Università di Genova

Pattern Recognition and Applications Lab

Università di Cagliari

Adversarial EXEmples: Evading Windows Malware Detectors

Luca Demetrio
luca.demetrio@unige.it
@zangobot

Machine Learning applied as Security Scanner

Spreading into commercial products
Companies claim to use machine learning technologies inside their detectors to spot Windows malware by learning patterns from data

Filter out known threats, generalize to variants
Deep networks learn “signatures”, and they can spot variants of the same malware

Machine Learning is vulnerable
They all might be target of evasion attacks!
<u>But how?</u>

SOPHOS
Security made simple.

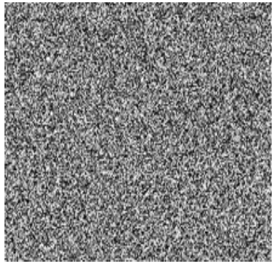
Intercept X Deep Learning

CrowdStrike Introduces Enhanced Endpoint Machine Learning Capabilities and Advanced Endpoint Protection Modules

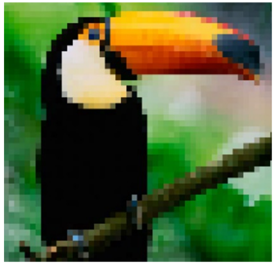
— Company continues to accelerate pace of replacement of legacy AV solutions in both enterprise and SMB markets —

Images and programs are different

toucan (97%)

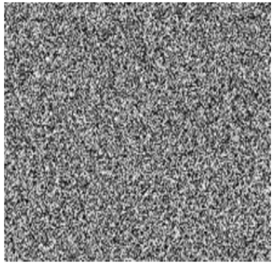


adversarial noise

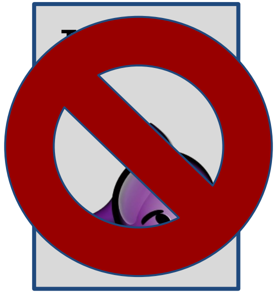


cat (95%)

malware (98%)



adversarial noise



Not runnable anymore!

Byte-sequences are not numbers

Programs and images are encoded in bytes

RGB is "continuous", code instructions are not!

Distance between programs is **undefined**

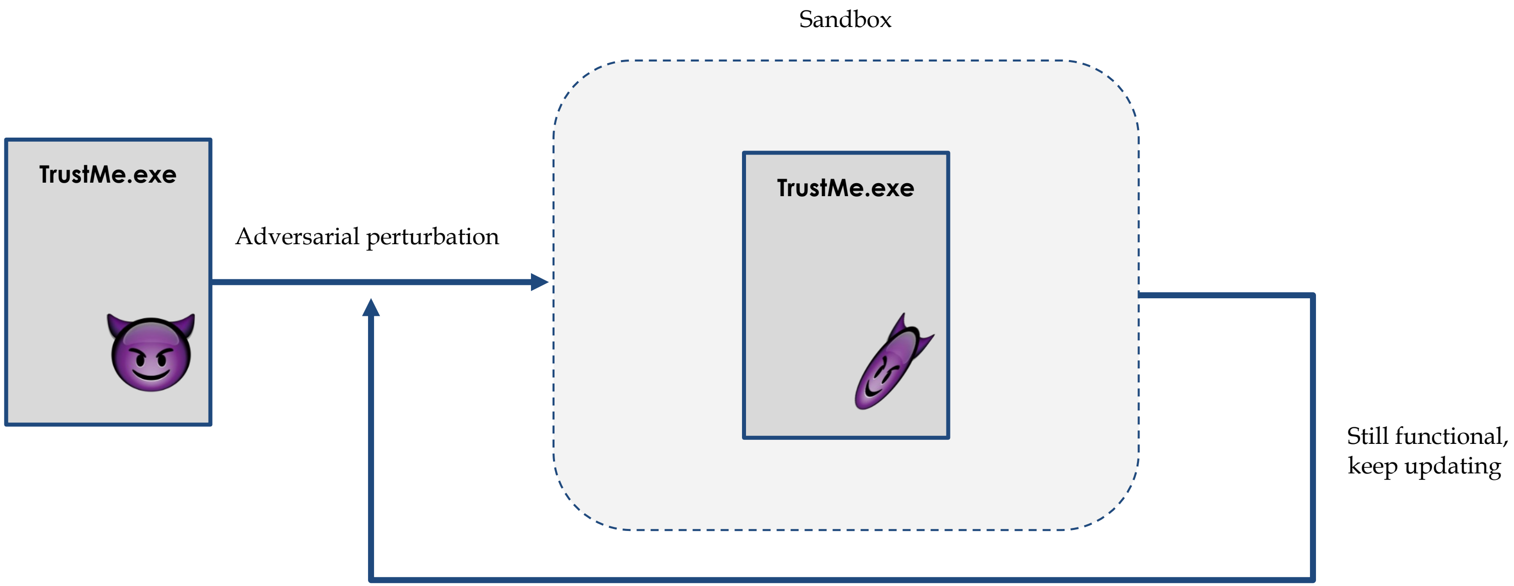
Example: ASCII table

What does $\sqrt{'a' - 'b'}$ means?

Dec	Hx	Oct	Char		Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL	(null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH	(start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX	(start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX	(end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT	(end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ	(enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK	(acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL	(bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS	(backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB	(horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF	(NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT	(vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF	(NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR	(carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO	(shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI	(shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE	(data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1	(device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2	(device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3	(device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4	(device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK	(negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN	(synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB	(end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN	(cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM	(end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB	(substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC	(escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS	(file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS	(group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS	(record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US	(unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.asciitable.com

Preserve the original functionality



Song et al., *MAB-Malware: A Reinforcement Learning Framework for Attacking Static Malware Classifiers*, arXiv preprint 2021
Castro et al., *AIMED: Evolving Malware with Genetic Programming to Evade Detection*, TrustCom/BigDataSE 2019

How to bridge these gaps?

1. Formulate the minimization problem differently
2. Study the format that represent programs
3. Understand how to exploit the format
4. Chose how to inject or perturb the content

Formulation of the problem

Adversarial attacks for images

$$\min_{\boldsymbol{\delta}} \boxed{L}\left(\boxed{\boldsymbol{x} + \boldsymbol{\delta}}, y; \boldsymbol{\theta}\right)$$

Network architecture in the loss
All the internals of a neural network / shallow model are hidden inside the loss

Additive Manipulation
Input samples are injected with additive noise, without any concern on the structure of the file

Adversarial attacks for security detectors

$$\min_{\delta} L(f(\phi(h(x;\delta)), y))$$

Model function and features
Need to explicit the model function and the features, since they might be non differentiable

Practical Manipulations
No additions, but a complex function that handles format specification by design

Demetrio et al., *Adversarial EXEmples: a Survey and Experimental Evaluation of Practical Attacks on Machine Learning for Windows Malware Detection*, ACM TOPS 2021

Take-home message: implementing an attack

$$\boxed{\min_{\boldsymbol{\delta}}} L(f(\phi(\boxed{h(x;\delta)}), y))$$

Define the Optimizer
Depending on the differentiability of the compontens, pick a gradient-based or gradient-free algorithm

