Train your own antivirus with Machine Learning

@xufuou

Who am I ?

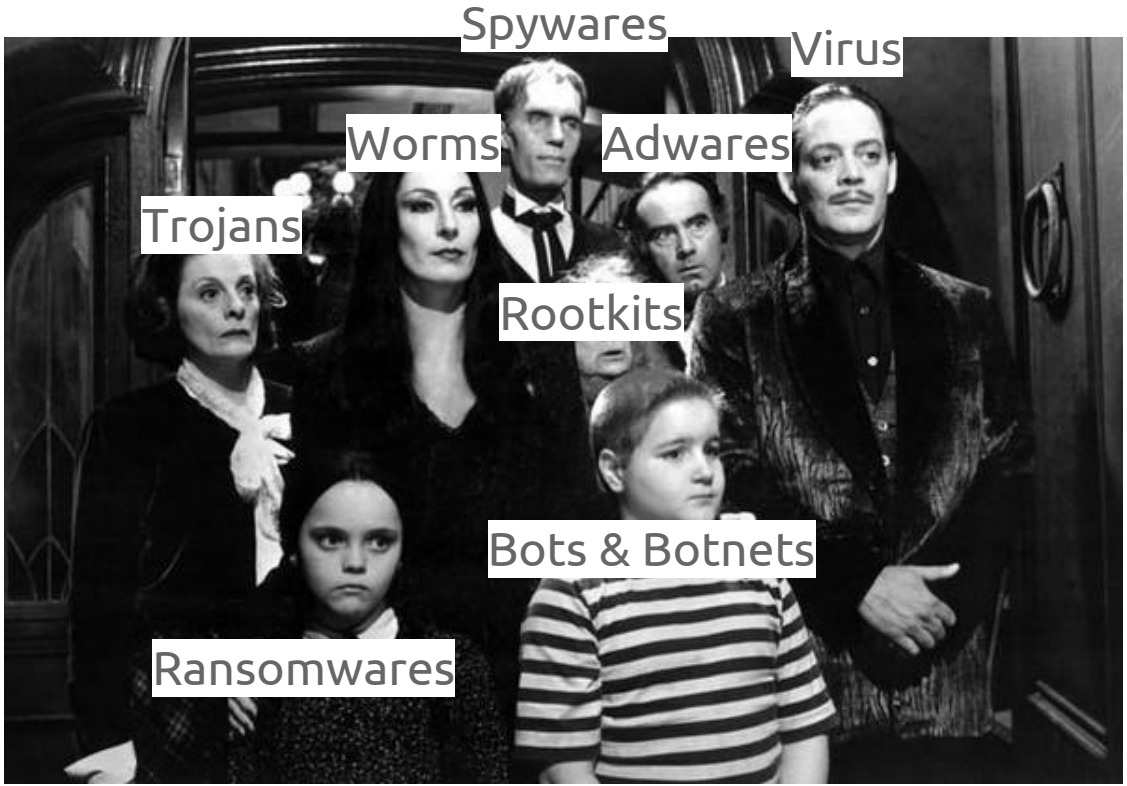
- Application Security Analyst at Checkmarx.
- Working with ConvNets since 2015.
- I do like a problem that keep me awake at night.


YOUR COMPUTER HAS A VIRUS
IT'S CALLED WINDOWS

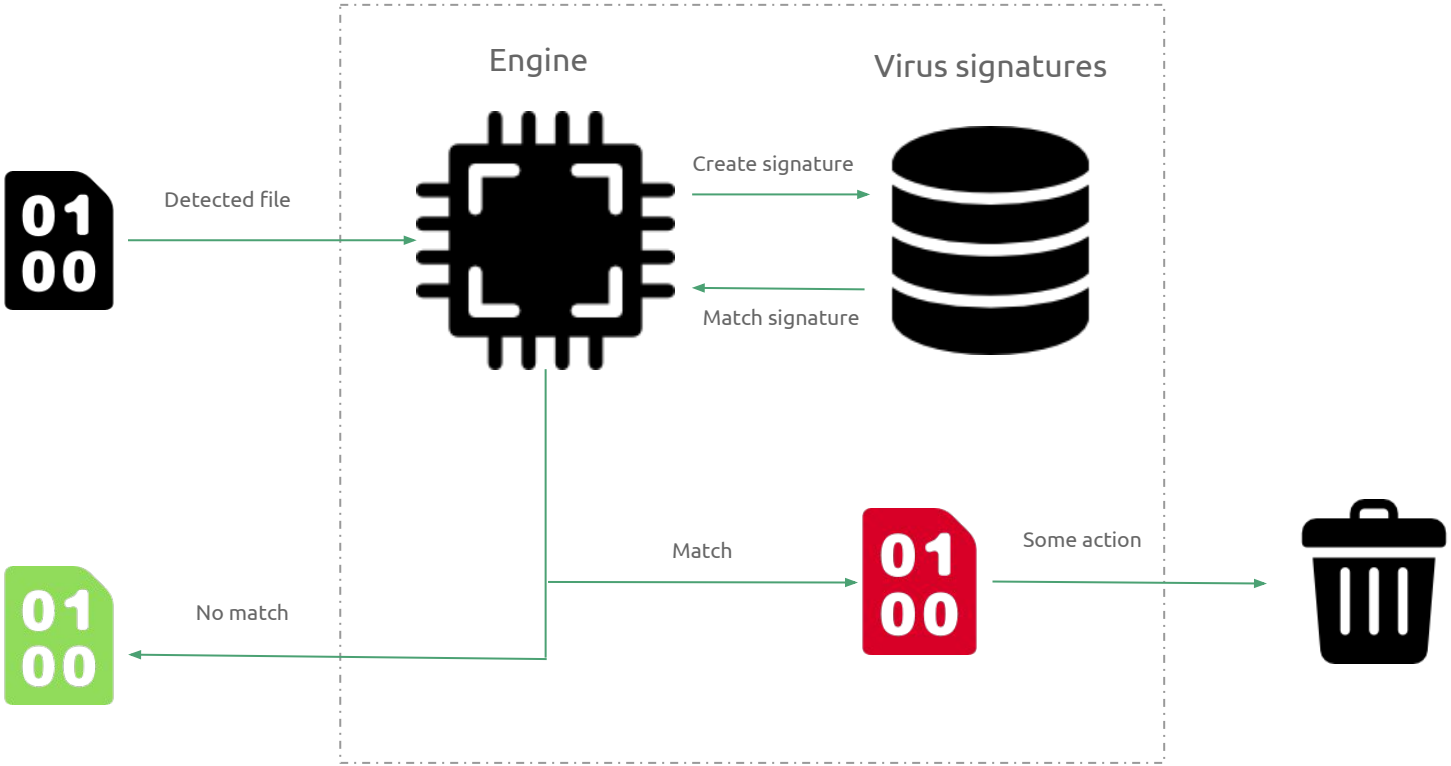
"a code that recursively replicates a possibly evolved copy of itself”

