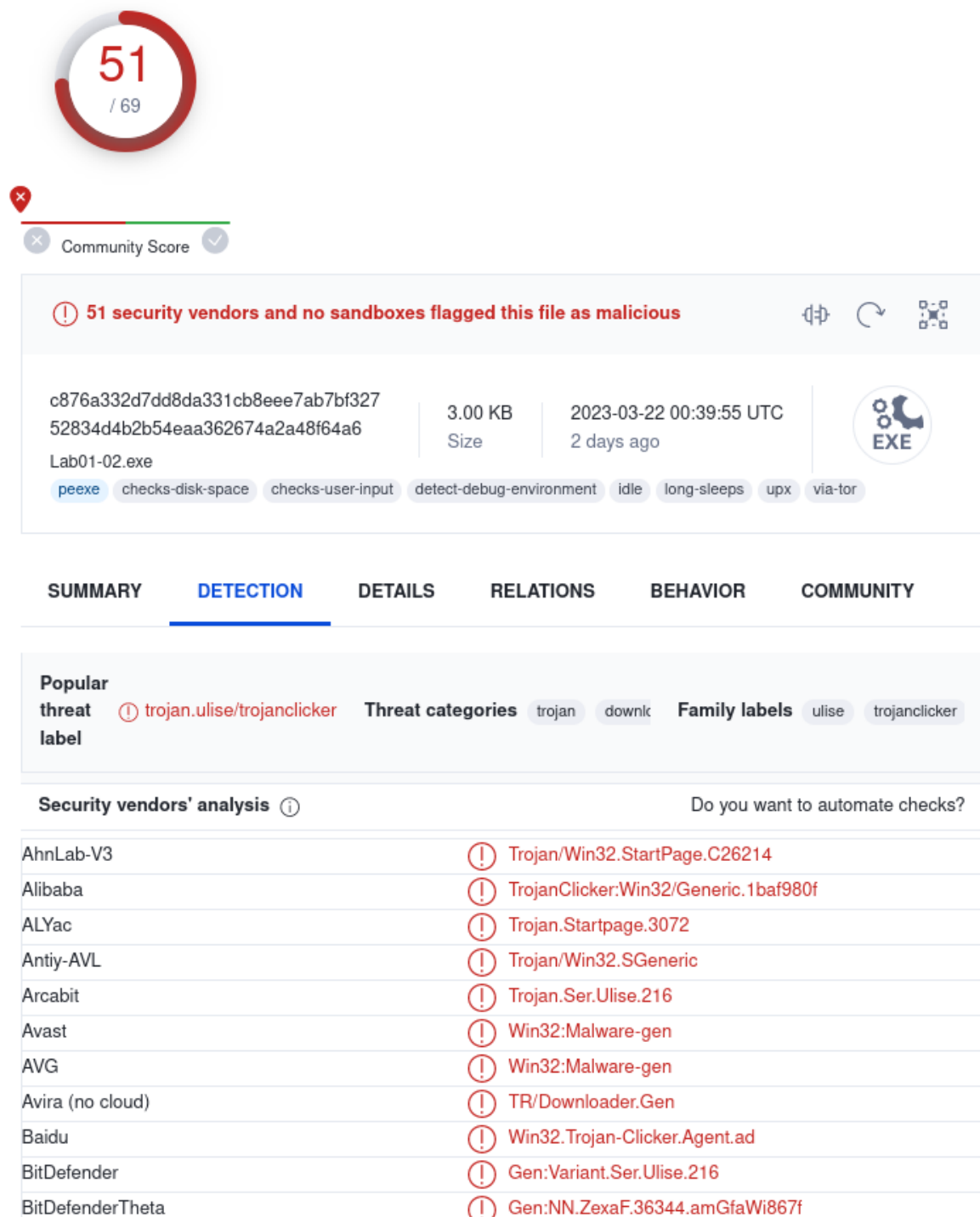
Program wykonywalny ma za zadanie podmianę biblioteki kernel32.dll na kerne132.dll. Następnie najprawdopodobniej łączy się z jakimś serwerem i próbuje nawiązać komunikację. Po drodze uruchamia on wiele innych procesów niezbędnych do działania (poprzez funkcję CreateProcess)

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.

```
arek@Arek:~/studia/malware/lab2/binaries$ md5sum Lab02-02.exe
8363436878404da0ae3e46991e355b83  Lab02-02.exe
```



W skanowaniu wyszło, że próbka jest bardzo szkodliwa - aż 51 na 69 vendor'ów stwierdziło ją jako złośliwą - jest to ewidenty Trojan.

