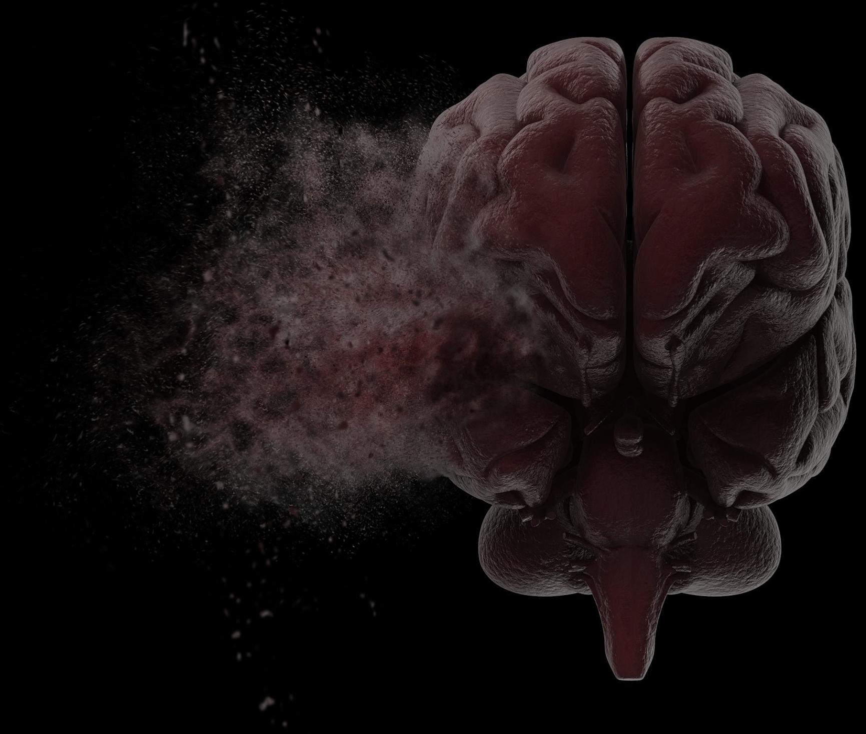
SKYLIGHT
ZERO COMPROMISE



Attacking Machine Learning
The Cylance Case Study

BSides Sydney 2019

About Us

Not even AI power users...

Adi Ashkenazy
CEO, Skylight Cyber

Wears T-Shirts in Corporate Headshots
[yes, I am wearing it now too!]

Heavy on the offensive cyber side (Government)

Red-team automation

Shahar Zini
CTO, Skylight Cyber

Category 5 stage fright

Sydney based consultancy

Help companies navigate cyber security

In a nutshell

Why is this important?

What are we looking to achieve?

AI in Cyber for people who understand quickly

How we approached the problem and reversing the product

Results!

Publication and Feedback

Questions

Challenge

Silver Bullet Hunting

Assess Technology

“The product is as close as you can get to a silver bullet in our space. Greater than 99% efficacy and protection against nearly every zero-day malware

WHY CYLANCE?

AI Centric, can buy it off the shelf, consistently ranks high

Zero-Day Attacks

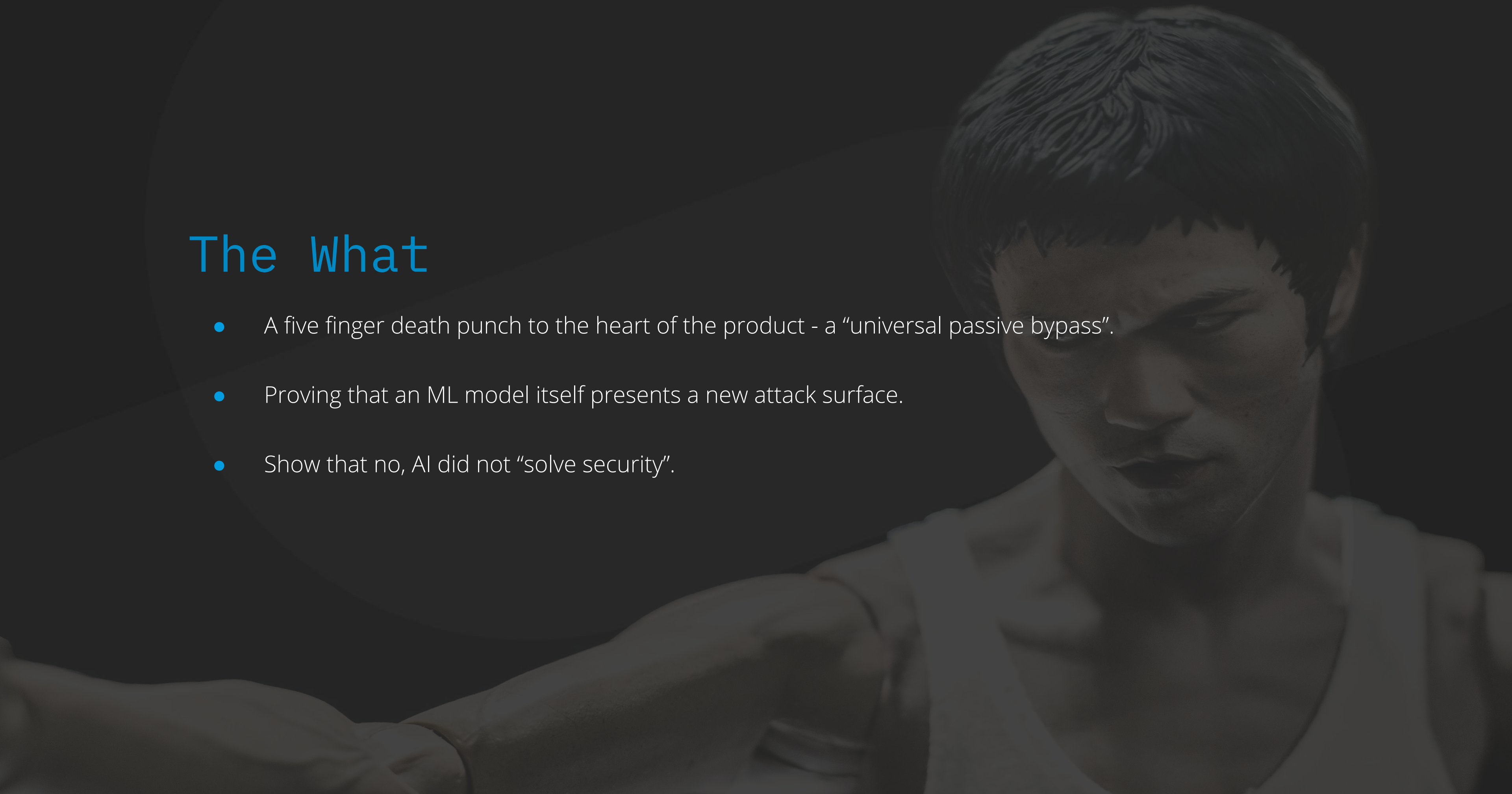
AI model prevents zero-day payloads from executing