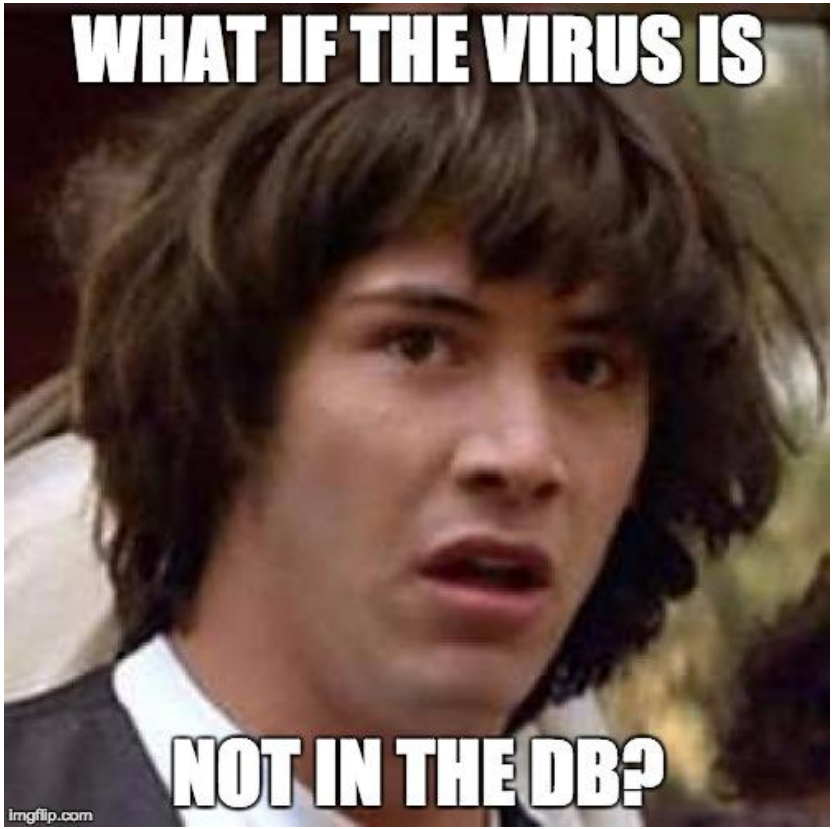
Definition of virus by Péter Ször, 2005

Meet the family



Signature matching - conceptual representation




WHAT IF THE VIRUS IS
NOT IN THE DB?
imgflip.com

How good are signature based approaches?

1. Discovery

2. Research

3. Validation

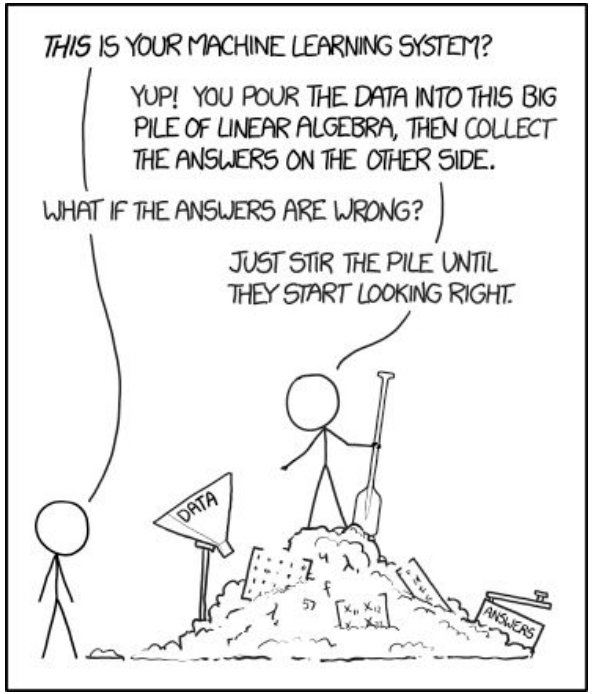
4. Update

How long until having a valid signature ?

Polymorphic virus and packing?

Heuristics and Behavior Based Antivirus

Machine Learning

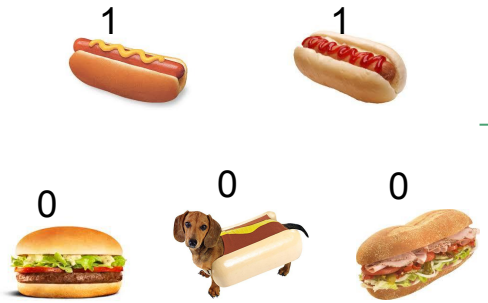


"Machine Learning is programming computers to optimize a performance criterion using example data or past experience."