History ⓘ	
Creation Time	2011-01-19 16:10:41 UTC
First Seen In The Wild	2010-11-20 23:29:33 UTC
First Submission	2011-07-02 17:02:09 UTC
Last Submission	2023-03-24 09:58:31 UTC
Last Analysis	2023-03-22 00:39:55 UTC

Próbka została dodana już w 2011 roku - a tym samym była już analizowana

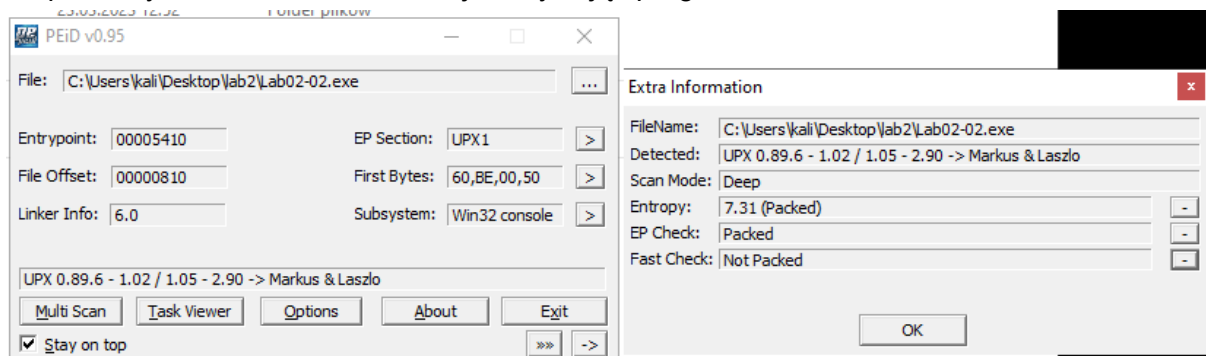
2. Sprawdź, czy coś świadczy o tym, że plik jest spakowany lub zaciemniony? Spróbuj go rozpakować.

TrID	UPX compressed Win32 Executable (34.7%)	Win32 EXE Yoda's Crypter (34.1%)	Win32 Dynamic Link Library (generic) (8.4%)	Win16 NE executable (generic) (6.4%)	Win32 Executable (generic) (5.7%)
------	---	----------------------------------	---	--------------------------------------	-----------------------------------

Plik jest w dość znaczącej części w formie skompresowanych plików wykonywalnych (UPX Win32)

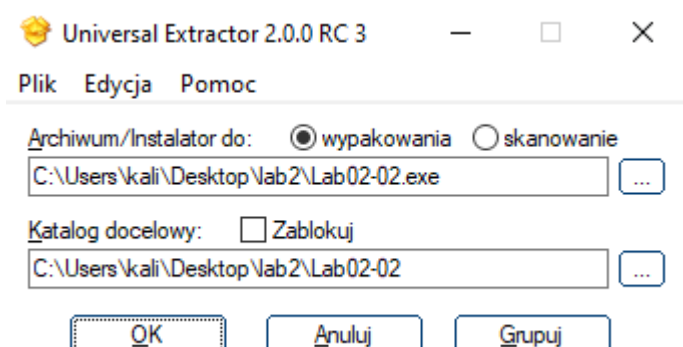
PEiD packer: UPX v0.89.6 - v1.02 / v1.05 -v1.24 -> Markus & Laszlo [overlay]

Do podobnych wniosków można dojść używając programu **PEiD**



Plik jest spakowany

Próbuję go odpakować programem **Universal Extractor**



Lab02-02.exe	19.01.2011 11:10	Aplikacja	3 KB
Lab02-02_wypakowany.exe	19.01.2011 11:10	Aplikacja	16 KB