Define the Manipulations
Study the format, understand its ambiguities, and write manipulations that do not break the original functionality

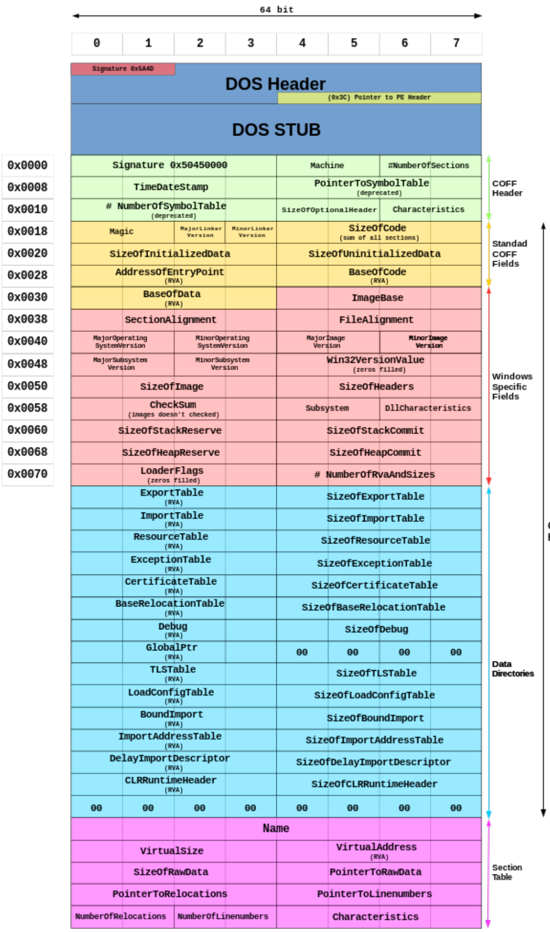
Demetrio et al., *Adversarial EXEmples: a Survey and Experimental Evaluation of Practical Attacks on Machine Learning for Windows Malware Detection*, ACM TOPS 2021

Windows PE File Format

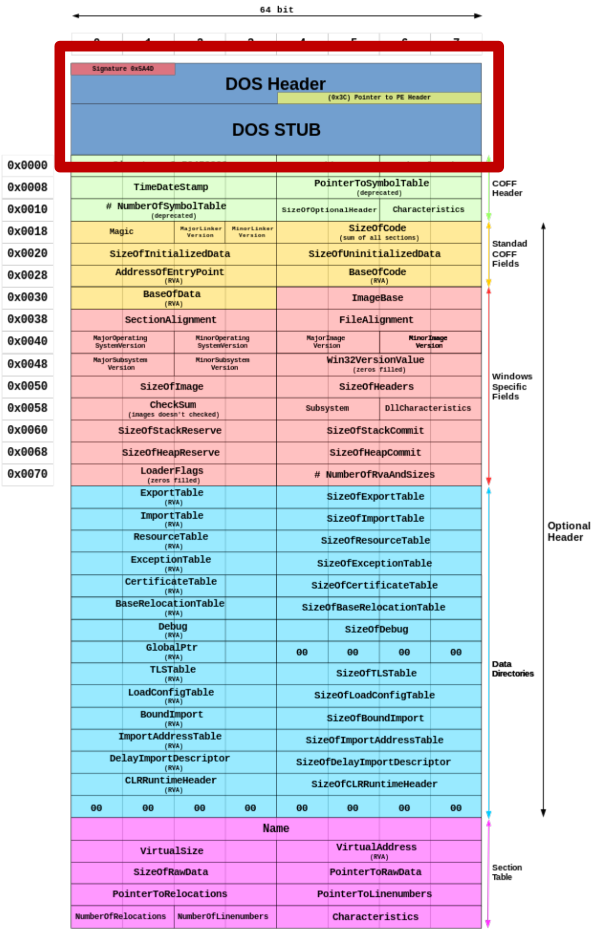
Windows PE File Format

Format adapted for "modern" programs (from Windows NT 3.1 on)

Before there were other formats, one is the DOS (kept for retrocompatibility)



Windows PE File Format



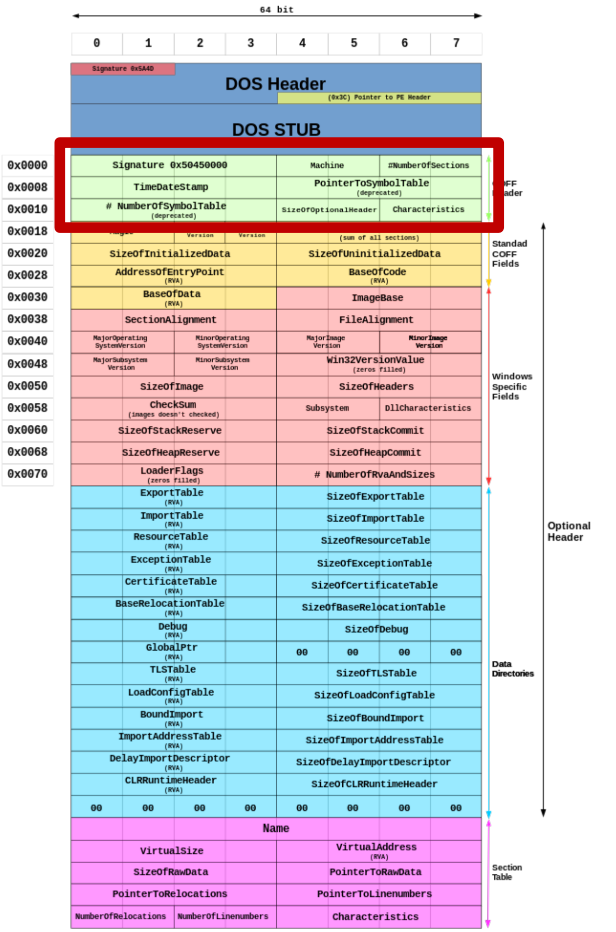
DOS Header + Stub

Metadata for DOS program

Executing a modern program in DOS will trigger the "This program cannot be run in DOS mode" output

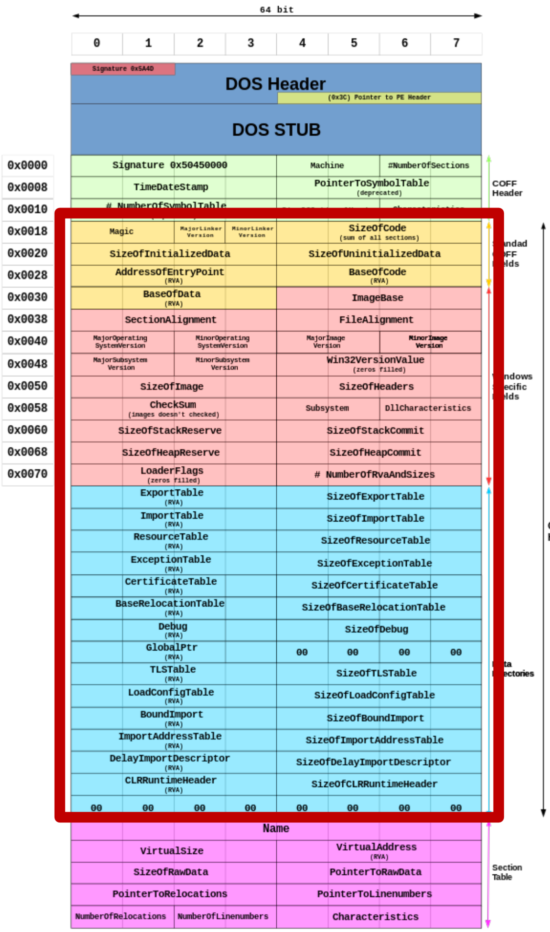
Windows PE File Format

PE Header
Real metadata of the program
Dscribes general information of the file



Windows PE File Format

Optional Header
Spoiler: not optional at all :)
Instructs the loader where to find each object inside the file