Their marketing didn’t help!

Forrester Report: Cylance Provides 251% ROI

The What

- A five finger death punch to the heart of the product - a “universal passive bypass”.
- Proving that an ML model itself presents a new attack surface.
- Show that no, AI did not “solve security”.



Classification with AI/ML - The Basics

"In machine learning and statistics, classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known"

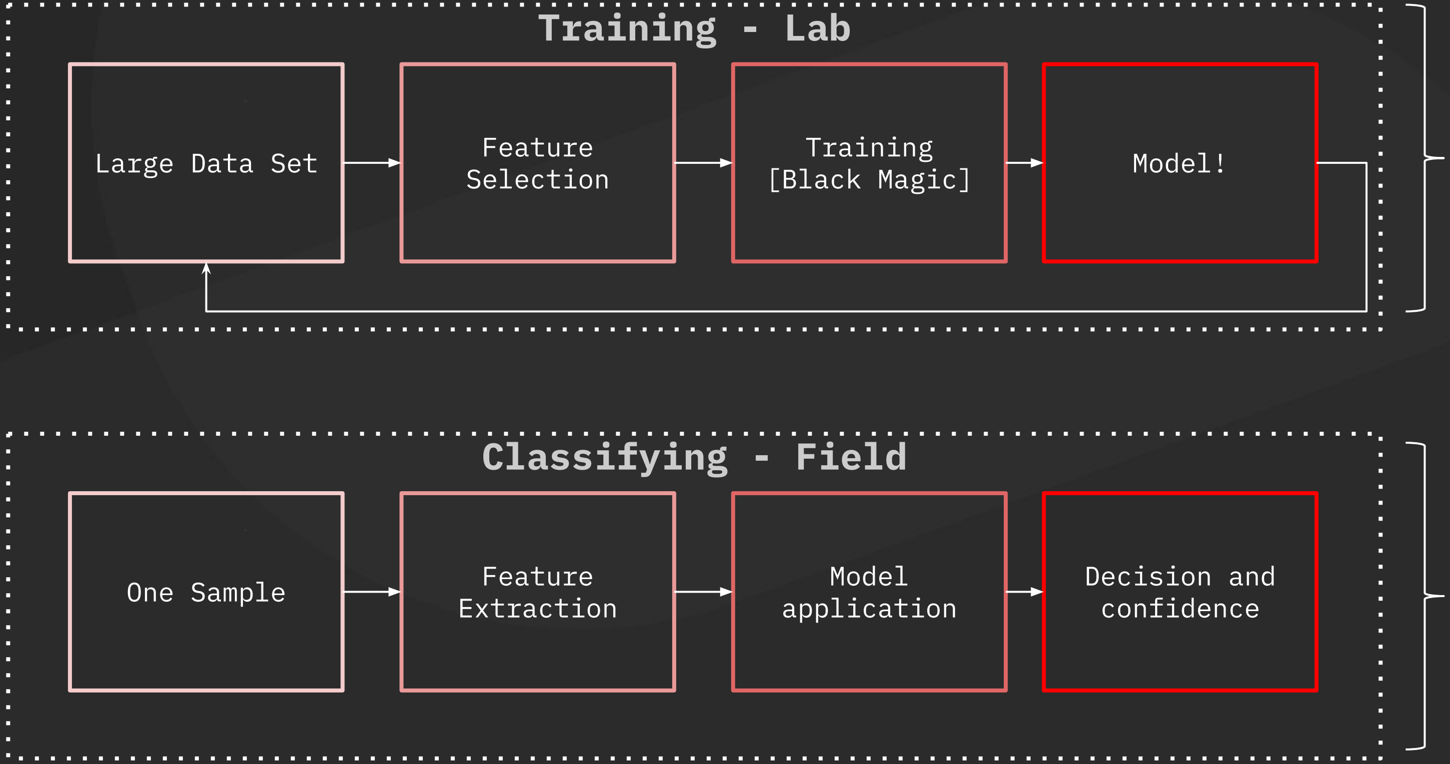
Wikipedia

What is THAT?