Operacja zakończona pomyślnie

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

Lab02-02_wypakowany.exe:

Offset	Name	Func. Count	Bound?	OriginalFirstThunk	TimeDateStamp	Forwarder
208C	KERNEL32.DLL	9	FALSE	0	0	0
20A0	ADVAPI32.dll	3	FALSE	0	0	0
20B4	MSVCRT.dll	13	FALSE	0	0	0
20C8	WININET.dll	2	FALSE	0	0	0

Call via	Name	Ordinal	Original Thunk	Thunk	Forwarder
2044	__p__initenv	-	-	2296	-
2048	__getmainargs	-	-	22A6	-
204C	__initterm	-	-	22B6	-
2050	__setusermatherr	-	-	22C2	-
2054	__adjust_fdiv	-	-	22D4	-
2058	__p__commode	-	-	22E2	-
205C	__p__fmode	-	-	22F0	-

Najciekawsze importy

KERNEL32.dll:

- CreateMutexA
- CreateThread
- SetWaitableTimer

ADVAPI32.dll:

- CreateServiceA
- StartServiceCtrlDispatcherA
- OpenSCManagerA

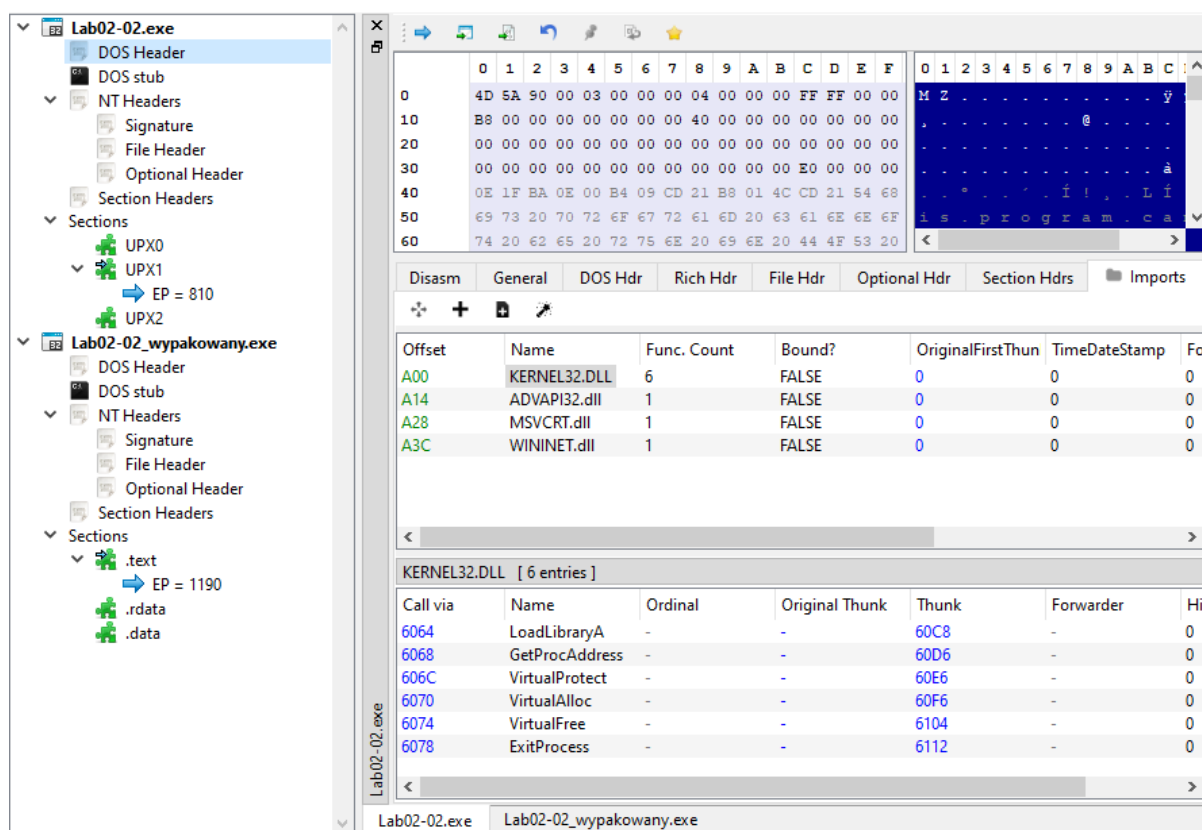
MSVCRT.dll:

