McAfee™

Together is power.

Machine Learning

What is it, what it is not,
how does it work and why
we need it…

Sylvain Dumas – SE

Are Artificial Intelligence and Machine Learning the same?

Short answer – NO

However – Machine Learning has been used in Artificial Intelligence

Unfortunately there is a lot of confusion about this

Artificial Intelligence vs Machine Learning

Artificial Intelligence:

“the term "artificial intelligence" is applied when a machine mimics "cognitive" functions that humans associate with other human minds, such as "learning" and "problem solving".”

(Source Wikipedia)

Machine Learning:

machine learning explores the study and construction of algorithms that can learn from and make predictions on data

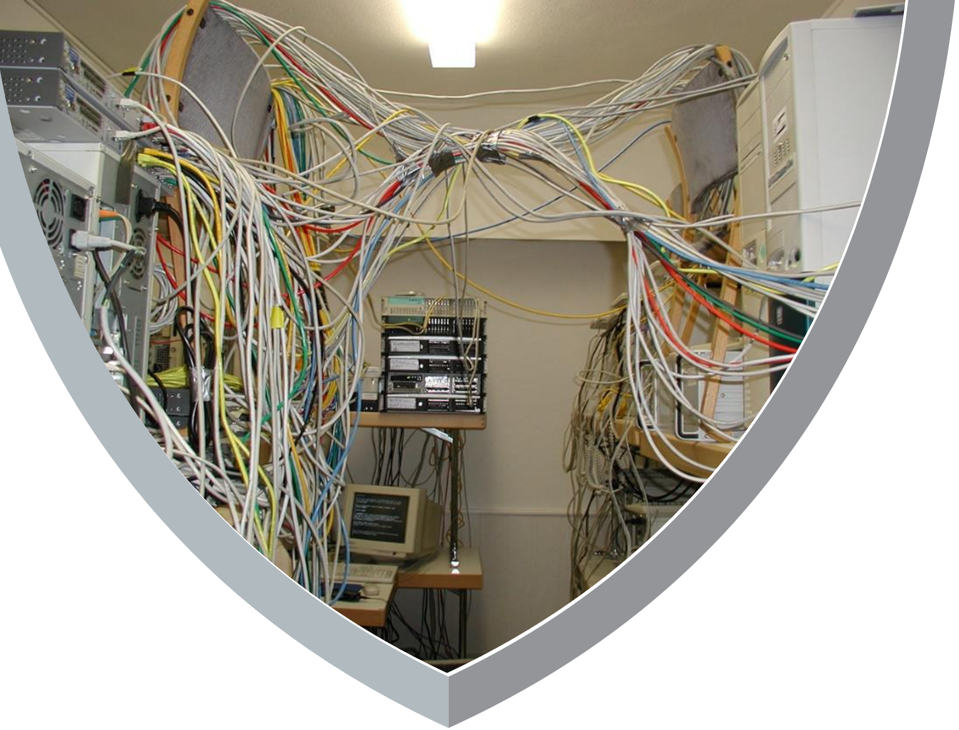
(Source Wikipedia)

Artificial Intelligence – Greek Mythology

A greek myth contain a story of mechanical men designed to mimic our own behaviour.

Talos was designed to protect Europa in Crete from pirates.

Talos was forged by the blacksmith god, Hephaestus, who gifted him to king Minos of Crete. Talos had a single vein, through which ichor, the divine blood of the Olympians, flowed. This vein was plugged by just one bronze nail.



Detection Challenges

The Age Of “Signatures” Is Fading…

Signatures identify with near certainty that an object is either malicious or clean

This technique is **reactive** by nature. Although very precise, the sheer number and growth in malware variants is making this **unsustainable**

Malware authors are continuously monitoring AV vendor detection and releasing **new variants**

Use of commercial, open source or underground packers and protectors makes **repacking new variants trivial**

Detection Challenges

What did this Snake eat for lunch? ;)

Image: https://www2.picturepush.com

Unpacking Challenges

Think of it as a file, inside another executable file…. which can be inside another executable file

Think Russian dolls (Matryoshka)

When executed, the ‘outer’ executable will unpack the contents of the ‘inner’ executable into memory and execute it.

The inner most executable is the ‘real’ executable!

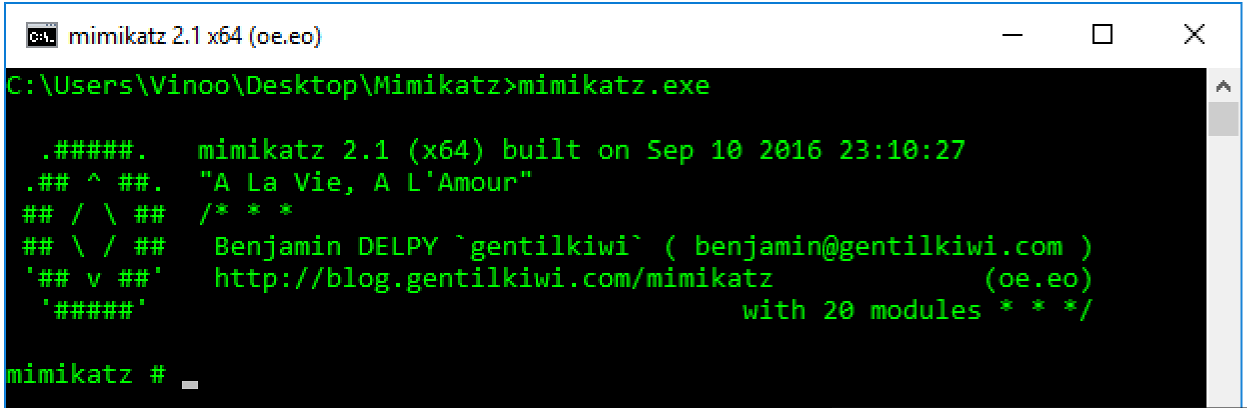
Image: https://www.pinterest.com

Field Example – What is Mimikatz

It's well known to extract plaintexts passwords, hash, PIN code and kerberos tickets from memory. mimikatz can also perform pass-the-hash, pass-the-ticket, build Golden tickets, play with certificates or private keys, vault, ... maybe make coffee?