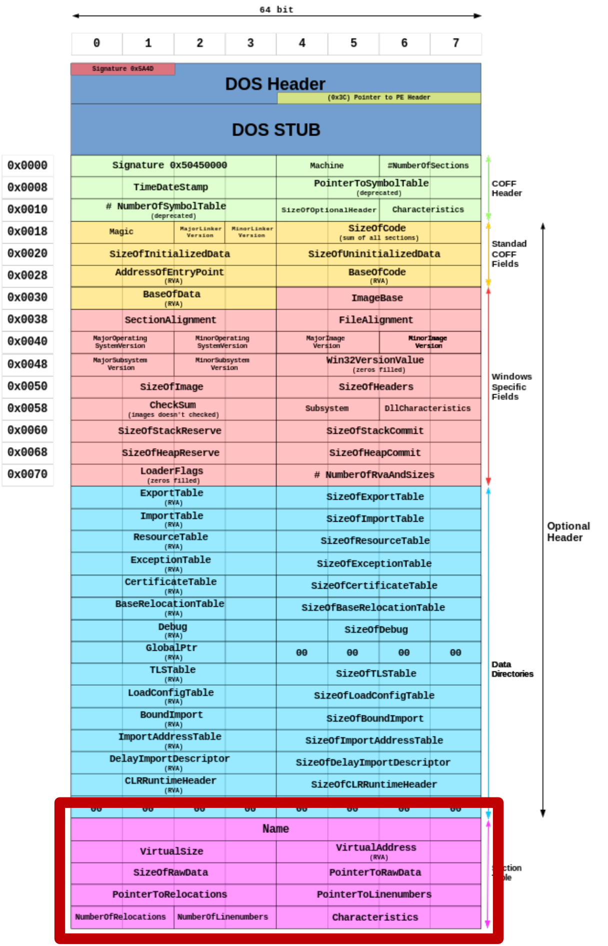
Windows PE File Format

Section Table and Sections

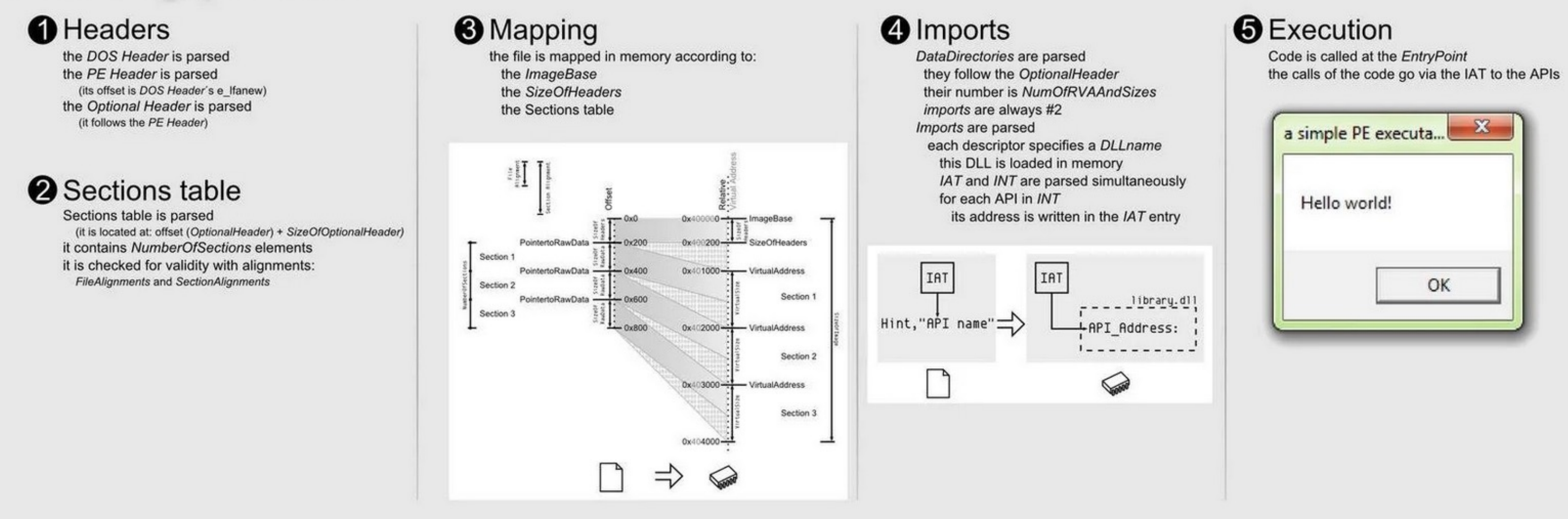
Describes where to find code, initialized data, resources, etc to the loader

These are “sections”, and each has a “section entry” with its characteristics

Examples: code is “.text”, read-only data is “.rodata”, resources are “.rsc”, and counting



How programs are loaded



https://code.google.com/archive/p/corkami/wikis/PE101.wiki

(Thanks Ange Albertini)