- initterm
- setupusermatherr
- exit

WININET.dll:

- InternetOpenUrlA
- InternetOpenA

Lab02-02.exe



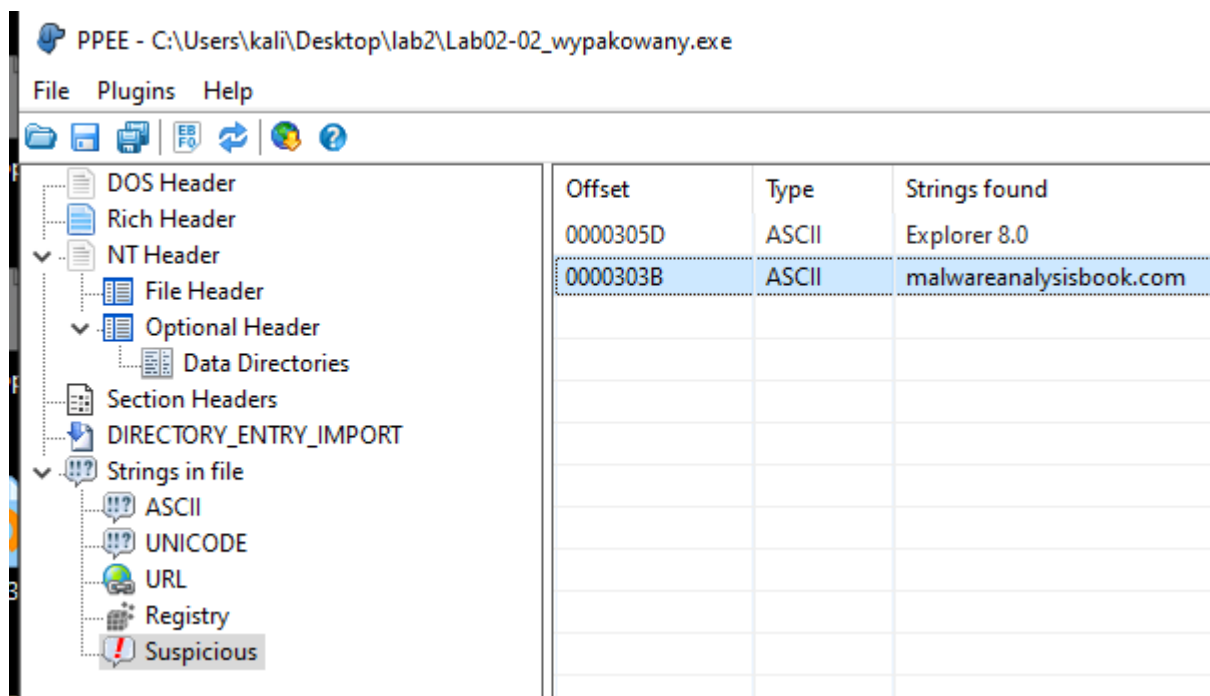
Wypakowany posiada znacznie więcej funkcji, pomimo że importują te same biblioteki.

Tabela różnic (ilość importowanych funkcji):

	KERNEL32.dll	ADVAPI32.dll	MSVCRT.dll	WININET.dll
Lab02-02.exe	6	1	1	1
Lab02-02_wypakowany.exe	9	3	13	2

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

Używam programu PPEE



W zakładce Suspicious jest informacja o pewnej domenie.