Ethem Alpaydin

Traditional supervised learning - training

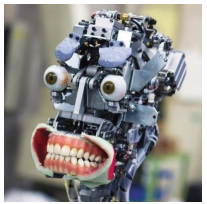
Labeled Data



Feature extraction

Sausage detector
Bread detector

Model



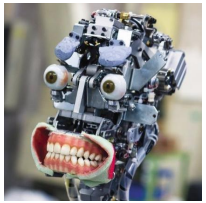
Traditional supervised learning

Unknow input → Feature extraction → Model → Prediction

→

Sausage detector
Bread detector

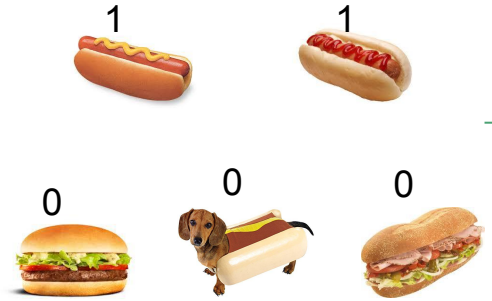
→



→ 1

Deep learning - Bonus

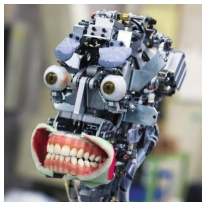
Labeled Data



Feature extraction

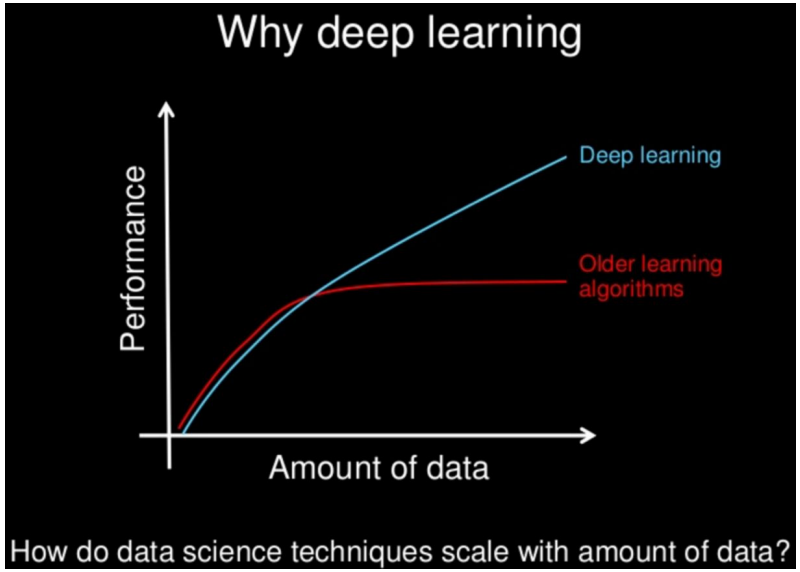
Sausage detector
Bread detector

Model

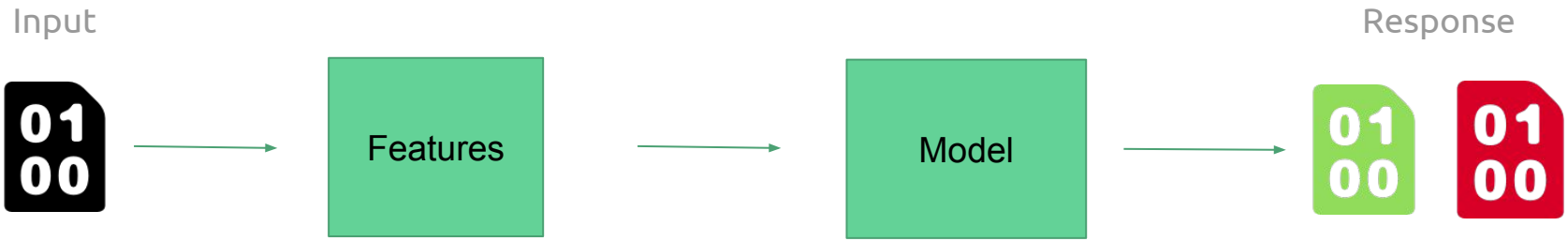


Training

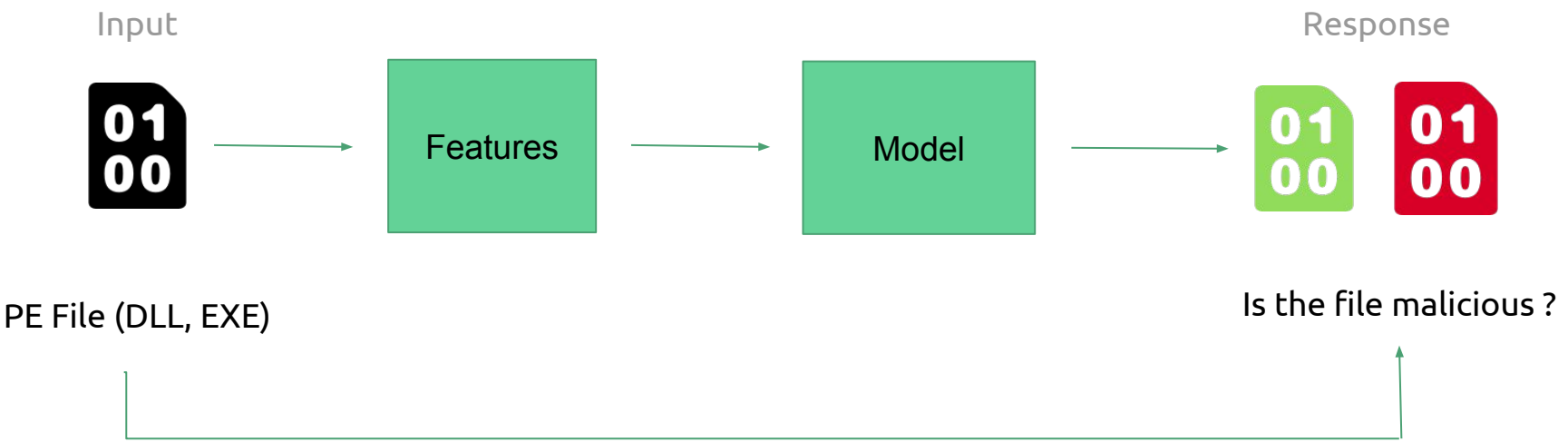
Deep learning - Bonus



Applying supervised ML (to malware detection)



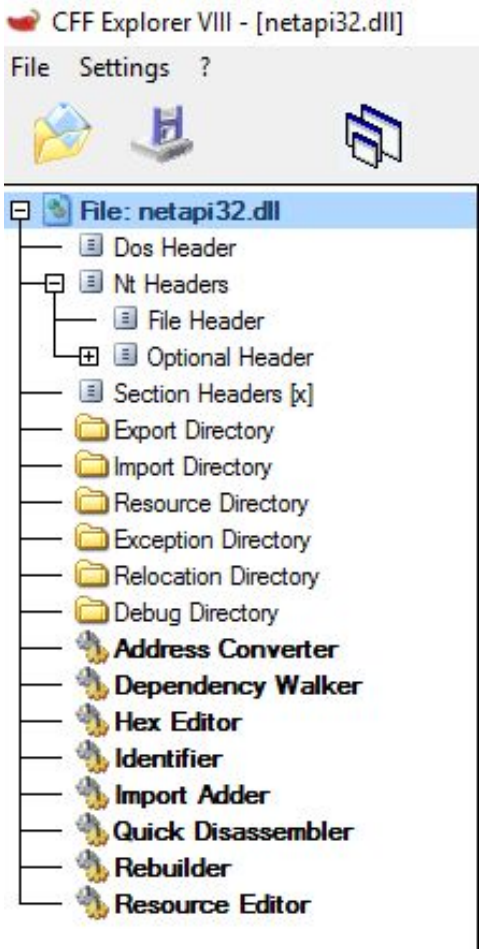
Applying supervised ML (to malware detection)