Cat

Dog

Classification with AI/ML



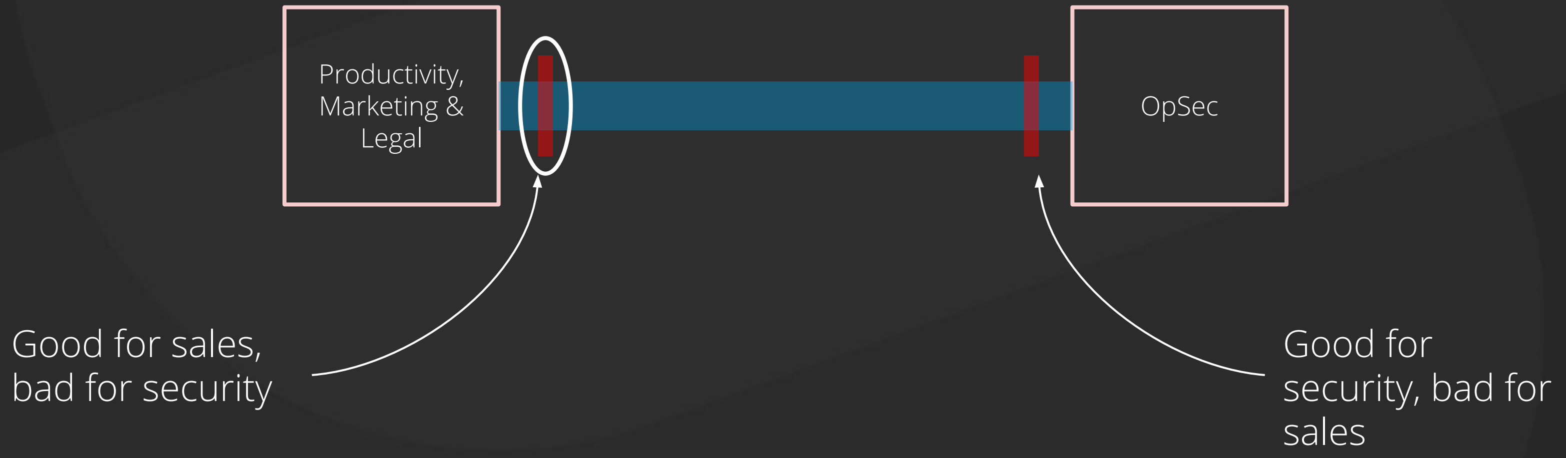
Lengthy, resource intensive

Quick and easy to replicate

An offensive mindset

- Classification is innately naive
- A model is only as good as its data
- How would we fool the bird vs. human classifier?