Niestety program nie odnalazł żadnego adresu ip - dlatego nie do końca wiadomo, czy powyższa domena to np. źródło pochodzenia pliku czy adres serwera z którym ów wirus próbuje się połączyć

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.

```
arek@Arek:~/studia/malware/lab2/binaries$ md5sum Lab02-03.exe
9c5c27494c28ed0b14853b346b113145 Lab02-03.exe
```




Community Score

58 security vendors and no sandboxes flagged this file as malicious



7983a582939924c70e3da2da80fd3352e
bc90de7b8c4c427d484ff4f050f0aec
Lab01-03.exe

4.64 KB
Size

2023-03-12 17:25:10 UTC
11 days ago



peexe checks-user-input via-tor overlay runtime-modules detect-debug-environment long-sleeps
direct-cpu-clock-access fsg

SUMMARY

DETECTION

DETAILS

RELATIONS

BEHAVIOR

COMMUNITY

Popular

threat
label

trojan.graftor/genome

Threat categories

trojan

spywar

Family labels

graftor

genome

Security vendors' analysis

Do you want to automate checks?

AhnLab-V3	Trojan/Win.Generic.R427327
Alibaba	TrojanClicker.Win32/Tnega.3bb840a6
ALYac	Gen:Variant.Graftor.968808
Antiy-AVL	Trojan/Win32.SGeneric
Arcabit	Trojan.Graftor.DEC868
Avast	Win32:Malware-gen
AVG	Win32:Malware-gen
Avira (no cloud)	TR/Clicker.knmor
Baidu	Win32.Trojan-Clicker.Agent.z
BitDefender	Gen:Variant.Graftor.968808

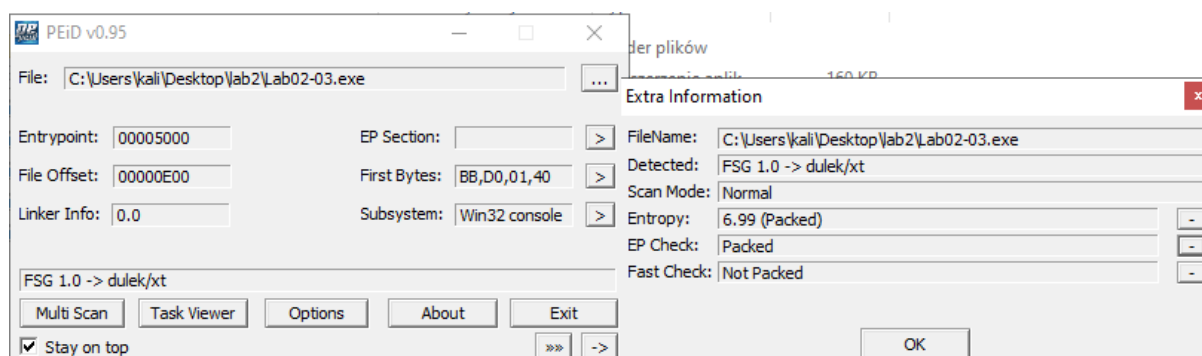
Program jest ewidentnie złośliwy

History ⓘ

First Seen In The Wild	2011-03-26 06:54:39 UTC
First Submission	2011-07-04 22:00:08 UTC
Last Submission	2023-03-24 10:32:42 UTC
Last Analysis	2023-03-12 17:25:10 UTC

Również został przeanalizowany w serwisie VirusTotal w 2011 roku

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?



Plik jest spakowany

```
arek@Arek:~/Downloads/upx-4.0.2-amd64_linux$ ./upx -d -o file ~/studio/malware/lab2/binaries/Lab02-03.exe
Ultimate Packer for eXecutables
Copyright (C) 1996 - 2023
UPX 4.0.2 Markus Oberhumer, Laszlo Molnar & John Reiser Jan 30th 2023

File size      Ratio      Format      Name
-----
upx: /home/arek/studio/malware/lab2/binaries/Lab02-03.exe: NotPackedException: not packed by UPX
Unpacked 0 files.
```

Niestety nie udało się go rozpakować, ponieważ nie został spakowany przez UPX

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

TimeDateStamp	00000000	Thu, 01 Jan 1970 00:00:00 UTC (19442 days, 14.91 hours ago)
---------------	----------	---

Wynik nie wydaje się być prawdziwy. Prawdopodobnie został uszkodzony lub celowo nadpisany/wyzerowany

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?