Source: http://blog.gentilkiwi.com/mimikatz

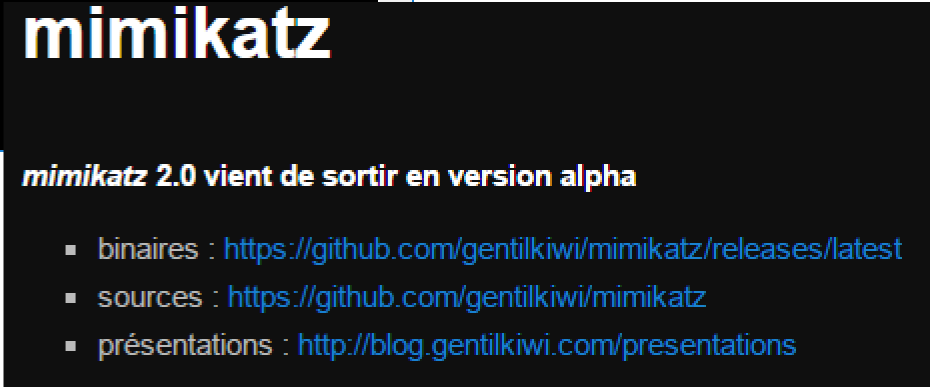
Field Example - Mimikatz



```
C:\Users\Vinoo\Desktop\Mimikatz>mimikatz.exe

  .#####.   mimikatz 2.1 (x64) built on Sep 10 2016 23:10:27
 .## ^ ##.  "A La Vie, A L'Amour"
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                     with 20 modules * * */

mimikatz # _
```



mimikatz

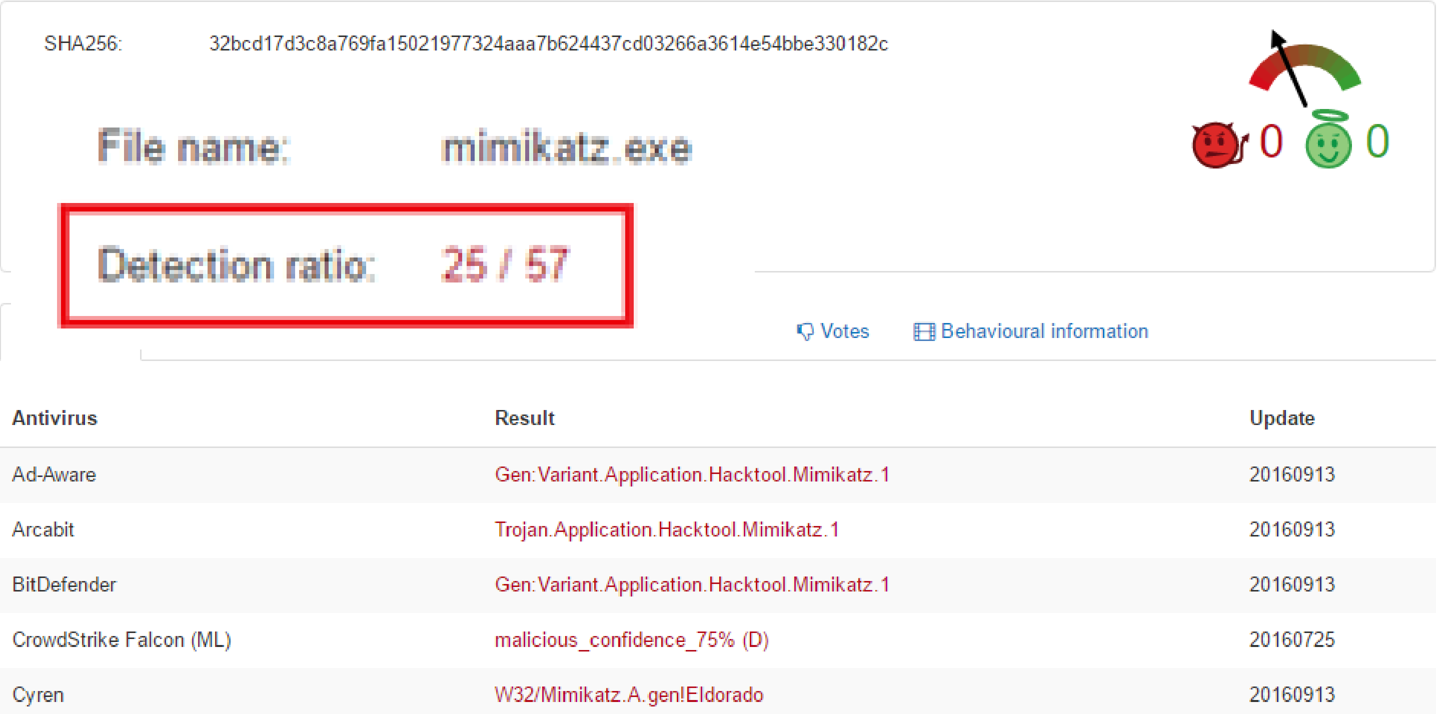
***mimikatz* 2.0 vient de sortir en version alpha**

- binaires : https://github.com/gentilkiwi/mimikatz/releases/latest
- sources : https://github.com/gentilkiwi/mimikatz
- présentations : http://blog.gentilkiwi.com/presentations