Practical Manipulations

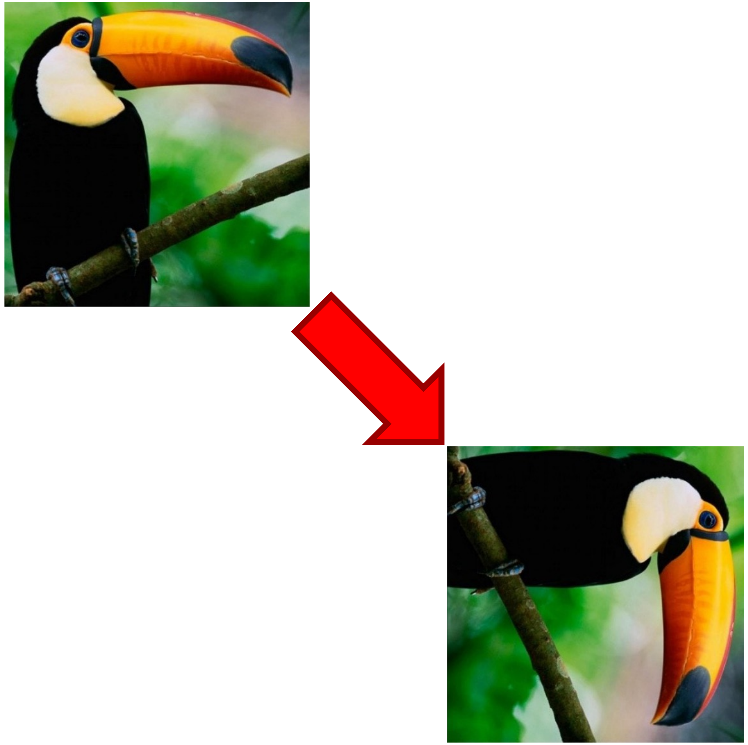
Practical Manipulations

Perturb the representation of a file

Keep intact the original functionality

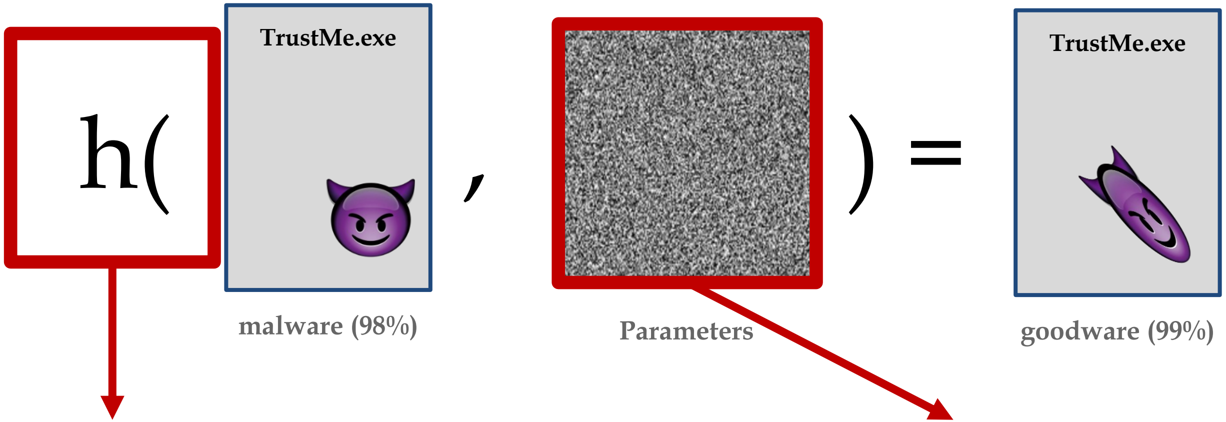
Example: rotation for images

How to bridge the gap?



Demetrio et al., *Adversarial EXEmples: a Survey and Experimental Evaluation of Practical Attacks on Machine Learning for Windows Malware Detection*, ACM TOPS 2021

How to manipulate programs

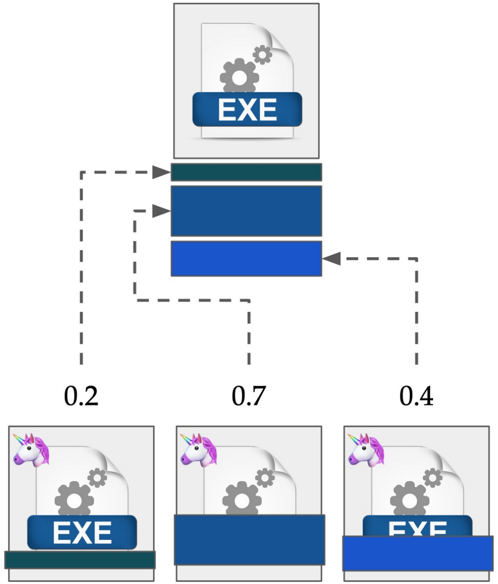


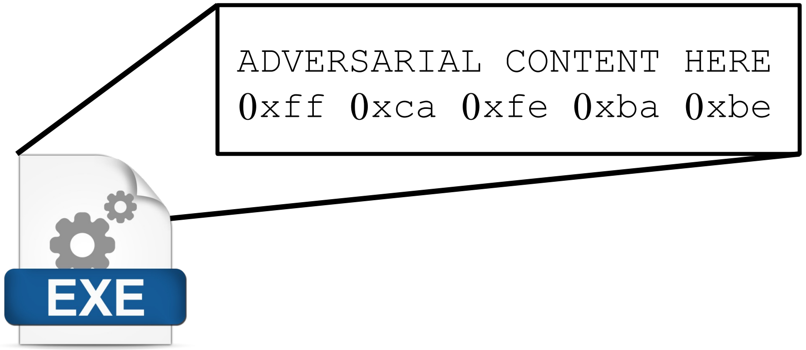
Practical Manipulation function
Alter file representation without destroying the structure and the functionalities and avoid usage of sandboxes

Parameters
Manipulations are parametrized so an optimization algorithm can fine tune them

Demetrio et al., *Adversarial EXEmples: a Survey and Experimental Evaluation of Practical Attacks on Machine Learning for Windows Malware Detection*, ACM TOPS 2021

Structural Manipulations





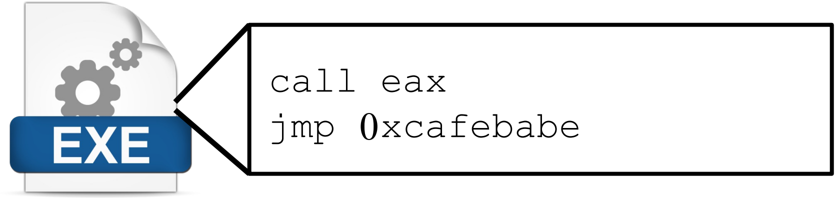
Injecting content
Alter file structure to include more byte sequences

Replacing content
Leverage ambiguous file format specifications to alter bytes that are not considered at runtime

Demetrio et al., *Functionality-preserving Black-box Optimization of Adversarial Windows malware*, IEEE TIFS 2021

Behavioral Manipulations

WARNING
more difficult to implement!



Packing and obfuscation
Encrypt program inside another one, or complicate the sequence of instructions

Inject new execution flows
Call APIs, add loops, jump to new code sections, and more

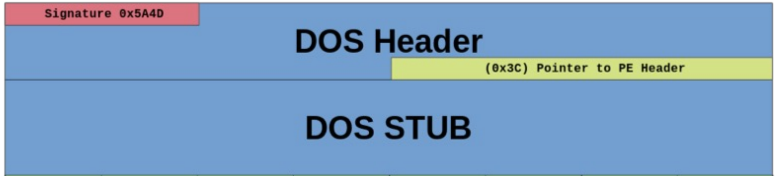
Demetrio et al., *Functionality-preserving Black-box Optimization of Adversarial Windows malware*, IEEE TIFS 2021