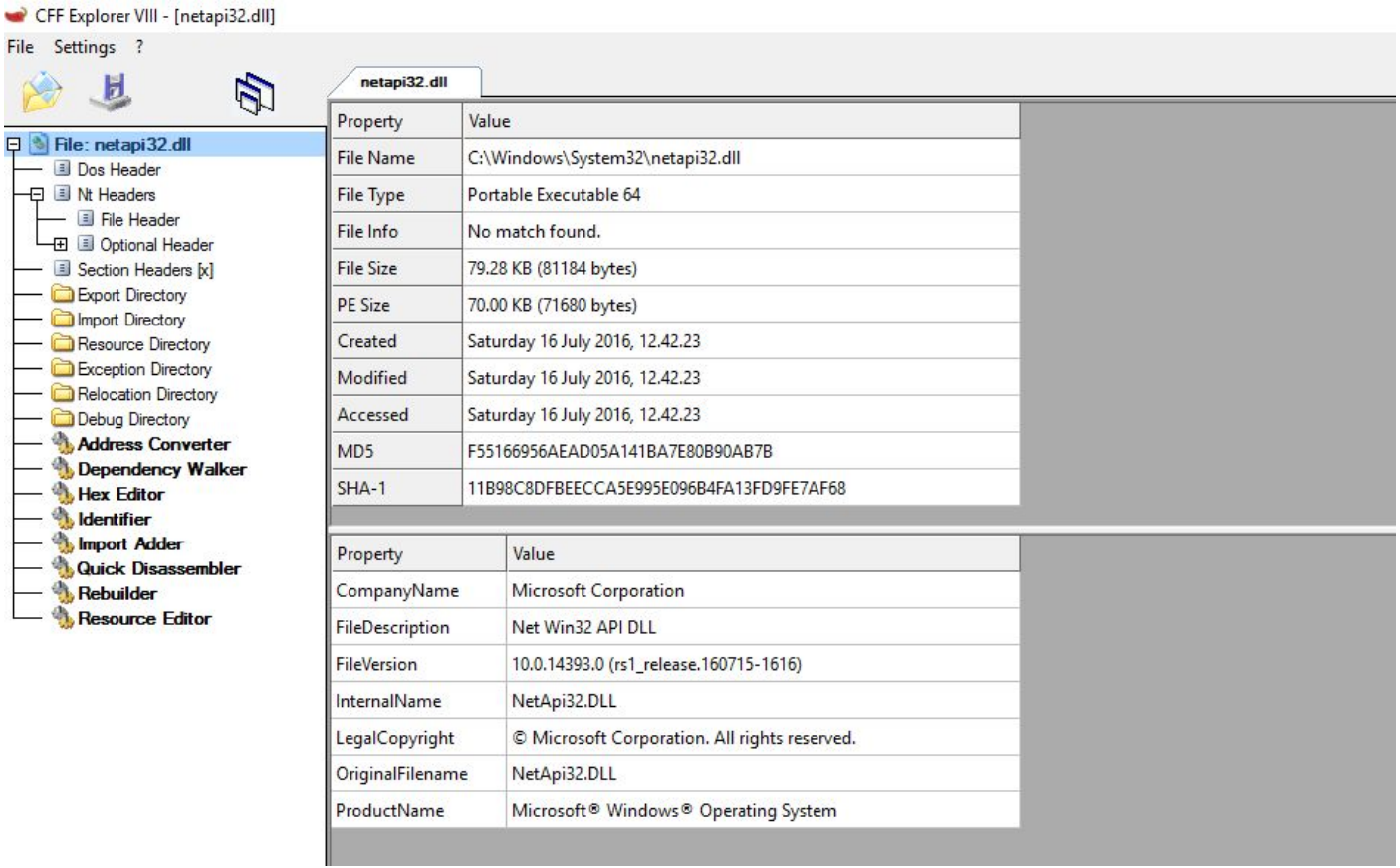


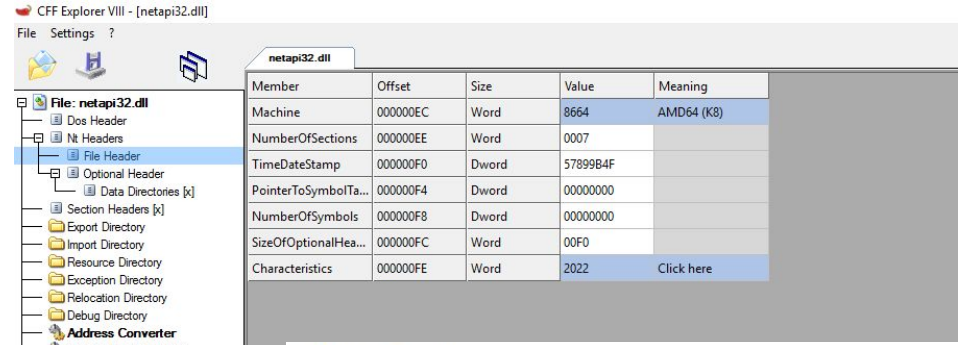
Feature extraction

The PE file format is a data structure that contains the information necessary for the Windows OS loader to manage the wrapped executable code.

- Version information
- DOS header
- COFF (NT) header
- PE optional header
- Data directories
- Import table
- Section table
- Resources




CFF Explorer VIII - [netapi32.dll]
File Settings ?
File: netapi32.dll
Dos Header
Nt Headers
File Header
Optional Header
Section Headers [x]
Export Directory
Import Directory
Resource Directory
Exception Directory
Relocation Directory
Debug Directory
Address Converter
Dependency Walker
Hex Editor
Identifier
Import Adder
Quick Disassembler
Rebuilder
Resource Editor
netapi32.dll
Property Value
File Name C:\Windows\System32\netapi32.dll
File Type Portable Executable 64
File Info No match found.
File Size 79.28 KB (81184 bytes)
PE Size 70.00 KB (71680 bytes)
Created Saturday 16 July 2016, 12.42.23
Modified Saturday 16 July 2016, 12.42.23
Accessed Saturday 16 July 2016, 12.42.23
MD5 F55166956AEAD05A141BA7E80B90AB7B
SHA-1 11B98C8DFBEECCA5E995E096B4FA13FD9FE7AF68
Property Value
CompanyName Microsoft Corporation
FileDescription Net Win32 API DLL
FileVersion 10.0.14393.0 (rs1_release.160715-1616)
InternalName NetApi32.DLL
LegalCopyright © Microsoft Corporation. All rights reserved.
OriginalFilename NetApi32.DLL
ProductName Microsoft® Windows® Operating System



NumberOfSections

The number of sections. This indicates the size of the section table, which immediately follows the headers. Note that the Windows loader limits the number of sections to 96.