Source: http://blog.gentilkiwi.com/mimikatz

Mimikatz – Compiled Binary

virustotal



SHA256: 32bcd17d3c8a769fa15021977324aaa7b624437cd03266a3614e54bbe330182c

File name: mimikatz.exe

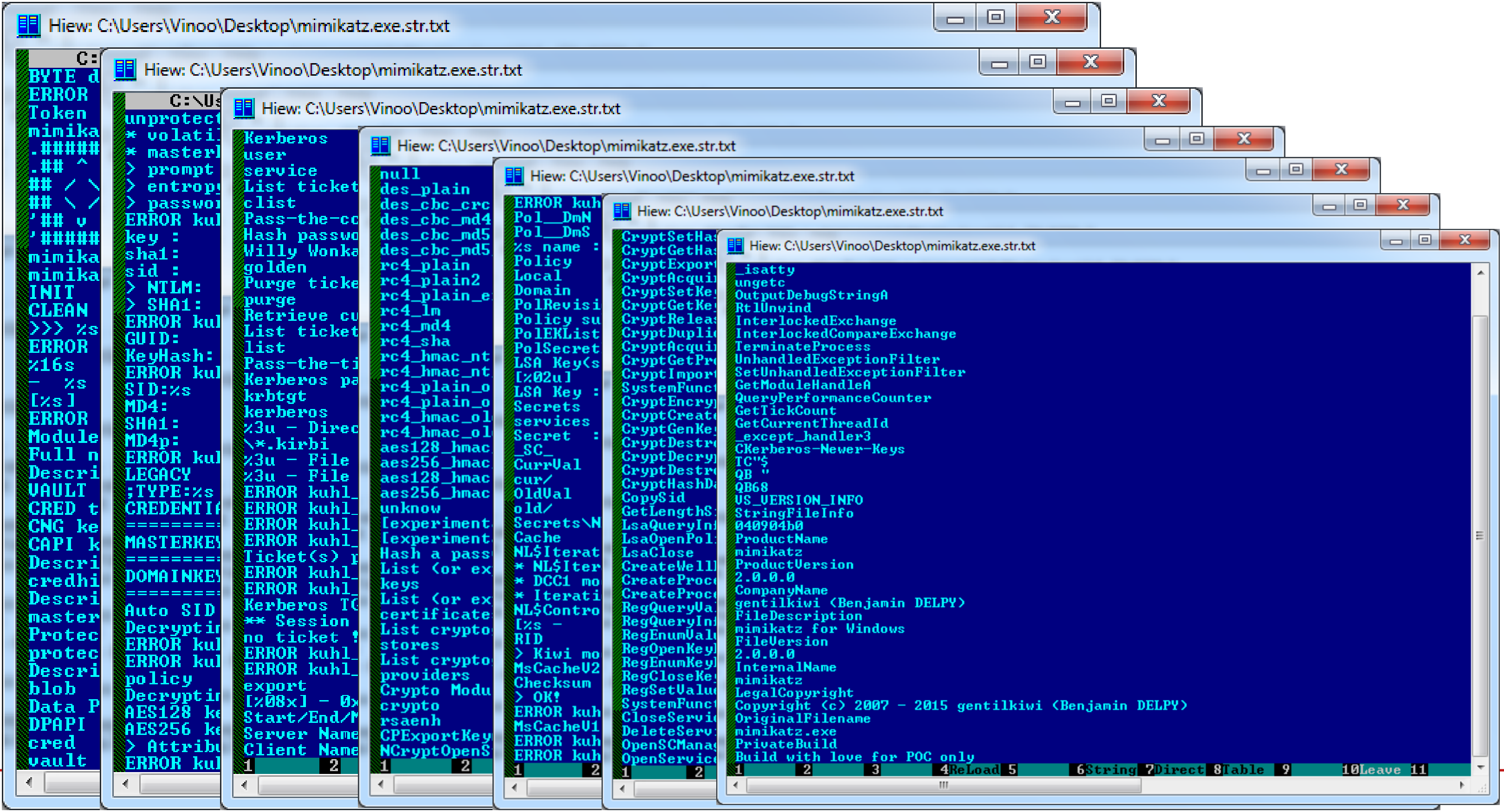
Detection ratio: 25 / 57

Votes

Behavioural information

Antivirus	Result	Update
Ad-Aware	Gen:Variant.Application.Hacktool.Mimikatz.1	20160913
Arcabit	Trojan.Application.Hacktool.Mimikatz.1	20160913
BitDefender	Gen:Variant.Application.Hacktool.Mimikatz.1	20160913
CrowdStrike Falcon (ML)	malicious_confidence_75% (D)	20160725
Cyren	W32/Mimikatz.A.gen!Eldorado	20160913

Mimikatz – Compiled Binary



```
Hiew: C:\Users\Vinoo\Desktop\mimikatz.exe.str.txt
C:
BYTE d
ERROR
Token
mimika
.#####
.## ^
## / \
## \ /
'## v
'#####
mimika
mimika
INIT
CLEAN
>>> %s
ERROR
%16s
-   %s
[%s]
ERROR
Module
Full n
Descri
VAULT
CRED t
CNG ke
CAPI k
Descri
credhi
Descri
master
Protec
protec
Descri
blob
Data P
DPAPI
cred
vault
```

```
Hiew: C:\Users\Vinoo\Desktop\mimikatz.exe.str.txt
C:\Us
unprotect
* volatil
* masterk
> prompt
> entropy
> passwor
ERROR kul
key :
sha1:
sid :
> NTLM:
> SHA1:
ERROR kul
GUID:
KeyHash:
ERROR kul
SID:%s
MD4:
SHA1:
MD4p:
ERROR kul
LEGACY
;TYPE:%s
CREDENTIA
==========
MASTERKEY
==========
DOMAINKEY
==========
Auto SID
Decryptin
ERROR kul
ERROR kul
policy
Decryptin
AES128 ke
AES256 ke
> Attribu
ERROR kul
```