The OpSec Paradox



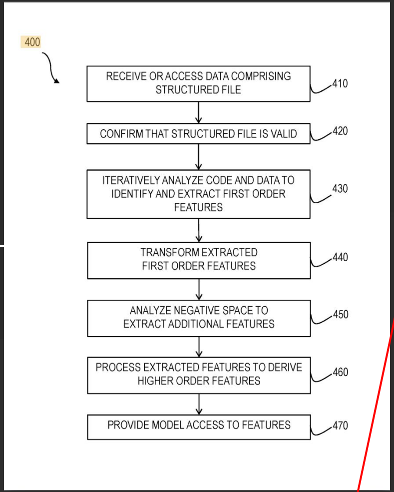
OSINT

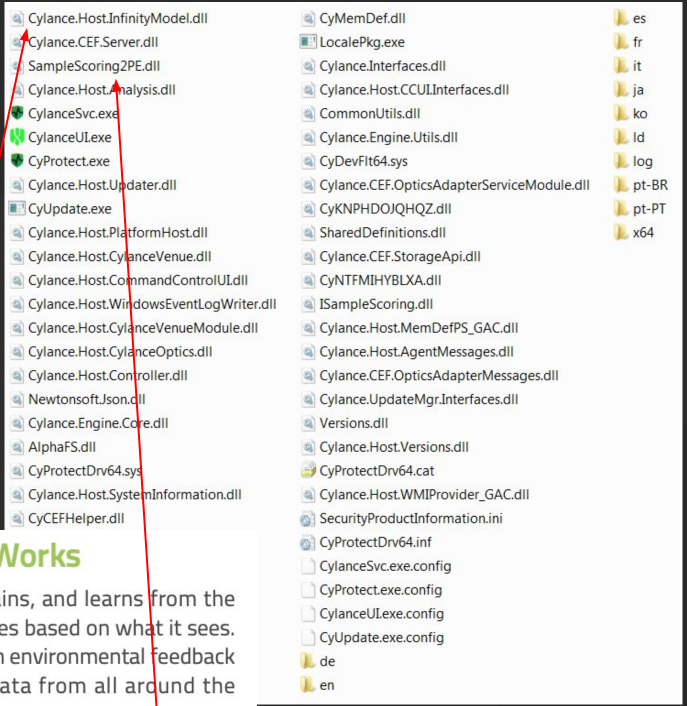
Marketing & Legal

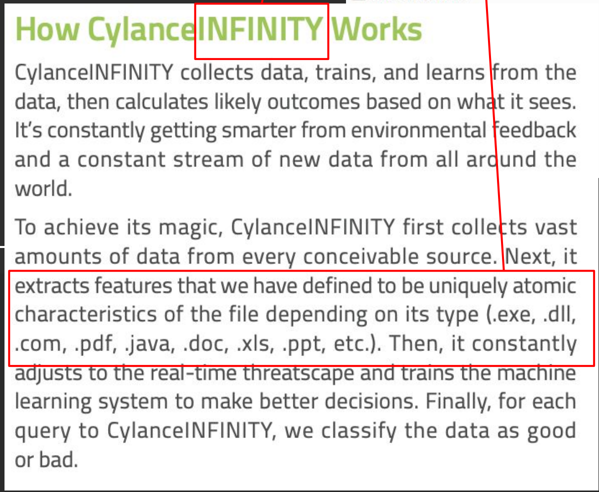
Patents

White Papers & Booklets

Conference Talks







How CylanceINFINITY Works

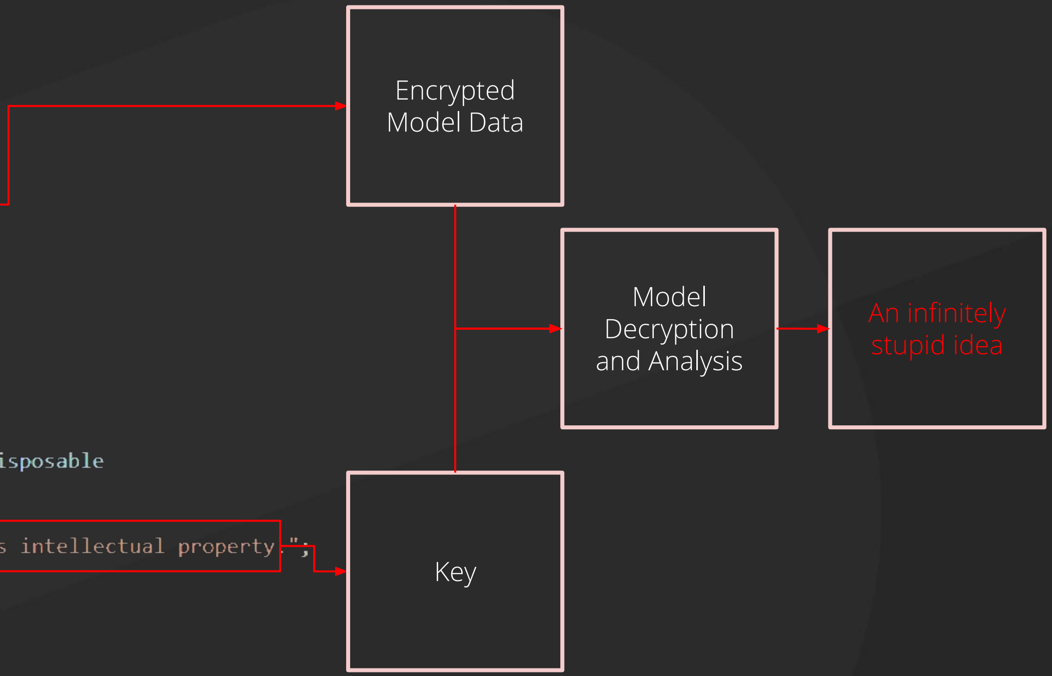
CylanceINFINITY collects data, trains, and learns from the data, then calculates likely outcomes based on what it sees. It's constantly getting smarter from environmental feedback and a constant stream of new data from all around the world.

To achieve its magic, CylanceINFINITY first collects vast amounts of data from every conceivable source. Next, it extracts features that we have defined to be uniquely atomic characteristics of the file depending on its type (.exe, .dll, .com, .pdf, .java, .doc, .xls, .ppt, etc.). Then, it constantly adjusts to the real-time threatscape and trains the machine learning system to make better decisions. Finally, for each query to CylanceINFINITY, we classify the data as good or bad.

Extracting the Model

```
◢ Cylance.Host.InfinityModel (2.0.1530.3, msil, .Net Framework v3.5)
   ▷ Metadata
   ▷ References
   ◢ Resources
        Cylance.Host.InfinityModel.model
        Cylance.Host.InfinityModel.padding
```

```
namespace Cylance.Engine.Core.Ensemble
{
  public class EnsembleReader : ILogAccess, IEnsembleHeader, IDisposable
  {
    protected const int RandomHeaderSize = 3072;
    protected const string KeyAndIv = "I am decrypting Cylance's intellectual property.";
    protected Stream _stream;
    protected byte[] _activeKey;
    protected byte[] _activeIV;
    protected bool _loadSectionData;
```



Our own classifier

Engineering Masterpiece!

Let’s build our own classifier so we can dynamically debug and follow the code

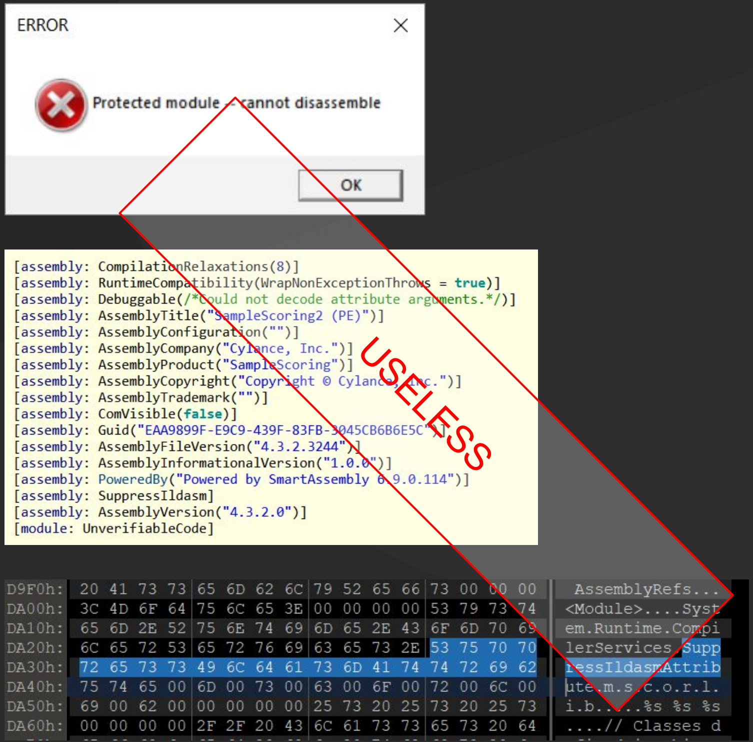
```
static void Main(string[] args)
{

    SampleScoreFactory2PE factory = new SampleScoreFactory2PE();
    SampleScoring2PE scorer = factory.Create("test_model.bin") as SampleScoring2PE;
    Stream test_file = File.Open("mimikatz_with_slight_modification.exe", FileMode.Open);
    Dictionary<string, object> extraData;
    double score = scorer.ComputeScore(test_file, out extraData);
}
```

Watch 1

Name	Value
score	-0.85276468809127071

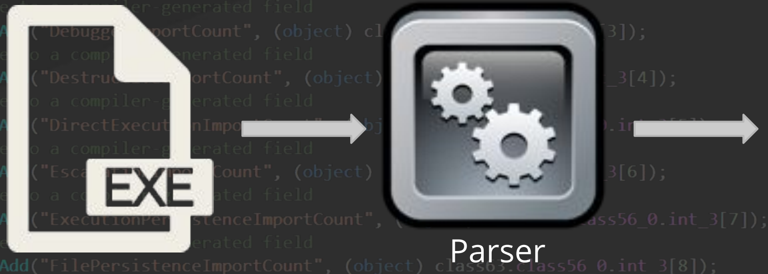
```
LocalAnalyzeItem, C:\Users\Administrator\Desktop\mimikatz_with_slight_modification.exe score -852 detector execution_control
```