TimeDateStamp

The low 32 bits of the time stamp of the image. This represents the date and time the image was created by the linker. The value is represented in the number of seconds elapsed since midnight (00:00:00), January 1, 1970, Universal Coordinated Time, according to the system clock.

PointerToSymbolTable

The offset of the symbol table, in bytes, or zero if no COFF symbol table exists.

NumberOfSymbols

The number of symbols in the symbol table.

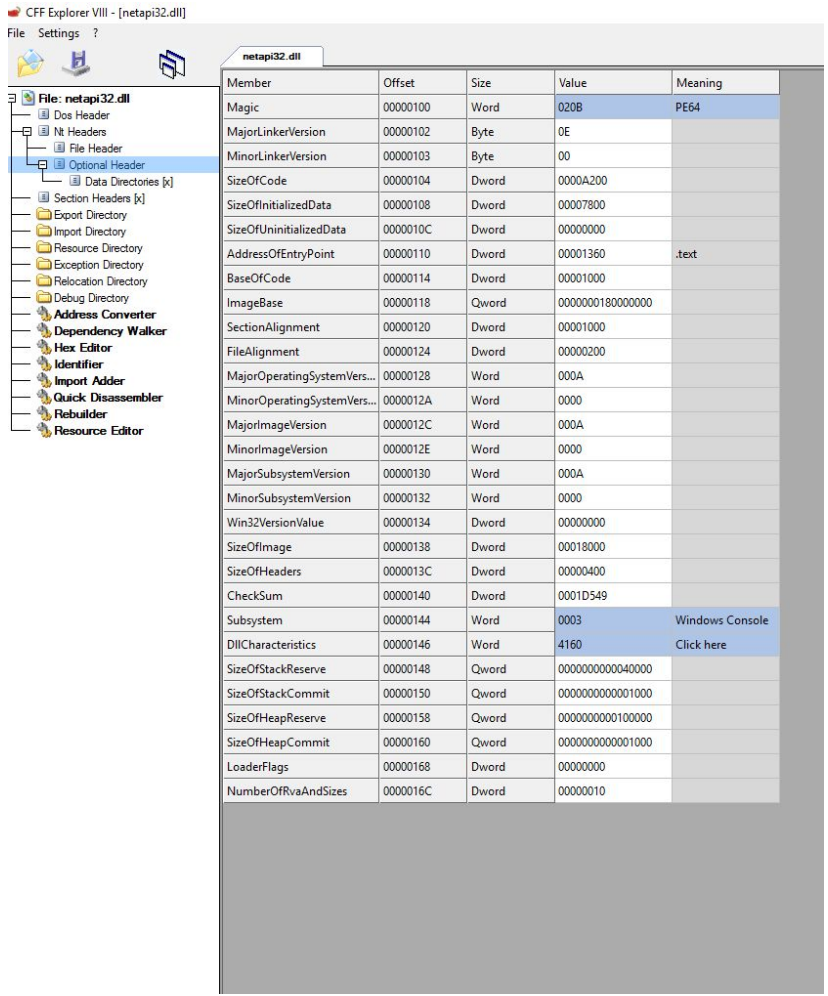
SizeOfOptionalHeader

The size of the optional header, in bytes. This value should be 0 for object files.

Characteristics

The characteristics of the image. This member can be one or more of the following values.

COFF header format from Windows Documentation



Member	Offset	Size	Value	Meaning
Magic	00000100	Word	020B	PE64
MajorLinkerVersion	00000102	Byte	0E	
MinorLinkerVersion	00000103	Byte	00	
SizeOfCode	00000104	Dword	0000A200	
SizeOfInitializedData	00000108	Dword	00007800	
SizeOfUninitializedData	0000010C	Dword	00000000	
AddressOfEntryPoint	00000110	Dword	00001360	.text
BaseOfCode	00000114	Dword	00001000	
ImageBase	00000118	Qword	0000000180000000	
SectionAlignment	00000120	Dword	00001000	
FileAlignment	00000124	Dword	00000200	
MajorOperatingSystemVers...	00000128	Word	000A	
MinorOperatingSystemVers...	0000012A	Word	0000	
MajorImageVersion	0000012C	Word	000A	
MinorImageVersion	0000012E	Word	0000	
MajorSubsystemVersion	00000130	Word	000A	
MinorSubsystemVersion	00000132	Word	0000	
Win32VersionValue	00000134	Dword	00000000	
SizeOfImage	00000138	Dword	00018000	
SizeOfHeaders	0000013C	Dword	00000400	
CheckSum	00000140	Dword	0001D549	
Subsystem	00000144	Word	0003	Windows Console
DllCharacteristics	00000146	Word	4160	Click here
SizeOfStackReserve	00000148	Qword	0000000000040000	
SizeOfStackCommit	00000150	Qword	0000000000001000	
SizeOfHeapReserve	00000158	Qword	0000000000100000	
SizeOfHeapCommit	00000160	Qword	0000000000001000	
LoaderFlags	00000168	Dword	00000000	
NumberOfRvaAndSizes	0000016C	Dword	00000010	

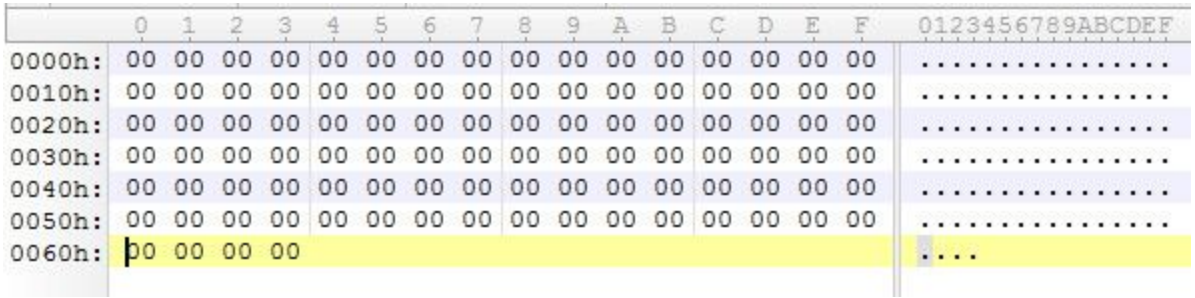
Can we detect packing?