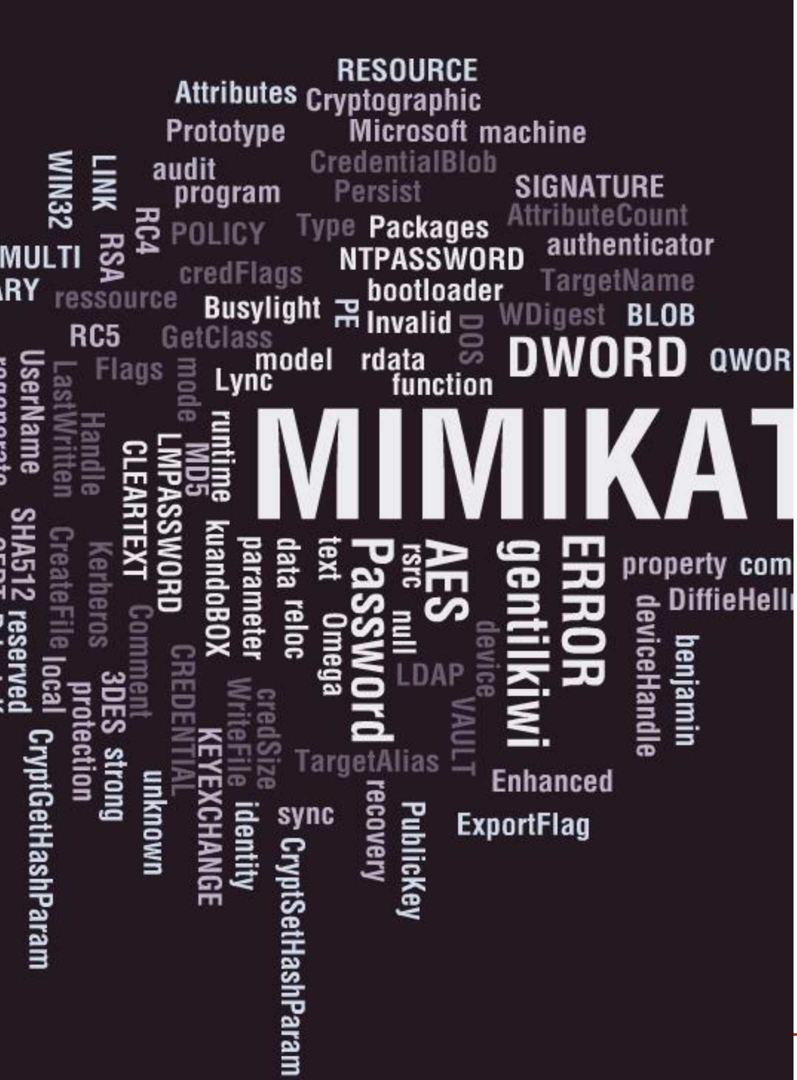
```
Hiew: C:\Users\Vinoo\Desktop\mimikatz.exe.str.txt
Kerberos
user
service
List ticket
clist
Pass-the-cc
Hash passwo
Willy Wonka
golden
Purge ticke
purge
Retrieve cu
List ticket
list
Pass-the-ti
Kerberos pa
krbtgt
kerberos
%3u - Direc
\*.kirbi
%3u - File
%3u - File
ERROR kuhl_
ERROR kuhl_
ERROR kuhl_
ERROR kuhl_
Ticket(s) p
ERROR kuhl_
ERROR kuhl_
Kerberos TG
** Session
no ticket !
ERROR kuhl_
ERROR kuhl_
export
[%08x] - 0x
Start/End/M
Server Name
Client Name
1          2
```

```
Hiew: C:\Users\Vinoo\Desktop\mimikatz.exe.str.txt
null
des_plain
des_cbc_crc
des_cbc_md4
des_cbc_md5
des_cbc_md5
rc4_plain
rc4_plain2
rc4_plain_e
rc4_lm
rc4_md4
rc4_sha
rc4_hmac_nt
rc4_hmac_nt
rc4_plain_o
rc4_plain_o
rc4_hmac_ol
rc4_hmac_ol
aes128_hmac
aes256_hmac
aes128_hmac
aes256_hmac
unknow
[experiment
[experiment
Hash a pass
List (or ex
keys
List (or ex
certificate
List crypto
stores
List crypto
providers
Crypto Modu
crypto
rsaenh
CPExportKey
NCryptOpenS
1          2
```

```
Hiew: C:\Users\Vinoo\Desktop\mimikatz.exe.str.txt
ERROR kuh
Pol__DmN
Pol__DmS
%s name :
Policy
Local
Domain
PolRevisi
Policy su
PolEKList
PolSecret
LSA Key(s
[%02u]
LSA Key :
Secrets
services
Secret  :
_SC_
CurrVal
cur/
OldVal
old/
Secrets\N
Cache
NL$Iterat
* NL$Iter
* DCC1 mo
* Iterati
NL$Contro
[%s -
RID
> Kiwi mo
MsCacheV2
Checksum
> OK!
ERROR kuh
MsCacheV1
ERROR kuh
ERROR kuh
1        2
```

```
Hiew: C:\Users\Vinoo\Desktop\mimikatz.exe.str.txt
CryptSetHas
CryptGetHas
CryptExport
CryptAcquir
CryptSetKey
CryptGetKey
CryptReleas
CryptDuplic
CryptAcquir
CryptGetPro
CryptImport
SystemFunct
CryptEncryp
CryptCreate
CryptGenKey
CryptDestro
CryptDecryp
CryptDestro
CryptHashDa
CopySid
GetLengthSi
LsaQueryInf
LsaOpenPoli
LsaClose
CreateWellK
CreateProce
CreateProce
RegQueryVal
RegQueryInf
RegEnumValu
RegOpenKeyE
RegEnumKeyE
RegCloseKey
RegSetValue
SystemFunct
CloseServic
DeleteServi
OpenSCManag
OpenService
1          2
```

```
Hiew: C:\Users\Vinoo\Desktop\mimikatz.exe.str.txt
_isatty
ungetc
OutputDebugStringA
RtlUnwind
InterlockedExchange
InterlockedCompareExchange
TerminateProcess
UnhandledExceptionFilter
SetUnhandledExceptionFilter
GetModuleHandleA
QueryPerformanceCounter
GetTickCount
GetCurrentThreadId
_except_handler3
CKerberos-Newer-Keys
TC"$
QB "
QB68
VS_VERSION_INFO
StringFileInfo
040904b0
ProductName
mimikatz
ProductVersion
2.0.0.0
CompanyName
gentilkiwi (Benjamin DELPY)
FileDescription
mimikatz for Windows
FileVersion
2.0.0.0
InternalName
mimikatz
LegalCopyright
Copyright (c) 2007 - 2015 gentilkiwi (Benjamin DELPY)
OriginalFilename
mimikatz.exe
PrivateBuild
Build with love for POC only
1       2       3       4ReLoad 5       6String 7Direct 8Table  9       10Leave 11
```