Name RVA	Name	OriginalFirstThunk	TimeDate Stamp	ForwarderChain	FirstThunk	Description (Read from file)
00005134	KERNEL32.dll	0000511C	00000000	00000000	00005128	Biblioteka DLL klienta Windows NT BASE API

Plik importuje zaledwie jedną bibliotekę - KERNEL32.dll

Dostępne są jednak importy funkcji

Member	Value	Comment
e_magic	5A4D	MZ
e_cblp	0090	
e_cp	0003	
e_crlc	0000	
e_cparhdr	0004	
e_minalloc	0000	
e_maxalloc	FFFF	
e_ss	0000	
e_sp	00B8	
e_csum	0000	
e_ip	0000	
e_cs	0000	
e_lfarlc	0040	
e_ovno	0000	
e_res[0]	0000	
e_res[1]	0000	
e_res[2]	0000	
e_res[3]	0000	
e_oemid	0000	
e_oeminfo	0000	
e_res2[0]	0000	
e_res2[1]	0000	
e_res2[2]	0000	

Więc nie, nie idzie sprawdzić w taki sam sposób importów jak w Laboratorium 1.1

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

Offset	Strings recognized ASCII
0000004D	!Windows Program
0000020F	`.rdata
00000237	@.data
00000F34	KERNEL32.dll
00000F42	LoadLibraryA
00000F50	GetProcAddress
00001020	":LI
00001025	3Bt> O
0000102E	VQ{8
00001043	2]<,M
00001060	P@M^
00001089	S>VW
000010BE	AQ=h
00001145	I*G9>
0000119B	e%nN
000011C5	ole32.vd
000011D2	Init
000011DD	FoCr
000011ED	U!!C
000011F4	}OLEAUTLA
0000120A	IMSVCR71"b
00001215	_getmas
0000121D	yrCs
00001235	P2r3Us
0000123F	p vuy
00001249	fmod
0000125B	xF*I

Brak rzucających się połączeniach z siecią - być może są w postaci zaciemnionej

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.

```
arek@Arek:~/studia/malware/lab2/binaries$ md5sum Lab02-04.exe
625ac05fd47adc3c63700c3b30de79ab  Lab02-04.exe
```

53

/ 69

Community Score

53 security vendors and 1 sandbox flagged this file as malicious

0fa1498340fca6c562cfa389ad3e93395f

44c72fd128d7ba08579a69aaf3b126

Lab01-04.exe

peexe

idle

via-tor

armadillo

checks-user-input

36.00 KB

Size

2023-03-18 21:20:03 UTC

6 days ago

EXE

SUMMARY

DETECTION

DETAILS

RELATIONS

BEHAVIOR

COMMUNITY 30 +

Crowdsourced IDS rules ⓘ

HIGH 0

MEDIUM 0

LOW 1

INFO 0

⚠

Matches rule **ET INFO TLS Handshake Failure**

Dynamic Analysis Sandbox Detections ⓘ

⚠

The sandbox **Lastline** flags this file as: **MALWARE**

Popular threat label ⓘ trojan.cerbu/genericrxew

Threat categories

trojan

downl

Family labels

cerbu

genericrxew

Security vendors' analysis ⓘ

Do you want to automate checks?

Alibaba

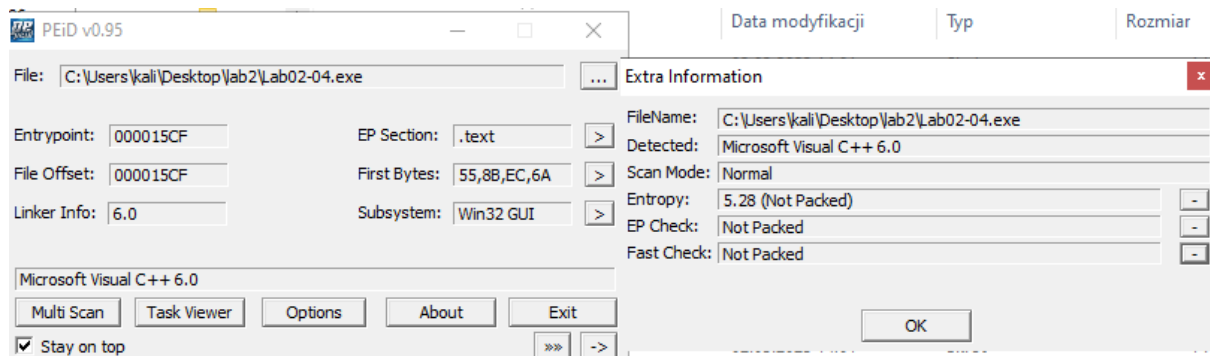
⚠ TrojanDownloader:Win32/DownLdr.080f6485