DOS header perturbations

The attacker edit as much bytes as they want

Untouched: magic number MZ and offset to real PE header

Content loaded in memory, not executed



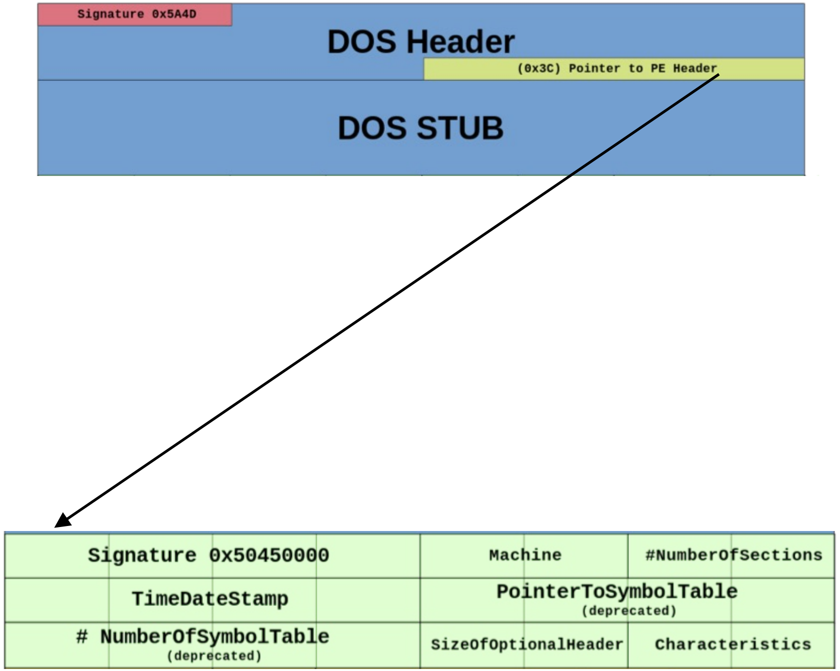
Demetrio et al., *Adversarial EXEmples: a Survey and Experimental Evaluation of Practical Attacks on Machine Learning for Windows Malware Detection*, ACM TOPS 2021

DOS header extension

Exploit offset to real header, increment value

Insert arbitrary content between DOS header and PE header

Content loaded into memory, not executed



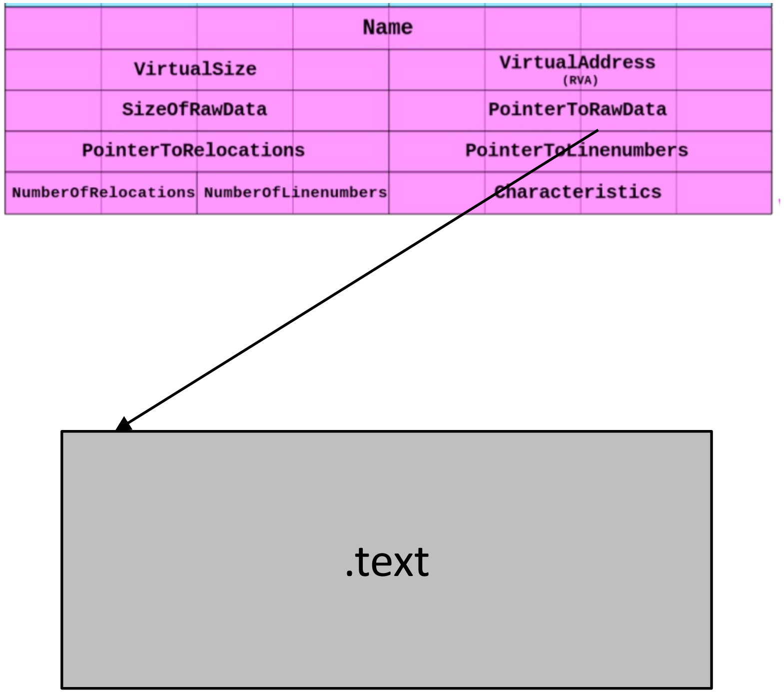
Demetrio et al., *Adversarial EXEmples: a Survey and Experimental Evaluation of Practical Attacks on Machine Learning for Windows Malware Detection*, ACM TOPS 2021

Content shifting

Exploit offset in section entry, increment to manipulate the loader in searching for section content

The attacker can inject content after the section table, or between sections

NOT LOADED IN MEMORY, skipped by the loader



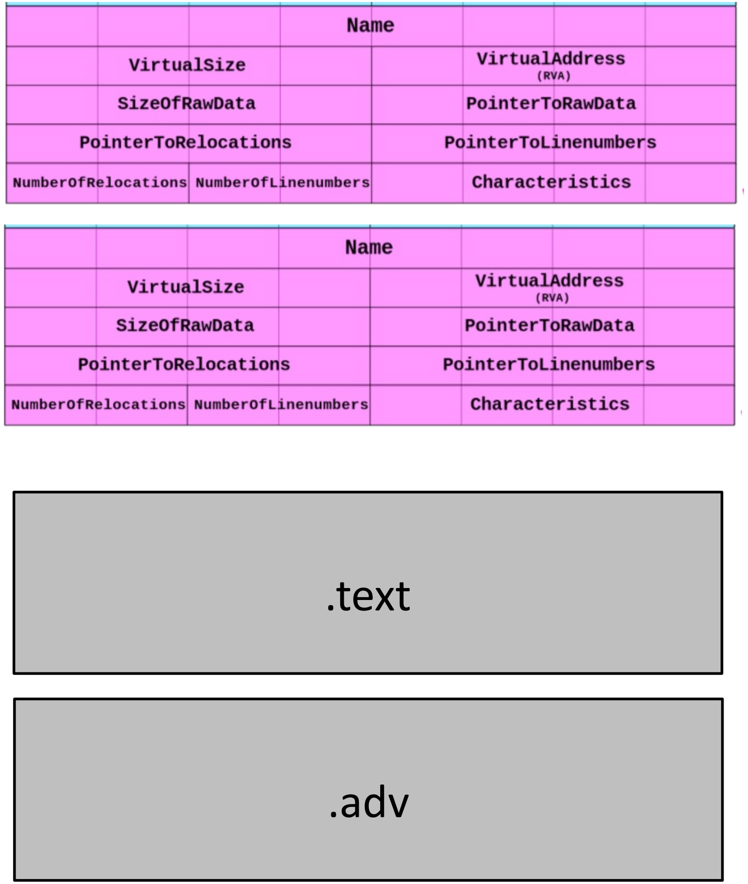
Demetrio et al., *Adversarial EXEmples: a Survey and Experimental Evaluation of Practical Attacks on Machine Learning for Windows Malware Detection*, ACM TOPS 2021

Section Injection

Manipulate section table to add new entry

Append chunk of bytes, referenced by newly added entry

Loaded in memory or not, depending by the characteristics set up inside the entry



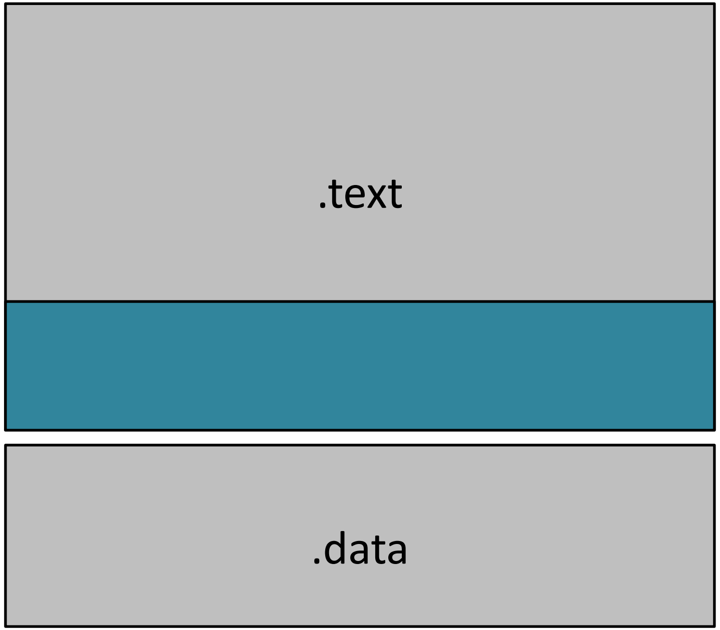
Demetrio et al., *Adversarial EXEmples: a Survey and Experimental Evaluation of Practical Attacks on Machine Learning for Windows Malware Detection*, ACM TOPS 2021