Mimikatz Detection

- Native binary has thousands of interesting features!
- Resources, strings, packer & compiler details, compile time, API & function calls etc. readily available for authoring signatures

Image: http://www.abcya.com/word_clouds.htm

Mimikatz – Compiled Binary

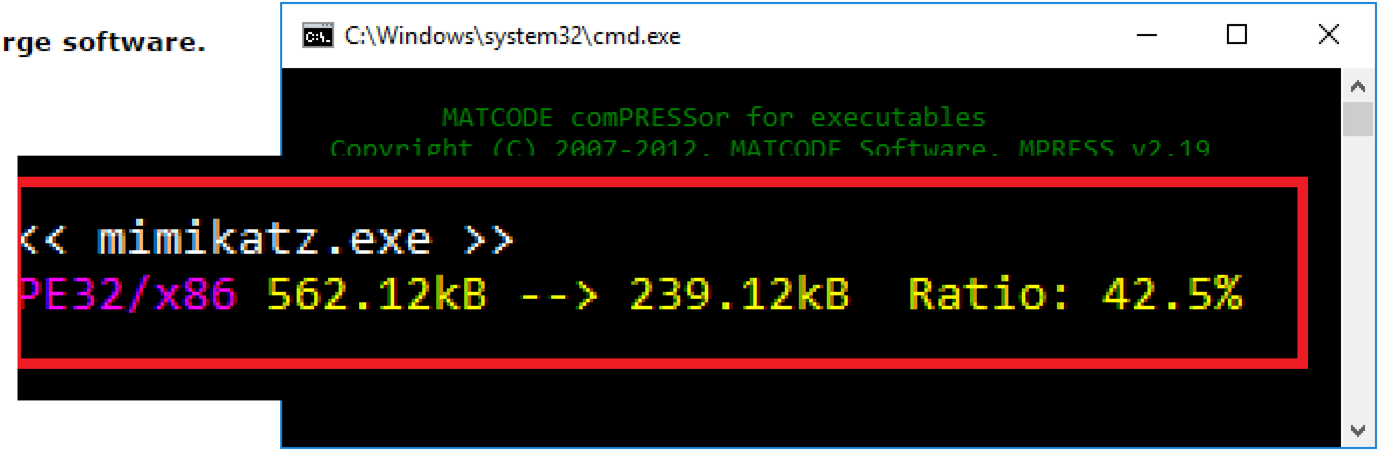
MPRESS free!!!

Version: 2.19

MPRESS is a free, high-performance executable packer for PE32/PE32+/.NET/MAC-DARWIN executable formats!

MPRESS makes programs and libraries smaller, and decrease start time when the application loaded from a slow removable media or from the network. It uses in-place decompression technique, which allows to decompress the executable without memory overhead or other drawbacks; it also protects programs against reverse engineering by non-professional hackers. Programs compressed with MPRESS run exactly as before, with no runtime performance penalties.

MPRESS is absolutely free of charge software.

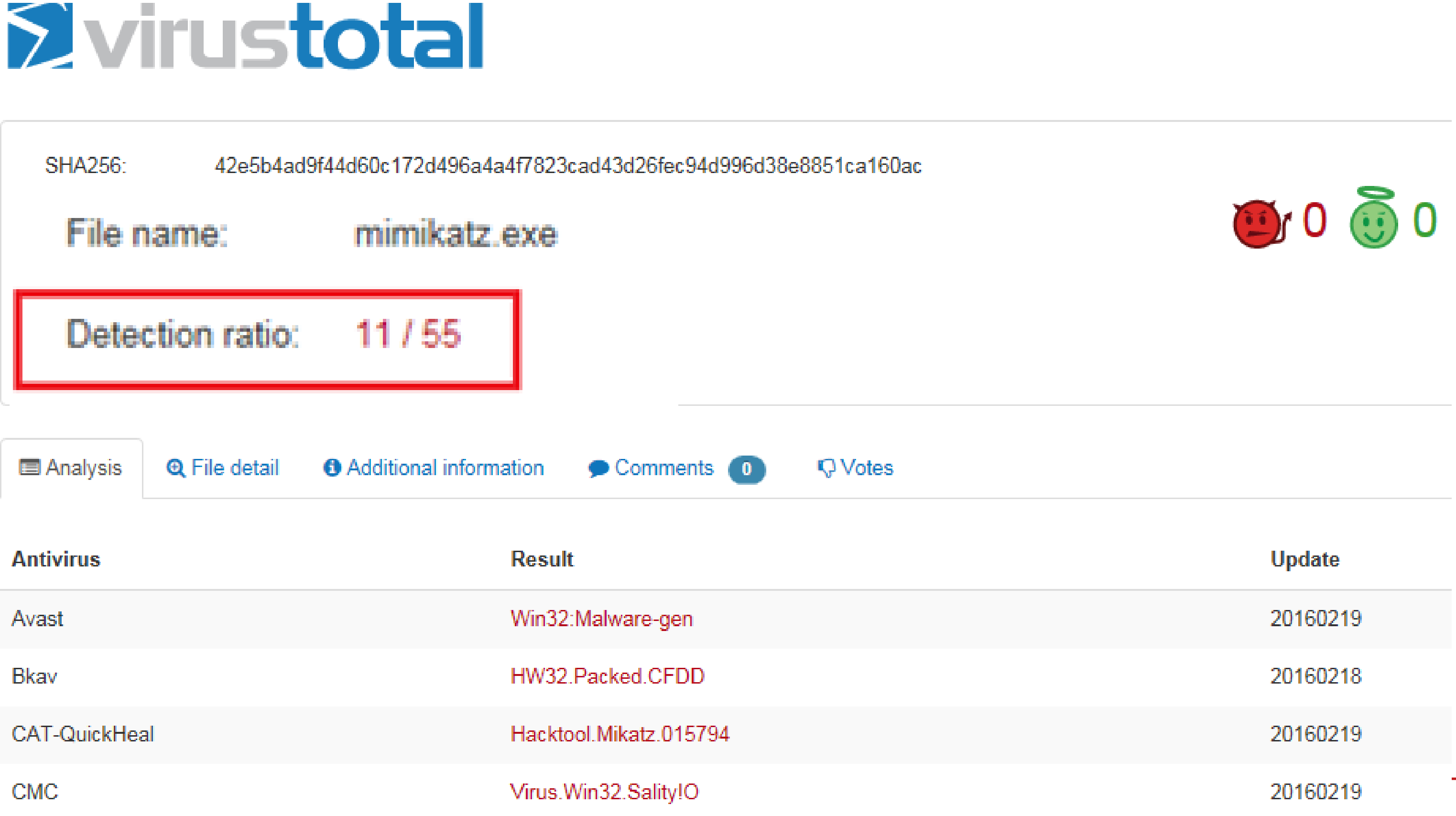


```
          MATCODE comPRESSor for executables
   Copyright (C) 2007-2012, MATCODE Software, MPRESS v2.19

<< mimikatz.exe >>
PE32/x86 562.12kB --> 239.12kB  Ratio: 42.5%
```