Shannon's entropy equation:

$$H(X) = -\sum_{i=0}^{N-1} p_i \log_2 p_i$$

*p*i is the probability of a given symbol
N is the number of unique symbols

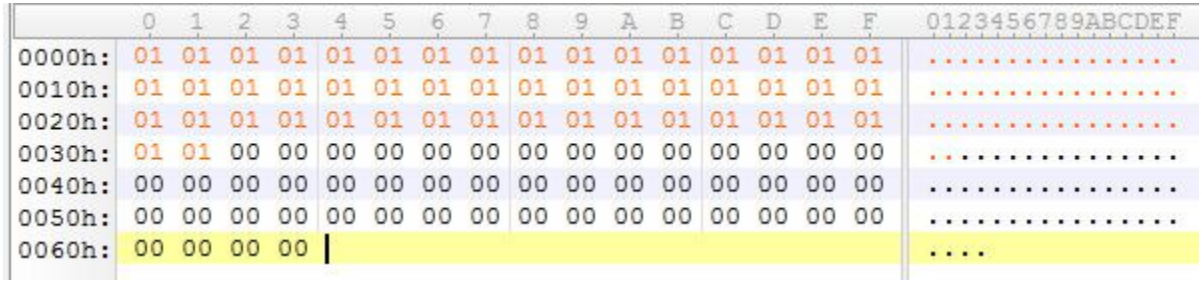
Case 1 - file with 100 bytes and value 0



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0010h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0020h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0030h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0040h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0050h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0060h:	00	00	00	00													....

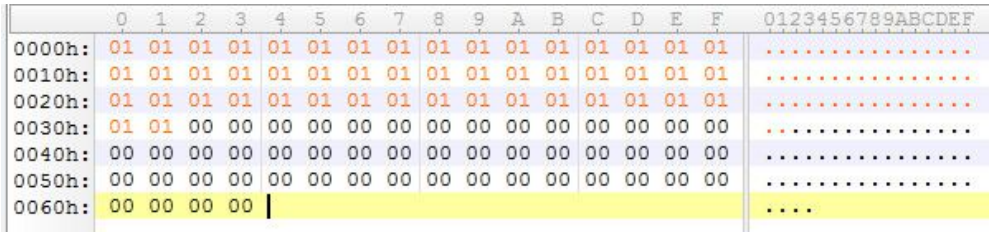
Entropy = 0

Case 2 - 100 byte file filled with half 0 and half 1



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	....
0010h:	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	....
0020h:	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	....
0030h:	01	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0040h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0050h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0060h:	00	00	00	00													....

Case 2 - 100 byte file filled with half zeros and half ones



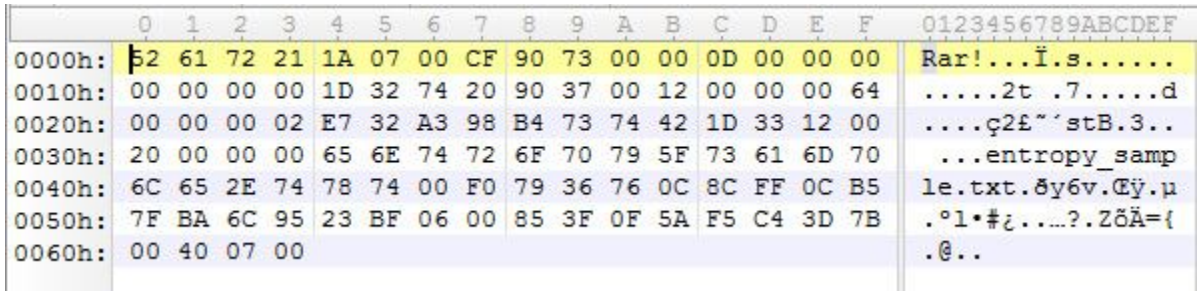
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
0010h:	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
0020h:	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	.....
0030h:	01	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0040h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0050h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0060h:	00	00	00	00													....

Shannon's entropy equation:

$$H(X) = -\sum_{i=0}^{N-1} p_i \log_2 p_i$$

Entropy = - (0.5*log2(0.5)+0.5*log2(0.5)) = -(-0.5-0.5) = 1

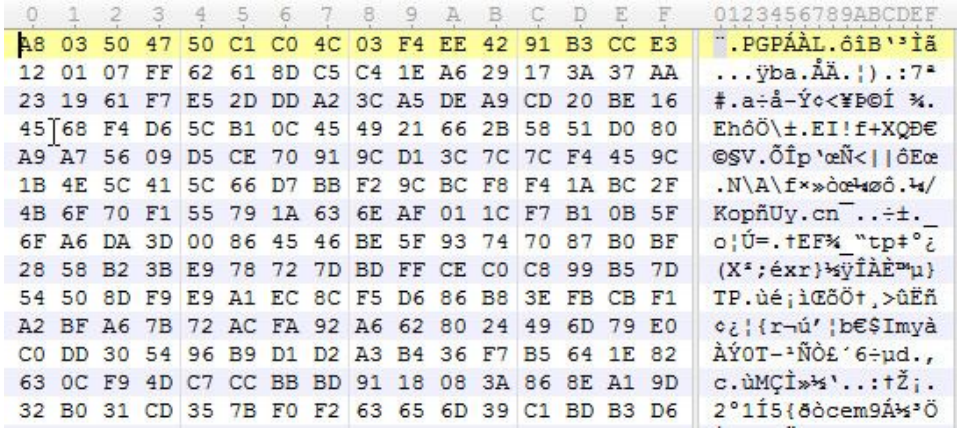
Case 3 - compress case 2 with Rar



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	52	61	72	21	1A	07	00	CF	90	73	00	00	0D	00	00	00	Rar!...Ï.s.....
0010h:	00	00	00	00	1D	32	74	20	90	37	00	12	00	00	00	64	.....2t .7.....d
0020h:	00	00	00	02	E7	32	A3	98	B4	73	74	42	1D	33	12	00	....ç2£˜´stB.3..
0030h:	20	00	00	00	65	6E	74	72	6F	70	79	5F	73	61	6D	70	...entropy_samp
0040h:	6C	65	2E	74	78	74	00	F0	79	36	76	0C	8C	FF	0C	B5	le.txt.ðy6v.Œÿ.µ
0050h:	7F	BA	6C	95	23	BF	06	00	85	3F	0F	5A	F5	C4	3D	7B	.°l•#¿...…?.ZõÄ={
0060h:	00	40	07	00													.@..