Anti-Tampering & Obfuscation



Parsing the Properties

```
this.Observations.Add("AWImportCount", (object) (class63.class56_0.int_0 + class63.class56_0.int_1));
// ISSUE: reference to a compiler-generated field
this.Observations.Add("TransactedImportCount", (object) class63.class56_0.int_2);
// ISSUE: reference to a compiler-generated field
this.Observations.Add("FileImportCount", (object) class63.class56_0.int_3[0]);
// ISSUE: reference to a compiler-generated field
this.Observations.Add("RegistryImportCount", (object) class63.class56_0.int_3[1]);
// ISSUE: reference to a compiler-generated field
this.Observations.Add("DebuggeeImportCount", (object) class63.class56_0.int_3[2]);
// ISSUE: reference    ler-generated field
this.Observations.A  ("Debugg   portCount", (object) cl   3]);
// ISSUE: reference  o a comp   nerated field
this.Observations.A  ("Destru   ortCount", (object)   3[4]);
// ISSUE: reference  o a compiler-g   erated field
this.Observations.A  ("DirectExecut   nImpo
// ISSUE: reference  o a compiler-   erated field
this.Observations.A  ("Esca   Count", (object)   3[6]);
// ISSUE: reference  o a c   ated field
this.Observations.A  ("ExecutionPer   stenceImportCount", (   ass56_0.int_3[7]);
// ISSUE: reference   erated field
this.Observations.Add("FilePersistenceImportCount", (object) class63.class56_0.int_3[8]);
// ISSUE: reference to a compiler-generated field
this.Observations.Add("InjectionImportCount", (object) class63.class56_0.int_3[9]);
// ISSUE: reference to a compiler-generated field
this.Observations.Add("InternetImportCount", (object) class63.class56_0.int_3[10]);
// ISSUE: reference to a compiler-generated field
this.Observations.Add("IPCImportCount", (object) class63.class56_0.int_3[11]);
// ISSUE: reference to a compiler-generated field
this.Observations.Add("ProcessEnumerationImportCount", (object) class63.class56_0.int_3[12]);
// ISSUE: reference to a compiler-generated field
this.Observations.Add("RemoteAudioVideoImportCount", (object) class63.class56_0.int_3[13]);
// ISSUE: reference to a compiler-generated field
this.Observations.Add("RemoteInputImportCount", (object) class63.class56_0.int_3[14]);
// ISSUE: reference to a compiler-generated field
this.Observations.Add("UserEnumerationImportCount", (object) class63.class56_0.int_3[15]);
```



Property	Value
Linker version	5.1
Num sections	5
Section casing	Uppercase
Entropy	0.2315
Timestamp	13102382120