Source: https://autohotkey.com/mpress/mpress_web.htm

Mimikatz – Packed With MPRESS

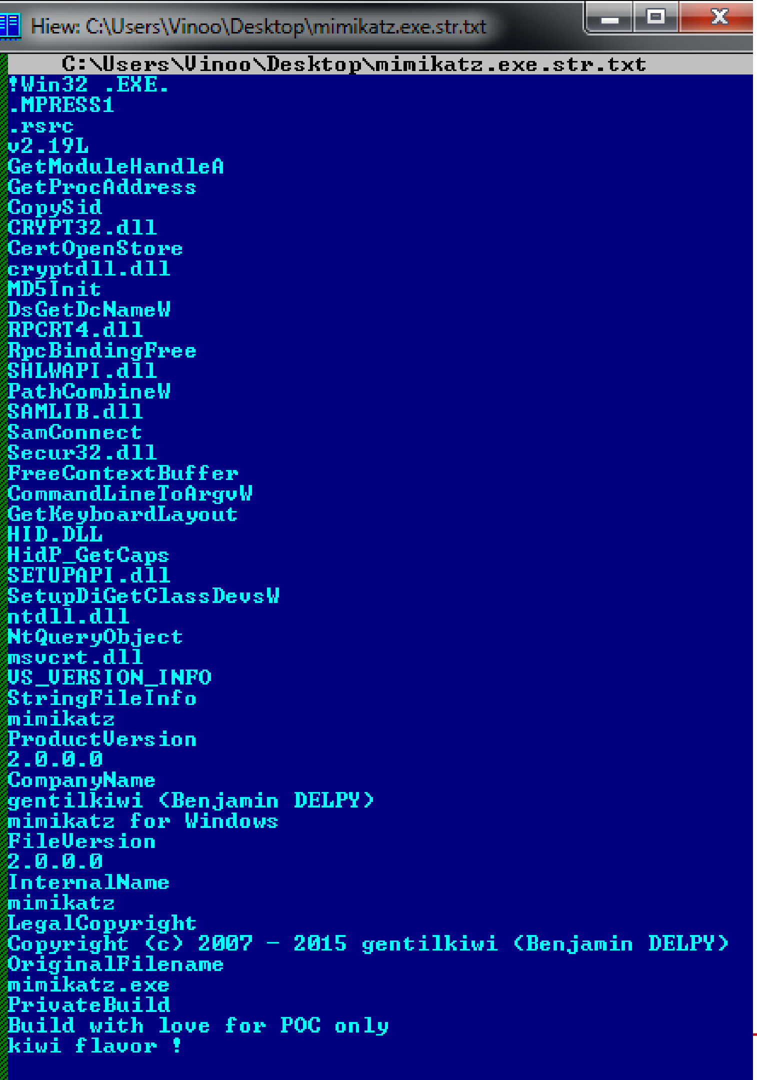


SHA256: 42e5b4ad9f44d60c172d496a4a4f7823cad43d26fec94d996d38e8851ca160ac

File name: mimikatz.exe

Detection ratio: 11 / 55

Antivirus	Result	Update
Avast	Win32:Malware-gen	20160219
Bkav	HW32.Packed.CFDD	20160218
CAT-QuickHeal	Hacktool.Mikatz.015794	20160219
CMC	Virus.Win32.Sality!O	20160219



```
C:\Users\Vinoo\Desktop\mimikatz.exe.str.txt
!Win32 .EXE.
.MPRESS1
.rsrc
v2.19L
GetModuleHandleA
GetProcAddress
CopySid
CRYPT32.dll
CertOpenStore
cryptdll.dll
MD5Init
DsGetDcNameW
RPCRT4.dll
RpcBindingFree
SHLWAPI.dll
PathCombineW
SAMLIB.dll
SamConnect
Secur32.dll
FreeContextBuffer
CommandLineToArgvW
GetKeyboardLayout
HID.DLL
HidP_GetCaps
SETUPAPI.dll
SetupDiGetClassDevsW
ntdll.dll
NtQueryObject
msvcrt.dll
VS_VERSION_INFO
StringFileInfo
mimikatz
ProductVersion
2.0.0.0
CompanyName
gentilkiwi (Benjamin DELPY)
mimikatz for Windows
FileVersion
2.0.0.0
InternalName
mimikatz
LegalCopyright
Copyright (c) 2007 - 2015 gentilkiwi (Benjamin DELPY)
OriginalFilename
mimikatz.exe
PrivateBuild
Build with love for POC only
kiwi flavor !
```

Mimikatz – Post MPRESS

- Previously available static features are destroyed and made unavailable by the packer!
- Limited choices available for authoring a generic signature.