ALYac

⚠ Gen:Variant.Cerbu.64782

Sprawa wygląda bardzo podobnie jak w przypadku poprzednich plików

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



Plik nie został spakowany

3. Kiedy ten plik został skompilowany?

Member	Value	Comment
Machine	014C	Intel 386
NumberOfSections	0004	
TimeDateStamp	5D69A2B3	Fri, 30 Aug 2019 22:26:59 UTC (1303 days, 16.73 hours ago)
PointerToSymbolTable	00000000	
NumberOfSymbols	00000000	
SizeOfOptionalHeader	00E0	
Characteristics	010F	

30 września 2019

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?

Name RVA	Name	OriginalFirstThunk	TimeDate Stamp	ForwarderChain	FirstThunk	Description (Read from file)
0000228E	KERNEL32.dll	00002104	00000000	00000000	00002010	Biblioteka DLL klienta Windows NT BASE API
000022E0	ADVAPI32.dll	000020F4	00000000	00000000	00002000	Advanced Windows 32 Base API
000022FA	MSVCRT.dll	00002148	00000000	00000000	00002054	Windows NT CRT DLL

Plik importuje 3 biblioteki oraz szereg następujących funkcji:

- KERNEL32.dll

000021CE	000021CE	013E	GetProcAddress
000021E0	000021E0	01C2	LoadLibraryA
000021F0	000021F0	02D3	WinExec
000021FA	000021FA	02DF	WriteFile
00002206	00002206	0034	CreateFileA
00002214	00002214	0295	SizeofResource
000021B8	000021B8	0046	CreateRemoteThread
00002236	00002236	00A3	FindResourceA
00002246	00002246	0126	GetModuleHandleA
0000225A	0000225A	017D	GetWindowsDirectoryA
00002272	00002272	01DD	MoveFileA
0000227E	0000227E	0165	GetTempPathA
000021A4	000021A4	00F7	GetCurrentProcess
00002196	00002196	01EF	OpenProcess
00002188	00002188	001B	CloseHandle
00002226	00002226	01C7	LoadResource

- ADVAPI32.dll

000022CC	000022CC	0142	OpenProcessToken
000022B4	000022B4	00F5	LookupPrivilegeValueA
0000229C	0000229C	0017	AdjustTokenPrivileges

- MSVCRT.dll

000022EE	000022EE	01AE	_snprintf
00002306	00002306	00D3	_exit
0000230E	0000230E	0048	_XcptFilter
0000231C	0000231C	0249	exit
00002324	00002324	0064	__p__initenv
00002334	00002334	0058	__getmainargs
00002344	00002344	010F	_initterm
00002350	00002350	0083	__setusermatherr
00002364	00002364	009D	_adjust_fdiv
00002374	00002374	006A	__p__commode
00002384	00002384	006F	__p__fmode
00002392	00002392	0081	__set_app_type
000023A4	000023A4	00CA	_except_handler3
000023B8	000023B8	00B7	_controlfp
000023C6	000023C6	01C1	_stricmp

Analizę można przeprowadzić w sposób podobny do sposobu z Laboratorium 1.1

Program najprawdopodobniej używa funkcji z biblioteki KERNEL32.dll do przenoszenia złośliwych plików oraz tworzenia procesów wraz z odpowiednimi uprawnieniami (biblioteka ADVAPI32.dll)

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

00007084	\\system32\\wupdmgrd.exe
0000709C	%s%s
000070A4	http://www.practicalmalwareanalysis.com/updater.exe