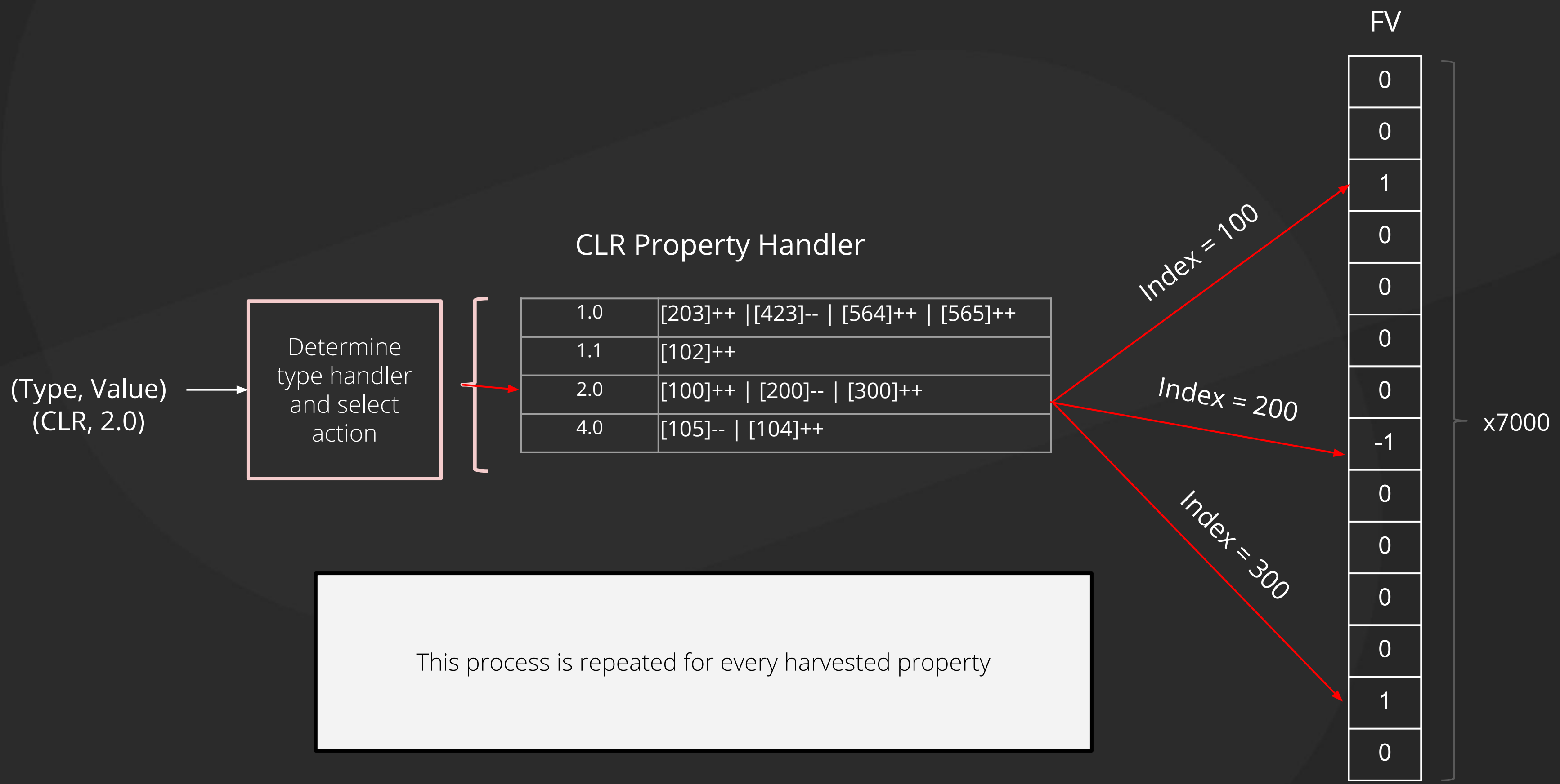
⋮

Max section size	827Kb
CLR version	4.0
#UI imports	98
#Process imports	14
#imports	412

Building the Feature Vector

Property->Feature Converter

Building the Feature Vector



CLR Property Handler

1.0	[203]++ |[423]-- | [564]++ | [565]++
1.1	[102]++
2.0	[100]++ | [200]-- | [300]++
4.0	[105]-- | [104]++

This process is repeated for every harvested property

Linear Algebra, How I Missed You

$$\begin{pmatrix} feature_1 & .. & feature_{7000} \end{pmatrix} \succ x \underbrace{\begin{pmatrix} A_{1,1} & \cdots & A_{1,256} \\ \vdots & & \vdots \\ A_{7000,1} & \cdots & A_{7000,256} \end{pmatrix} \succ tanh \succ x \begin{pmatrix} B_{1,1} & \cdots & B_{1,256} \\ \vdots & \iddots & \vdots \\ B_{256,1} & \cdots & B_{256,256} \end{pmatrix} \succ tanh \succ x \begin{pmatrix} C_{1,1} \\ \vdots \\ C_{256,1} \end{pmatrix} \succ sigmoid \succ}_{\text{The Model Core}} \begin{pmatrix} score \end{pmatrix}$$

Populated by property processing

The Model Core

[-1, 1]
Finally!