Machine Learning Approaches

Early “Machine Learning” – Bayesian detection for SPAM

Easy / Naïve statistical analysis

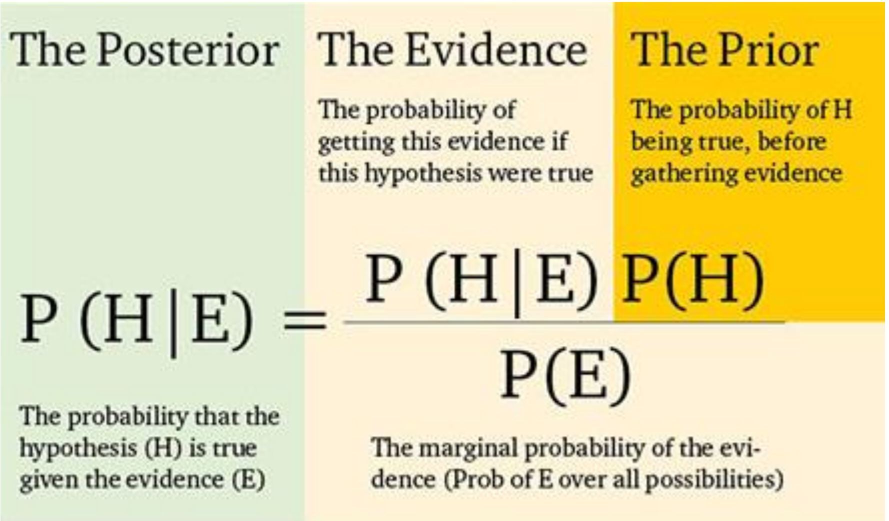
Looking at the word “click” will catch 79% of spam with only 1.2% of false positives

Looking for bad words in proportion of good words

> vi@gra or v1@gr@ vs Viagra – Bayesian will learn the differences and adjust accordingly

Relations to words – the word Act, versus Act now – declared a higher probability of spam.

Bayesian poisoning



The Posterior

The Evidence

The probability of getting this evidence if this hypothesis were true

The Prior

The probability of H being true, before gathering evidence

$$P(H|E) = \frac{P(H|E)\,P(H)}{P(E)}$$

The probability that the hypothesis (H) is true given the evidence (E)

The marginal probability of the evidence (Prob of E over all possibilities)

Source - http://www.paulgraham.com/spam.html

What is a Neural Network

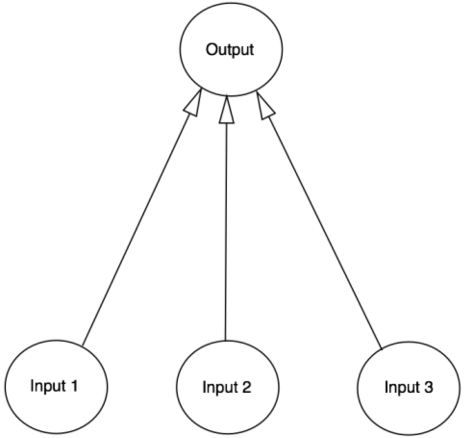
The human brain consists of 100 billion cells called neurons, connected together by synapses. If sufficient synaptic inputs to a neuron fire, that neuron will also fire. We call this process “thinking”.

We can model this process by creating a neural network on a computer.

It’s not necessary to model the biological complexity of the human brain at a molecular level, just its higher level rules.

We use a mathematical technique called matrices, which are grids of numbers.

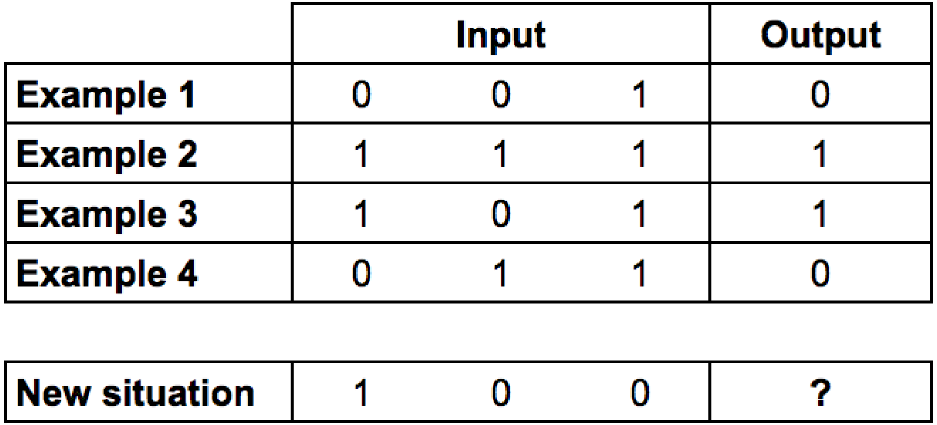
To make it really simple, we will just model a single neuron, with three inputs and one output.



Source – https://medium.com/technology-invention-and-more/how-to-build-a-simple-neural-network-in-9-lines-of-python-code-cc8f23647ca1

How does a Neural Network work

We’re going to train the neuron to solve the problem below. The first four examples are called a training set. Can you work out the pattern? Should the ‘?’ be 0 or 1?



	Input			Output
Example 1	0	0	1	0
Example 2	1	1	1	1
Example 3	1	0	1	1
Example 4	0	1	1	0

New situation	1	0	0	?

Source – https://medium.com/technology-invention-and-more/how-to-build-a-simple-neural-network-in-9-lines-of-python-code-cc8f23647ca1

Creating a model and training it

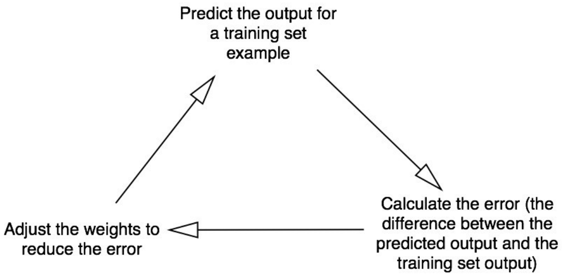
We will give each input a weight, which can be a positive or negative number. An input with a large positive weight or a large negative weight, will have a strong effect on the neuron’s output. Before we start, we set each weight to a random number. Then we begin the training process:

Take the inputs from a training set example, adjust them by the weights, and pass them through a special formula to calculate the neuron’s output.

Calculate the error, which is the difference between the neuron’s output and the desired output in the training set example.

Depending on the direction of the error, adjust the weights slightly.

Repeat this process 10,000 times.