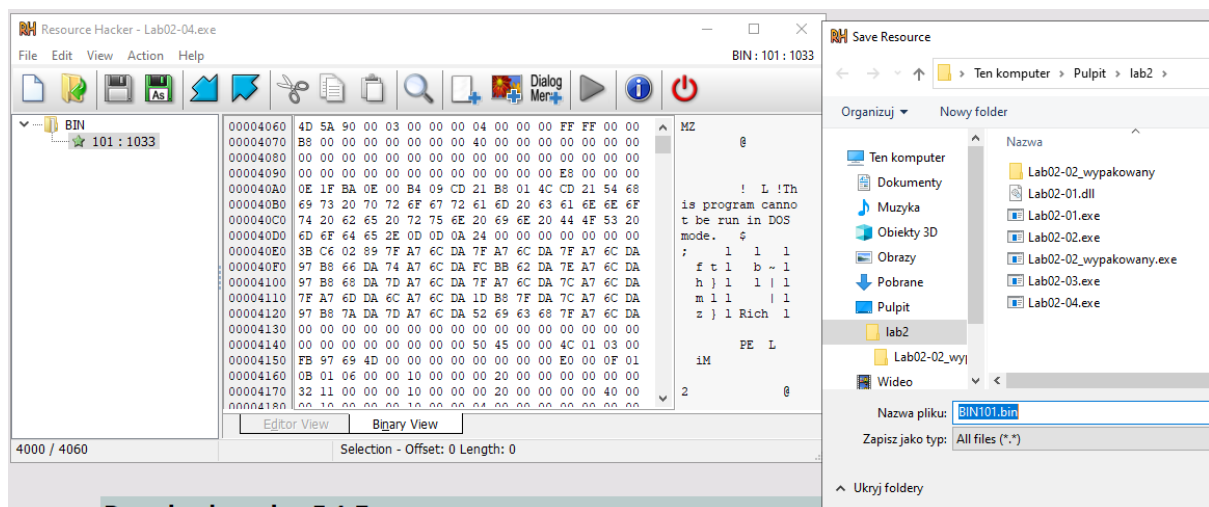
Podobnie jak w jednym z poprzednich laboratoriów - program nie odnalazł żadnego adresu ip - dlatego nie do końca wiadomo, czy powyższa domena to np. źródło pochodzenia pliku czy adres serwera z którego wirus próbuje coś pobrać

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

0000616A	GetWindowsDirectoryA
00006182	WinExec
0000618C	GetTempPathA
0000619A	KERNEL32.dll
000061AA	URLDownloadToFileA
000061BE	urlmon.dll

Tak, próbuje on pobrać plik z jakiegoś serwera

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?



Po otwarciu pliku w programie PE-bear zmieniła się jedna funkcja ADVAPI32.dll -> urlmon.dll

Offset	Name	Func. Count	Bound?	OriginalFirstThun	TimeDateStamp
2064	KERNEL32.dll	3	FALSE	20B4	0
2078	urlmon.dll	1	FALSE	2100	0
208C	MSVCRT.dll	14	FALSE	20C4	0

Call via	Name	Ordinal	Original Thunk	Thunk	Forwarder
204C	URLDownloadT...	-	2148	2148	-

Zawiera ona wcześniej opisaną funkcję - służącą pobieraniu zasobu z serwera

Biblioteka KERNEL32.dll posiada teraz znacznie mniej importowanych funkcji

Offset	Name	Func. Count	Bound?	OriginalFirstThun	TimeDateStamp
2064	KERNEL32.dll	3	FALSE	20B4	0
2078	urlmon.dll	1	FALSE	2100	0
208C	MSVCRT.dll	14	FALSE	20C4	0

Call via	Name	Ordinal	Original Thunk	Thunk	Forwarder
2000	WinExec	-	2120	2120	-
2004	GetTempPathA	-	212A	212A	-
2008	GetWindowsDir...	-	2108	2108	-

MSVCRT.dll wydaje się pozostać niezmienną

Najprawdopodobniej wirus próbował pierwotnie ukryć swoje działanie w formie importowania innych bibliotek i funkcji - po wyodrębnieniu pokazuje swoje prawdziwe działanie - pobranie pliku z serwera oraz później uruchomienie go za pomocą funkcji WinExec