Slack space

Section content is padded with 0 to keep file alignments

The attacker can rewrite such slack space

Loaded in memory, not executed



Demetrio et al., *Adversarial EXEmples: a Survey and Experimental Evaluation of Practical Attacks on Machine Learning for Windows Malware Detection*, ACM TOPS 2021

Padding

Appending content at the end

Most trivial manipulation

Not loaded in memory

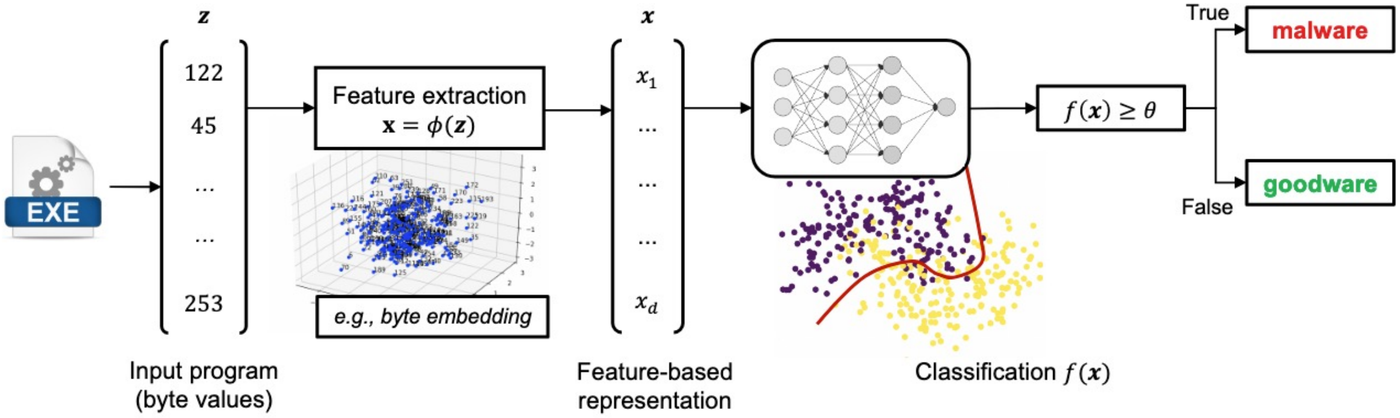
Demetrio et al., *Adversarial EXEmples: a Survey and Experimental Evaluation of Practical Attacks on Machine Learning for Windows Malware Detection*, ACM TOPS 2021

Optimization Algorithms

Chosing the strategy accordingly

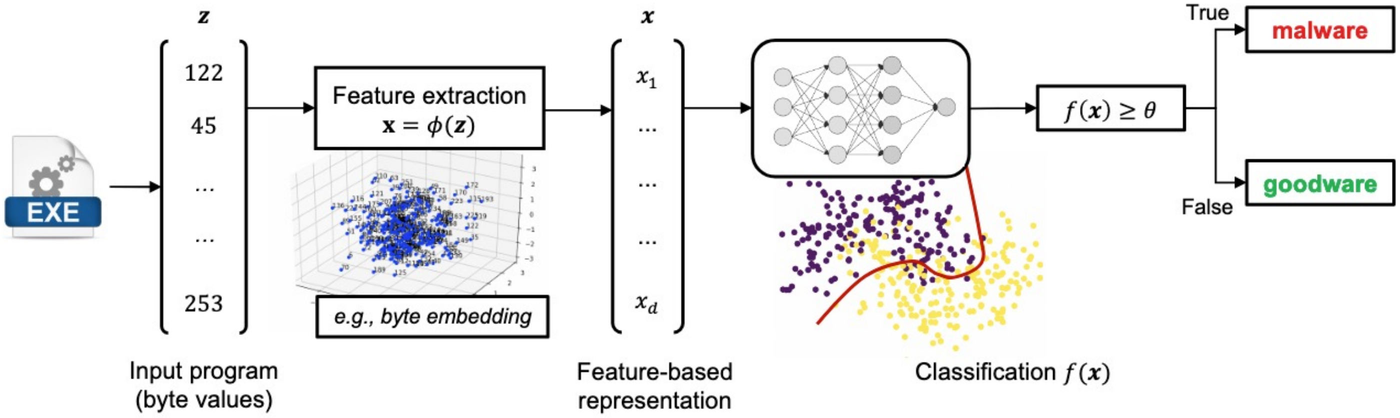
Gradient-based	**Gradient-free**
Own the model AND Model is differentiable	Model not accessible OR Model is not differentiable

Gradient-based strategies



Use gradient-descent to compute adversarial examples (as for images)

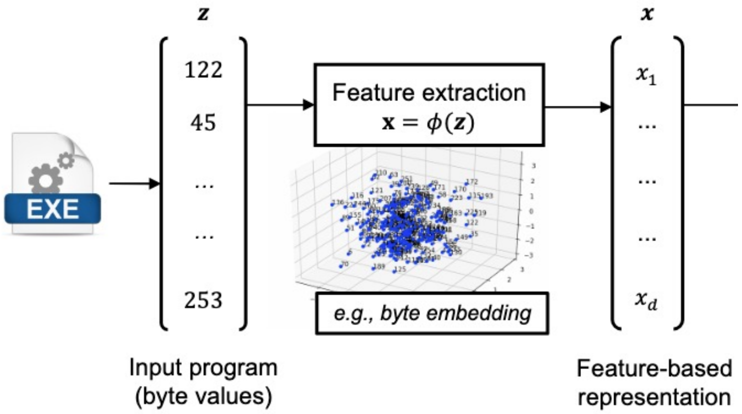
Gradient-based strategies



~~Use gradient-descent to compute adversarial examples (as for images)~~

Bytes do not have a distance metric, a feature extractor is **ALWAYS** needed to compute something meaningful

Embedding for end-to-end networks



All bytes are replaced with a vector learned at training time, where a distance metric is imposed…

… but the embedding layer is **not differentiable**

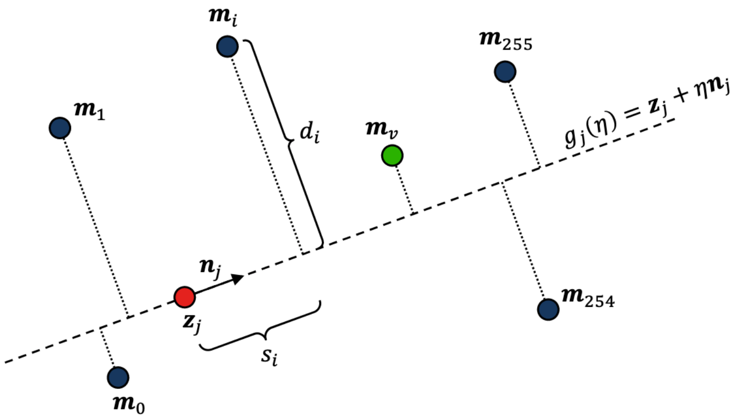
How to propagate gradient information?

$$\boxed{\frac{\partial L}{\partial \delta}} = \frac{\partial L}{\partial f}\frac{\partial f}{\partial \phi}\boxed{\frac{\partial \phi}{\partial h}\frac{\partial h}{\partial \delta}}$$

End-to-end gradient you would like to compute

Non-differentiable manipulations and embedding!