Entropy = 5.0592468625650353

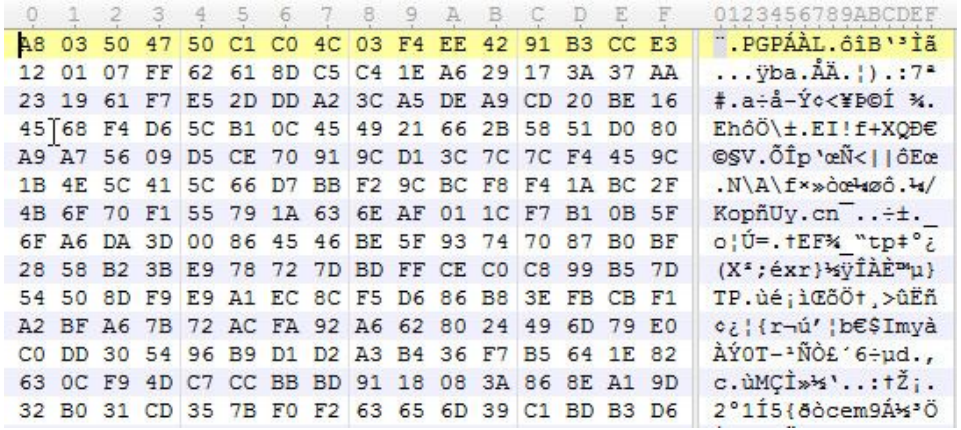
Case 4 - encrypt case 2 with PGP



0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
A8	03	50	47	50	C1	C0	4C	03	F4	EE	42	91	B3	CC	E3	¨.PGPÁÀL.ôîB'³Ìã
12	01	07	FF	62	61	8D	C5	C4	1E	A6	29	17	3A	37	AA	...ÿba.ÅÄ.¦).:7ª
23	19	61	F7	E5	2D	DD	A2	3C	A5	DE	A9	CD	20	BE	16	#.a÷å-Ý¢<¥Þ©Í ¾.
45	68	F4	D6	5C	B1	0C	45	49	21	66	2B	58	51	D0	80	EhôÖ\±.EI!f+XQÐ€
A9	A7	56	09	D5	CE	70	91	9C	D1	3C	7C	7C	F4	45	9C	©§V.ÕÎp'œÑ<||ôEœ
1B	4E	5C	41	5C	66	D7	BB	F2	9C	BC	F8	F4	1A	BC	2F	.N\A\f×»òœ¼øô.¼/
4B	6F	70	F1	55	79	1A	63	6E	AF	01	1C	F7	B1	0B	5F	KopñUy.cn¯..÷±._
6F	A6	DA	3D	00	86	45	46	BE	5F	93	74	70	87	B0	BF	o¦Ú=.†EF¾_"tp‡°¿
28	58	B2	3B	E9	78	72	7D	BD	FF	CE	C0	C8	99	B5	7D	(X²;éxr}½ÿÎÀÈ™µ}
54	50	8D	F9	E9	A1	EC	8C	F5	D6	86	B8	3E	FB	CB	F1	TP.ùé¡ìŒõÖ†¸>ûËñ
A2	BF	A6	7B	72	AC	FA	92	A6	62	80	24	49	6D	79	E0	¢¿¦{r¬ú'¦b€$Imyà
C0	DD	30	54	96	B9	D1	D2	A3	B4	36	F7	B5	64	1E	82	ÀÝ0T–¹ÑÒ£´6÷µd.,
63	0C	F9	4D	C7	CC	BB	BD	91	18	08	3A	86	8E	A1	9D	c.ùMÇÌ»½'..:†Ž¡.
32	B0	31	CD	35	7B	F0	F2	63	65	6D	39	C1	BD	B3	D6	2°1Í5{ðòcem9Á½³Ö

Entropy = 7.8347915272089166

Case 4 - encrypt case 2 with PGP



0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
A8	03	50	47	50	C1	C0	4C	03	F4	EE	42	91	B3	CC	E3	¨.PGPÁÀL.ôîB'³Ìã
12	01	07	FF	62	61	8D	C5	C4	1E	A6	29	17	3A	37	AA	...ÿba.ÅÄ.¦).:7ª
23	19	61	F7	E5	2D	DD	A2	3C	A5	DE	A9	CD	20	BE	16	#.a÷å-Ý¢<¥Þ©Í ¾.
45	68	F4	D6	5C	B1	0C	45	49	21	66	2B	58	51	D0	80	EhôÖ\±.EI!f+XQÐ€
A9	A7	56	09	D5	CE	70	91	9C	D1	3C	7C	7C	F4	45	9C	©§V.ÕÎp'œÑ<||ôEœ
1B	4E	5C	41	5C	66	D7	BB	F2	9C	BC	F8	F4	1A	BC	2F	.N\A\f×»òœ¼øô.¼/
4B	6F	70	F1	55	79	1A	63	6E	AF	01	1C	F7	B1	0B	5F	KopñUy.cn¯..÷±._
6F	A6	DA	3D	00	86	45	46	BE	5F	93	74	70	87	B0	BF	o¦Ú=.†EF¾_"tp‡°¿
28	58	B2	3B	E9	78	72	7D	BD	FF	CE	C0	C8	99	B5	7D	(X²;éxr}½ÿÎÀÈ™µ}
54	50	8D	F9	E9	A1	EC	8C	F5	D6	86	B8	3E	FB	CB	F1	TP.ùé¡ìŒõÖ†¸>ûËñ
A2	BF	A6	7B	72	AC	FA	92	A6	62	80	24	49	6D	79	E0	¢¿¦{r¬ú'¦b€$Imyà
C0	DD	30	54	96	B9	D1	D2	A3	B4	36	F7	B5	64	1E	82	ÀÝ0T–¹ÑÒ£´6÷µd.,
63	0C	F9	4D	C7	CC	BB	BD	91	18	08	3A	86	8E	A1	9D	c.ùMÇÌ»½'..:†Ž¡.
32	B0	31	CD	35	7B	F0	F2	63	65	6D	39	C1	BD	B3	D6	2°1Í5{ðòcem9Á½³Ö

Entropy = 7.8347915272089166

How can I get the data ?