White/Black-Listing

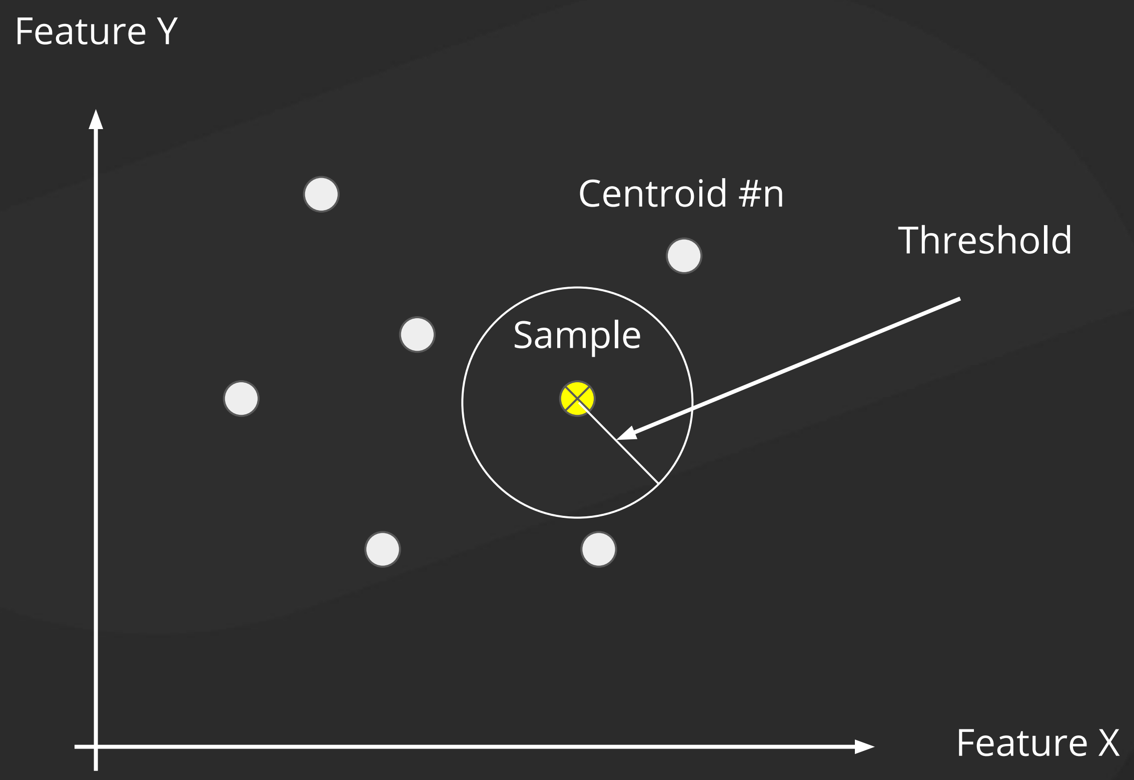
Reduce and normalize features

↓

Check Distance from Centroids

↓

Reverse score if below defined threshold



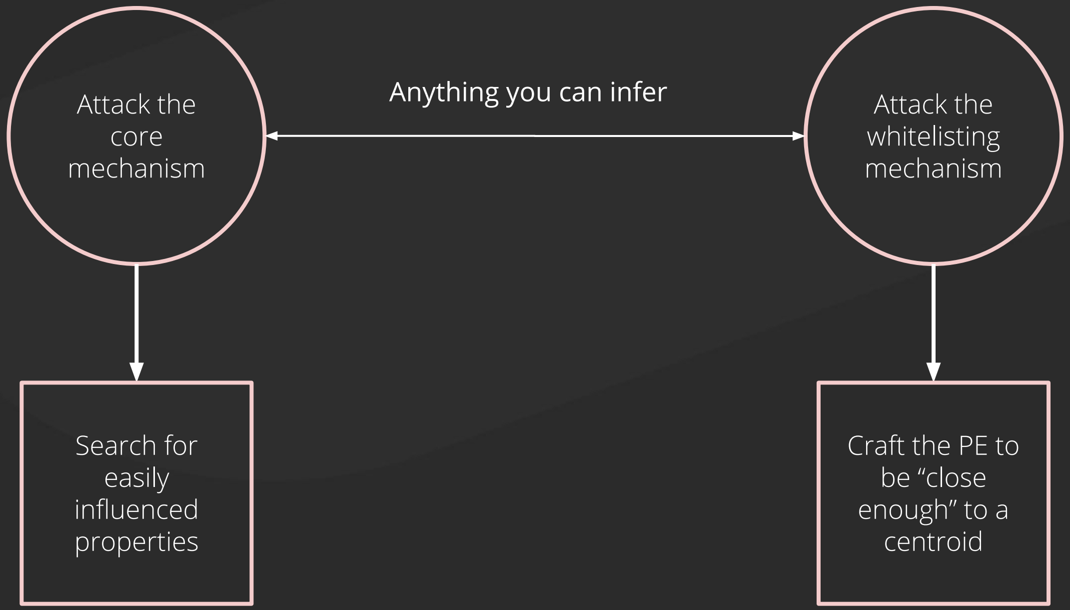
Rocket What?

Watch 1

Name	Value
	{[white, System.Collections.Generic.Dictionary`2[System.String,Cylance.Engine.Core.Centroids.Centroid]]}
	"white"
	Count = 0x00000010
	{[fullspace_dalek_white__trusted_kbhomes_preso_centroid_1-C1CEA4C9136C7909AA1655C4566B21F948514666D90289334A550DDC5DEE56B6, Cylance.Engine.Core.Centroids.Centroid]}
	{[fullspace_dalek_white__trusted_torqtek__packer_pcgaurd_centroid_2-6A6B21AAFB68714406E3E6FA238F51EAACC2C8F8AC6FEA3AAAA90F0352626F28, Cylance.Engine.Core.Centroids.Centroid]}
	{[fullspace_dalek_white__trusted_ztp_centroid_3-BDE4ED001B100FB08FC8C32CFAE4BC181EC4AFB3E7DD464E92ADDEAAF5DA2B9F, Cylance.Engine.Core.Centroids.Centroid]}
	{[fullspace_dalek_white__cylance_inspect_centroid_0-01CF56B85EBF820765D38EFDC9C6677ECF4C08F72E55AEE9B5899662A79F078B, Cylance.Engine.Core.Centroids.Centroid]}
	{[fullspace_dalek_white__trusted_torqtek__packer_pcgaurd_centroid_3-5EBF6E2EC35DA09AA1D9A255AE790660B6050170C68F293ED8019D3661CA8138, Cylance.Engine.Core.Centroids.Centroid]}
	{[fullspace_dalek_white__usmt_centroid_0-C16D6ED90F3CC7F4D1014CA1CFE4FF202EEFF739ECE647CE79B17F7237AB9CF6, Cylance.Engine.Core.Centroids.Centroid]}
	{[fullspace_dalek_white__trusted_ztp_centroid_2-F31D416F55A43ED4B273CD04B0279148E5425C0F88C29BA1AD6745895DEE6E3A, Cylance.Engine.Core.Centroids.Centroid]}
	{[fullspace_dalek_white__trusted_torqtek__packer_pcgaurd_centroid_4-3EAC4DF016685F6A8D3DE0D09A47DCBC144166505E456E7E91EEC8964158856D, Cylance.Engine.Core.Centroids.Centroid]}
	{[fullspace_dalek_white__trusted_ztp_centroid_1-69270CBF3FFCB5FBA3803C84835971C105020BA83B9848A32EF64EB61EF734BF, Cylance.Engine.Core.Centroids.Centroid]}
	{[fullspace_dalek_white__trusted_rocketleague_centroid_0-E3D3D8B2893F36C10A71B203A064FDA3E8400A6D84095C4024A1241234C8E01B, Cylance.Engine.Core.Centroids.Centroid]}
	{[fullspace_dalek_white__trusted_ztp_centroid_4-94A72D94AE2AE5AA9BF1830C725E7977AF43F06C7AD06A7F2F60F128FFF45887, Cylance.Engine.Core.Centroids.Centroid]}
	{[fullspace_dalek_white__trusted_torqtek__packer_pcgaurd_centroid_0-D7EFBEE08E2E6B715F1D25FFB4BC55E3D40A89B5ECD97676273E77C84DA2BC84, Cylance.Engine.Core.Centroids.Centroid]}
	{[fullspace_dalek_white__trusted_torqtek__packer_pcgaurd_centroid_1-33B621DF787852D99C34F3B9ACC7CA10E01AD8E55B740CF1AE7805223A3CCB26, Cylance.Engine.Core.Centroids.Centroid]}
	{[fullspace_dalek_white__trusted_kbhomes_preso_centroid_0-69C6C38C3A1BC4DABC02E344F377EA3DB4E10BC36CF4366AEC88BDAA3FD40898, Cylance.Engine.Core.Centroids.Centroid]}
	{[fullspace_dalek_white__trusted_ztp_centroid_5-7FC4B110F3ED9B7945DC3EC309265115AF41BCCE2FB341D1F598E64DE35D4CE9, Cylance.Engine.Core.Centroids.Centroid]}
	{[fullspace_dalek_white__trusted_ztp_centroid_0-9CE022C5886014AA4C5B13CD1F6F241D561A7889881B830B1ABB356F692D7B11, Cylance.Engine.Core.Centroids.Centroid]}

Hmmm… This could be interesting,
hold that thought

Let's Pause and Hypothesise



Strings Galore

Location of the String Type handler

WOAH, that's a large handler!

```
for (int index = 0; index < this.imagePEFile_0.Strings.Length; ++index)
{
            if (!this.method_26(this.imagePEFile_0.Strings[index].S, 95088, 854069, 0))
                this.method_14(this.list_0[0], 15166118410741992125UL, 2847678, 0);
}
```

Process property function

Strings Galore, Contd.