Solution: change the optimizer



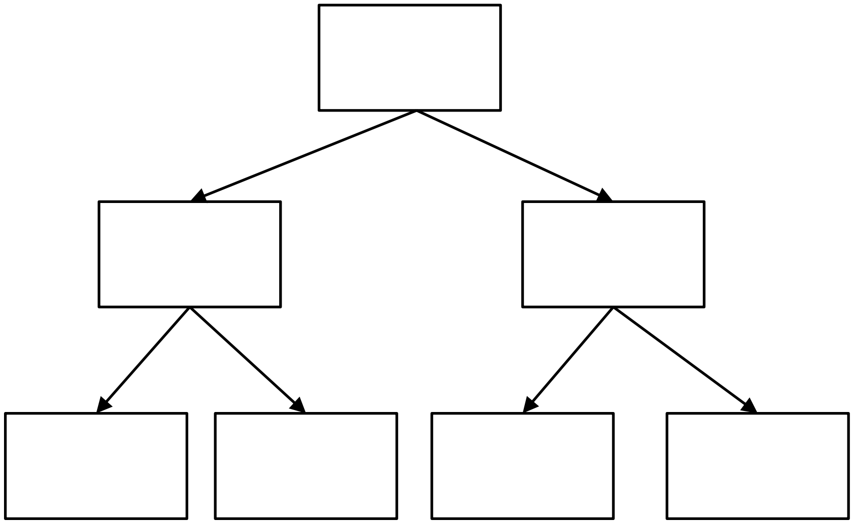
1. Compute gradient in feature space
2. Define a way for replacing values
   For bytes: inverse look-up of embedding
3. Follow the direction of gradient and replace byte with other byte

Kolosnjaji et al., *Adversarial malware binaries: Evading deep learning for malware detection in executables*, EUSIPICO 2018

Chosing the strategy accordingly

Gradient-based	**Gradient-free**
Own the model AND Model is differentiable	Model not accessible OR Model is not differentiable

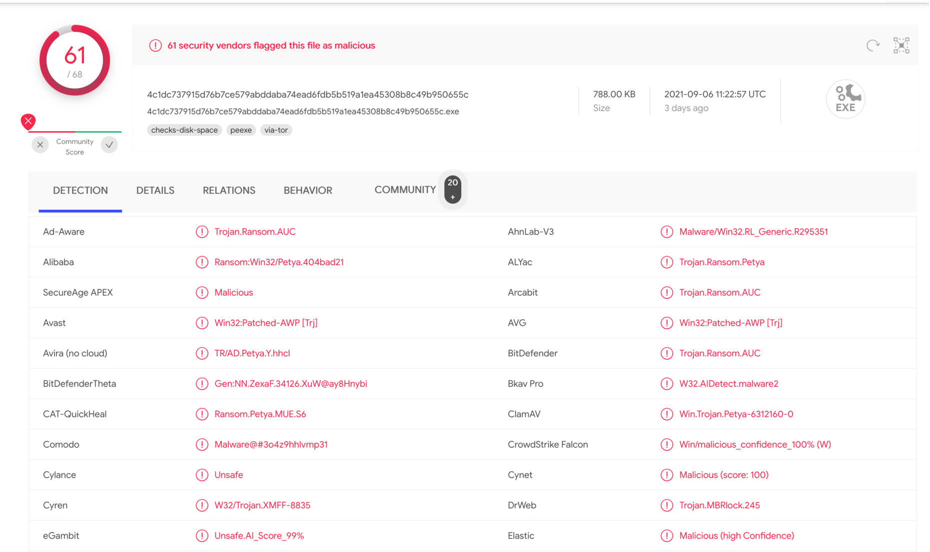
Reality check: robust models are not differentiable



State-of-the-art classifiers use decision trees

No gradients can be computed

Reality check: most models are unavailable

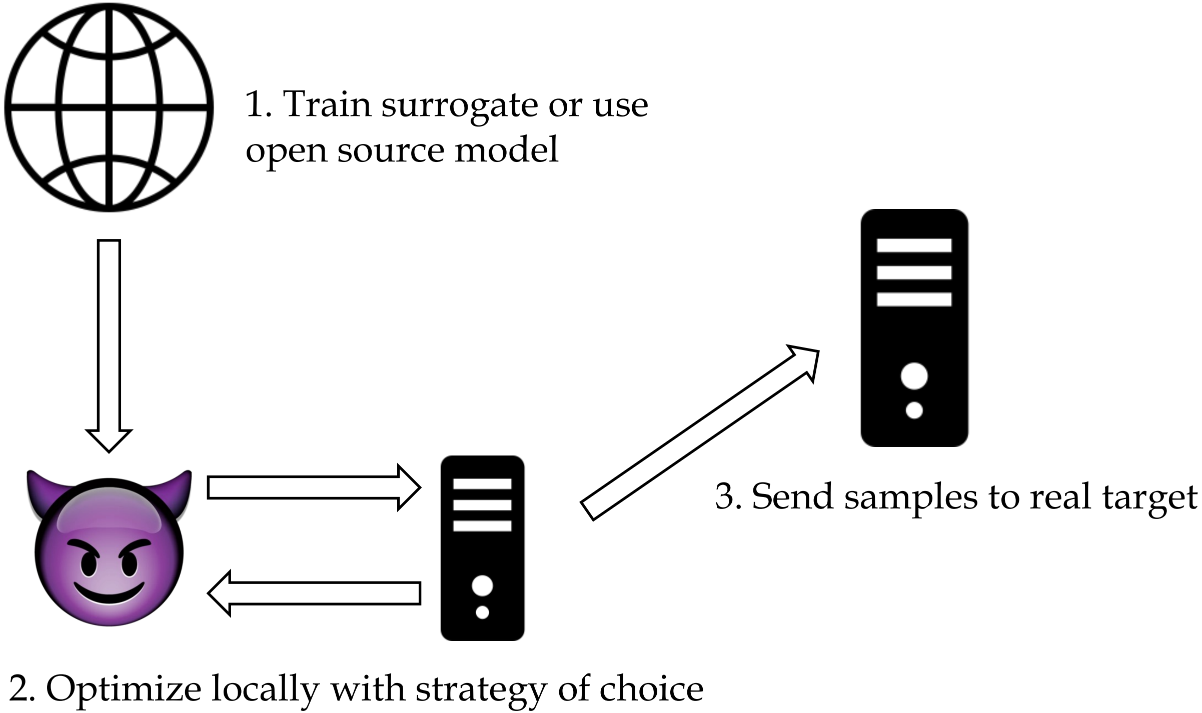


Most models are hosted on private servers

Detection performed in cloud

No gradients can be computed

Transfer attacks



Demetrio et al., *Functionality-preserving Black-box Optimization of Adversarial Windows malware*, IEEE TIFS 2021