- Kaggle
- Virus Share
- Dasmalwerk
- Virus Total
- VX Heaven

Dataset

Clean

Windows 2008, XP, 7 and 10

41323

Malicious

Virus Share and Dasmalwerk

97812

Let’s play with some data

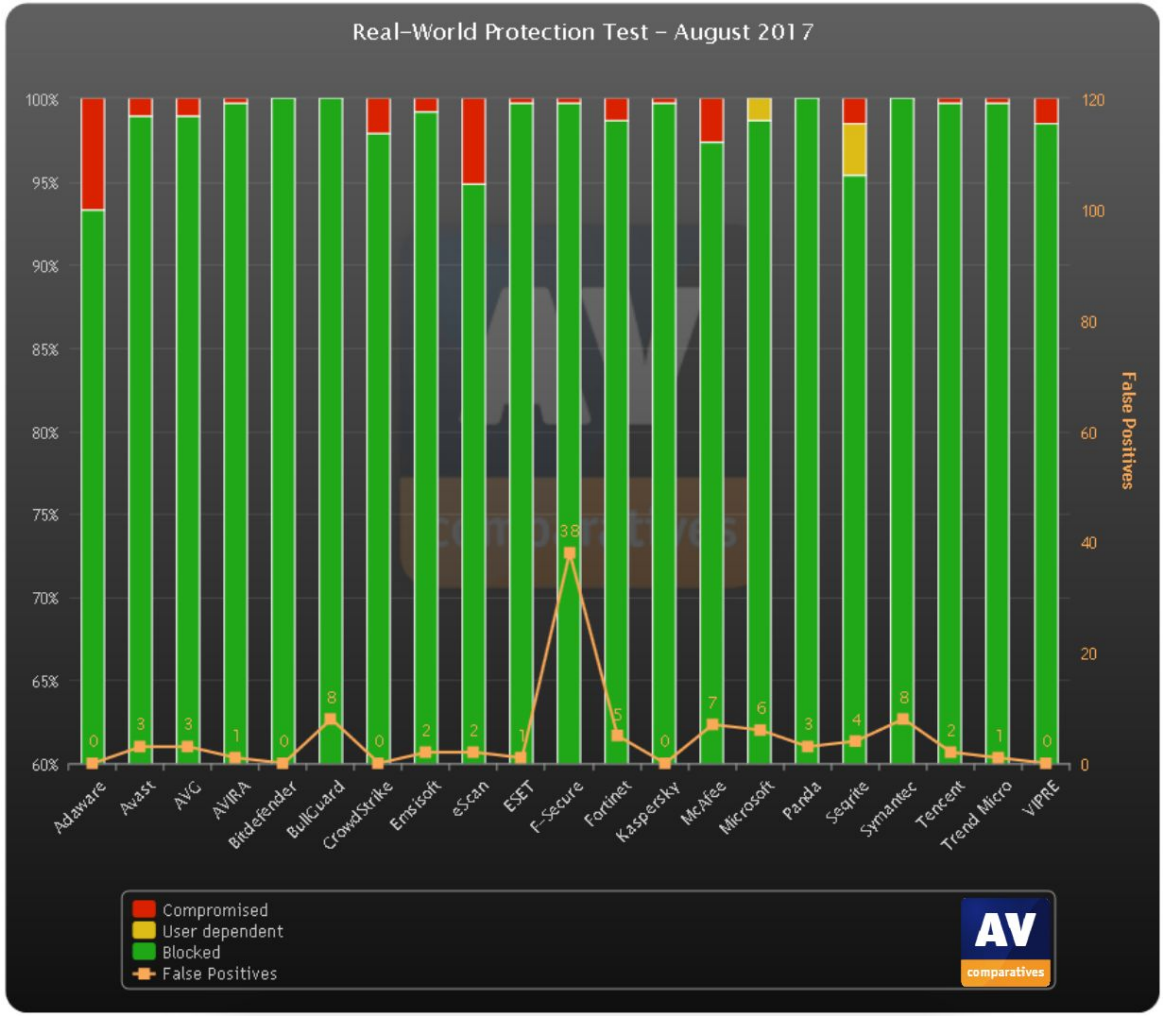
Trained models

Model	Accuracy
RandomForest	**99.416398 %**
MLP	30.005175 %
GradientBoosting	98.827047 %
GNB	70.322562 %
LinearRegression	58.461576 %
DecisionTree	99.011040 %
AdaBoost	98.556808 %

Training: **75%**

Testing: **25%**

Accuracy = (TP + TN) / Total


Real-World Protection Test – August 2017
100%
95%
90%
85%
80%
75%
70%
65%
60%
120
100
80
60
40
20
0
False Positives
0
3
3
1
0
8
0
2
2
1
38
5
0
7
6
3
4
8
2
1
0
Adaware
Avast
AVG
AVIRA
Bitdefender
BullGuard
CrowdStrike
Emsisoft
eScan
ESET
F-Secure
Fortinet
Kaspersky
McAfee
Microsoft
Panda
Seqrite
Symantec
Tencent
Trend Micro
VIPRE
Compromised
User dependent
Blocked
False Positives
AV
comparatives

Assumptions

- Variations and new malware can be detected since we look for general patterns.
- Packed/obfuscated malware can be detected since we use entropy.

Conclusions

- Signature matching is too limited, behavioral analysis ftw.
- We can apply ML to problems which we are not experts.
- ML is easy with python and its libraries.
- The results are good but building an commercial grade AV is another story.

Questions

Why relying on antivirus signatures is simply not enough anymore
https://www.webroot.com/blog/2012/02/23/why-relying-on-antivirus-signatures-is-simply-not-enough-anymore/

Computer virus: What are they and how to avoid them
https://www.eecis.udel.edu/~portnoi/publications/pcvirus-eng.html

Making an antivirus engine : the guidelines
https://www.adlice.com/making-an-antivirus-engine-the-guidelines/

ClamAV
https://www.clamav.net/

The PE file format
http://www.pelib.com/resources/luevel.txt

Malware Sample Sources for Researchers
https://zeltser.com/malware-sample-sources/

Utilizing entropy to identify undetected malware
https://www.guidancesoftware.com/docs/default-source/document-library/whitepaper/utilizing-entropy-to-identify-undetected-malware.pdf?sfvrsn=16

Shannon Entropy
http://www.bearcave.com/misl/misl_tech/wavelets/compression/shannon.html