Hash(string) → Examine

HASH(Str #1)	[203]++ |[423]-- | [564]++ | [565]++
HASH(Str #2)	[1020]++
HASH(Str #3)	[866]++ |[533]--
HASH(Str #4)	[53]++ |[4]-- | [2464]++ | [5432]++
HASH(Str #5)	[4500]++ |[3223]--
HASH(Str #6)	[10]++ |[400]-- | [3444]++
HASH(Str #7)	[453]++

⋮

HASH(Str #854063)	[23]++ |[25]-- | [55]++
HASH(Str #854064)	[6088]++ |[48]-- | [4332]++ | [2]++
HASH(Str #854065)	[100]++
HASH(Str #854066)	[1335]++ |[3234]--
HASH(Str #854067)	[64]++ |[233]-- | [44]++
HASH(Str #854068)	[12]++ |[14]--
HASH(Str #854069)	[6778]++

The Hypothesis

Strings have the potential for disproportionate impact on the feature vector

The Whitelist provides a hint as to what type of executables are “good” (e.g. Rocket League) and may have been used to retrain the model at a later stage

If we strip the strings from the good PEs and **carefully inject them into a malicious payload**, we may be able to fool the model, as they will overpower the effect of “negative” properties. Note that the model does not regard “attacker economics”.

Note that we are NOT aiming to fool the whitelisting mechanism, rather the main model!

This would never work, right?

```
⋮
66 1B C9 66 23 C2 49 8B D7 66 81 E1 00 04 66 0B    f.Éf#ÂI‹×f.á..f.
C8 0F BF 45 0A 89 44 24 30 48 8B 45 00 66 89 4C    È.¿E.‰D$0H‹E.f‰L
24 28 48 8B CE 48 89 44 24 20 E8 39 68 00 00 33    $(H‹ÎH‰D$ è9h..3
C9 48 8B D8 48 3B C1 0F 84 97 0C 00 00 0F B7 40    ÉH‹ØH;Á.„—....·@
30 66 3B 45 0A 7D 04 66 89 45 0A 45 33 ED 41 3A    0f;E.}.f‰E.E3íA:
FD 74 19 0F B6 4B 3A 0F B6 84 24 89 00 00 00 41    ýt..¶K:.¶„$‰...A
```

+

```
52 75 73 73 69 61 6E 0D 0A 74 0D 0A 52 65 6D 6F    Russian..t..Remo
76 65 0D 0A 39 38 75 0D 0A 44 65 6C 65 74 65 0D    ve..98u..Delete.
0A 6D 64 35 0D 0A 35 35 35 35 0D 0A 43 68 61 6E    .md5..5555..Chan
6E 65 6C 0D 0A 64 65 73 63 72 69 70 74 69 6F 6E    nel..description
0D 0A 4D 45 54 0D 0A 25 73 0D 0A 30 31 32 33 34    ..MET..%s..01234
35 36 37 38 39 41 42 43 44 45 46 0D 0A 77 69 6E    56789ABCDEF..win
33 32 0D 0A 23 0D 0A 25 30 32 78 0D 0A 29 0D 0A    32..#..%02x..)..
46 54 0D 0A 25 30 64 0D 0A 31 35 33 36 0D 0A 52    FT..%0d..1536..R
45 44 0D 0A 4C 6F 67 0D 0A 31 30 31 30 0D 0A 42    ED..Log..1010..B
⋮
```

=

```
Russian
t
Remove
98u
Delete
md5
5555
Channel
description
MET
%s
0123456789ABCDEF
win32
#
%02x
)
FT
%0d
1536
RED
Log
1010
B6
B14
UG
DLL
```

Let’s have a look...

Summon the Malware Hordes

Malware	Score Before	Score After
CoinMiner	-826	884
Dridex	-999	996
Emotet	-923	625
Gh0stRAT	-975	998
Kovter	-999	856
Nanobot	971	999
Pushdo	-999	999
Qakbot	-998	991
Trickbot	-973	774
Zeus	-997	997

Tests on 384 samples from theZoo repository:

88.54% of malware passed as benign
Average score before treatment = -0.92 (min is -1)
Average score after mutation = 0.75 (max is 1)
Average change in score = +1.67 (out of a range of 2).

Publication & Cylance’s Response

July 21st, Cylance’s Threat Vector

> …researchers publicly disclosed a specific bypass of CylancePROTECT®. We verified the issue was not a universal bypass as reported, but rather a technique that allowed for one of the anti-malware components of the product to be bypassed in certain circumstances. The issue has been resolved for cloud-based scoring and a new agent will be rolled out to endpoints in the next few days.

We are still waiting for a fix for the SmartAV product…

Kim Zetter @KimZetter · Aug 14
Ryan Permeh, founder/chief scientist at Cylance, has responded and said SmartAV (the consumer version) trails behind corporate version in update rollouts. He said the issue was fixed in the corp version of their AV and is checking now on what's going on with the consumer version.
7
Show this thread

Kim Zetter @KimZetter · Aug 14
I already reached out to Cylance and will update if I hear more.
2 7
Show this thread

Kim Zetter @KimZetter · Aug 14
Cylance pr told me the company fixed bypass I wrote about last month and rolled out fix to customers July 26. But researchers who discovered issue told me they installed fresh version of SmartAV agent this wk and can still bypass it using same trick

Researchers Easily Trick Cylance's AI-Based Antivirus Into Thinking …
By taking strings from an online gaming program and appending them to malicious files, researchers were able to trick Cylance's AI-based antivirus...
vice.com

Questions?

SKYLIGHT

ZERO COMPROMISE

Thank You!

adi@skylightcyber.com