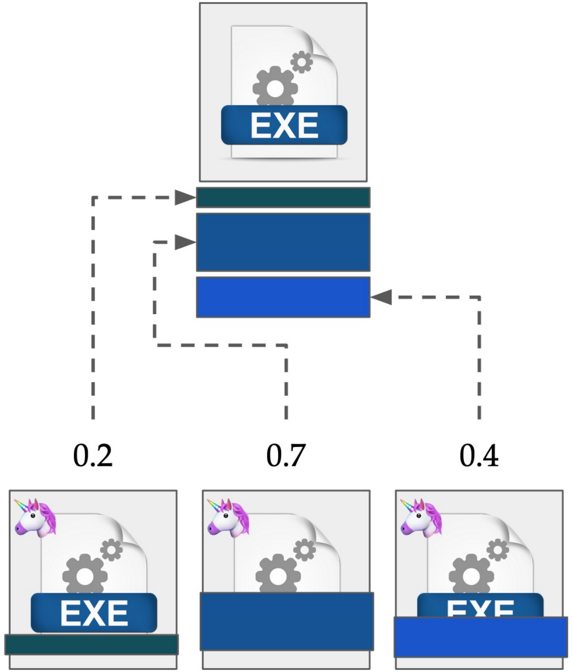
Query attacks

Very slow if optimizer works byte-per-byte

Demetrio et al., *Functionality-preserving Black-box Optimization of Adversarial Windows malware*, IEEE TIFS 2021

Speeding up by injecting benign content

Intuition
classifiers can be fooled by introducing content of the goodware class!



Demetrio et al., *Functionality-preserving Black-box Optimization of Adversarial Windows malware*, IEEE TIFS 2021

(In)Famous example: CyLance

Injecting bytes

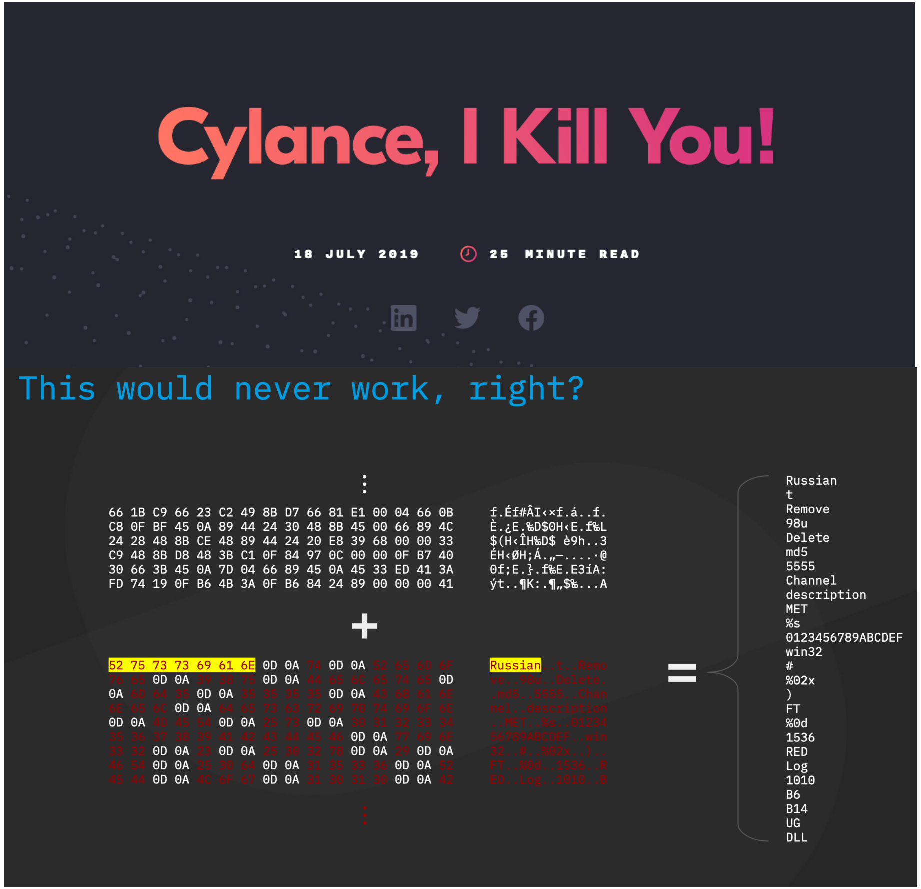
Reversing the code with some tricks, discovered that the model leverages STRINGS

Inject “benign” values

Extract byte sequences from “Rocket League” and include them inside input exacutable

Evasion completed!

The company rolled out an update to try to mitigate the issue

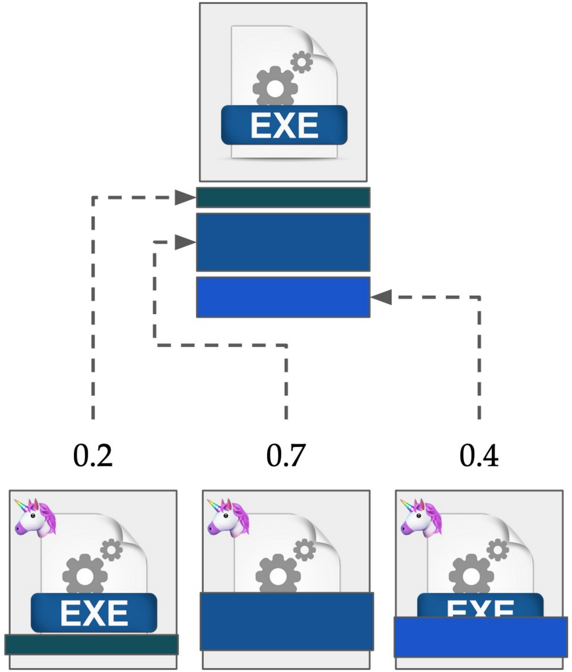


P.S. they did not, it is still vulnerable on VirusTotal, the write-up is 3 years old now

[Skylight Cyber. Cylance, I Kill You! https://skylightcyber.com/2019/07/18/cylance-i-kill-you/]

Benign content to fool the network

The optimizer explore less space, no modification byte-per-byte, but it relies on portions of goodware programs injected with practical manipulations



Demetrio et al., *Functionality-preserving Black-box Optimization of Adversarial Windows malware*, IEEE TIFS 2021

Laboratory time!

Thanks!

Luca Demetrio
luca.demetrio93@unica.it
@zangobot

If you know the enemy and know yourself, you need not fear the result of a hundred battles
Sun Tzu, The art of war, 500 BC