Predict the output for a training set example
Calculate the error (the difference between the predicted output and the training set output)
Adjust the weights to reduce the error

Source – https://medium.com/technology-invention-and-more/how-to-build-a-simple-neural-network-in-9-lines-of-python-code-cc8f23647ca1

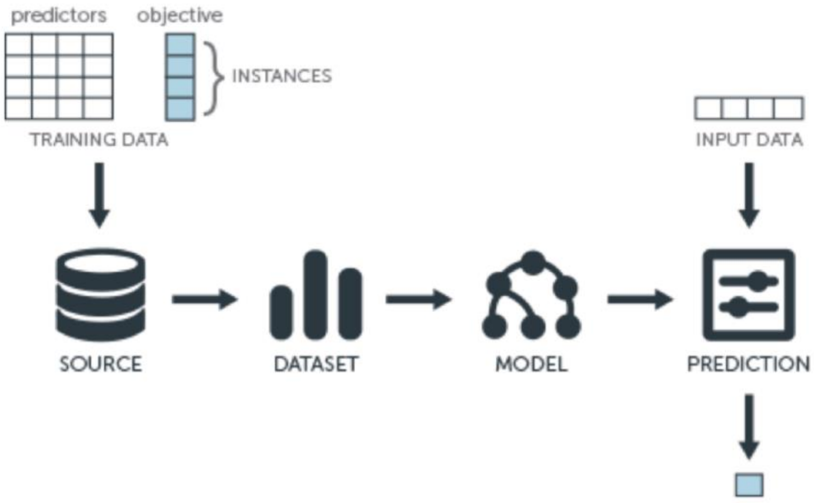
Training the model

Eventually the weights of the neuron will reach an optimum for the training set. If we allow the neuron to think about a new situation, that follows the same pattern, it should make a good prediction.

This process is called back propagation.

First the neural network assigned itself random weights, then trained itself using the training set. Then it considered a new situation [1, 0, 0] and predicted 0.99993704. The correct answer was 1. So very close!

Traditional computer programs normally can’t learn. What’s amazing about neural networks is that they can learn, adapt and respond to new situations. Just like the human mind.



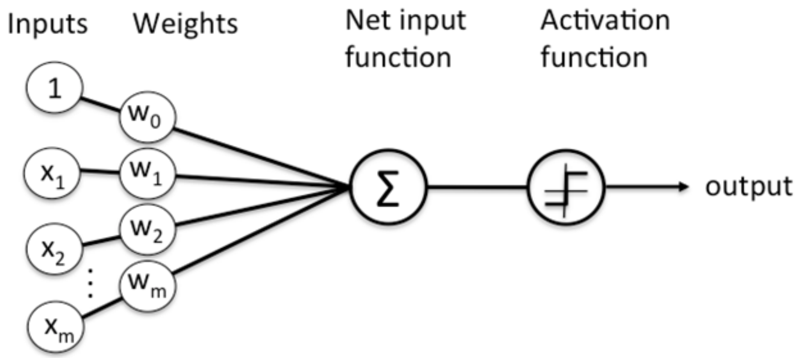
Neural Network

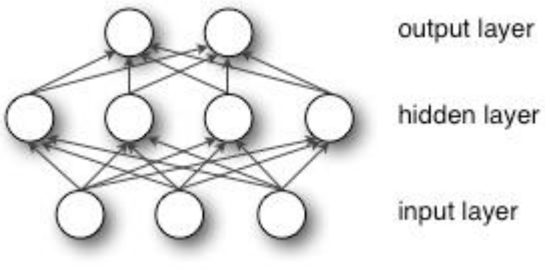
Computer system designed to work by classifying information the same way a human brain would

It can be taught to recognize, for example, images and classify them according to what they contain.

Essentially it works on a system of probability.

Here's a diagram of what one node might look like.





Source – Forbes – Bernard Marr 2016 - Source – deeplearning4j.org

Deep Neural Network

Deep-learning networks are distinguished from the more commonplace single-hidden-layer neural networks by their **depth**; that is, the number of node layers through which data passes in a multistep process of pattern recognition.

Traditional machine learning relies on shallow nets, composed of one input and one output layer, and at most one hidden layer in between. More than three layers (including input and output) qualifies as “deep” learning. So deep is a strictly defined, technical term that means more than one hidden layer.

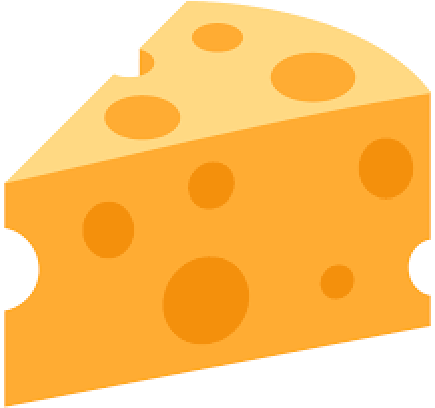
In deep-learning networks, each layer of nodes trains on a distinct set of features based on the previous layer’s output. The further you advance into the neural net, the more complex the features your nodes can recognize, since they aggregate and recombine features from the previous layer.

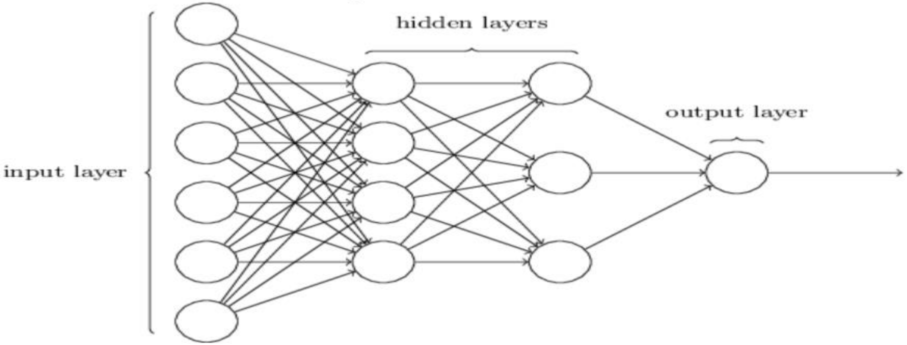
Source – deeplearning4j.org

Deep Neural Network

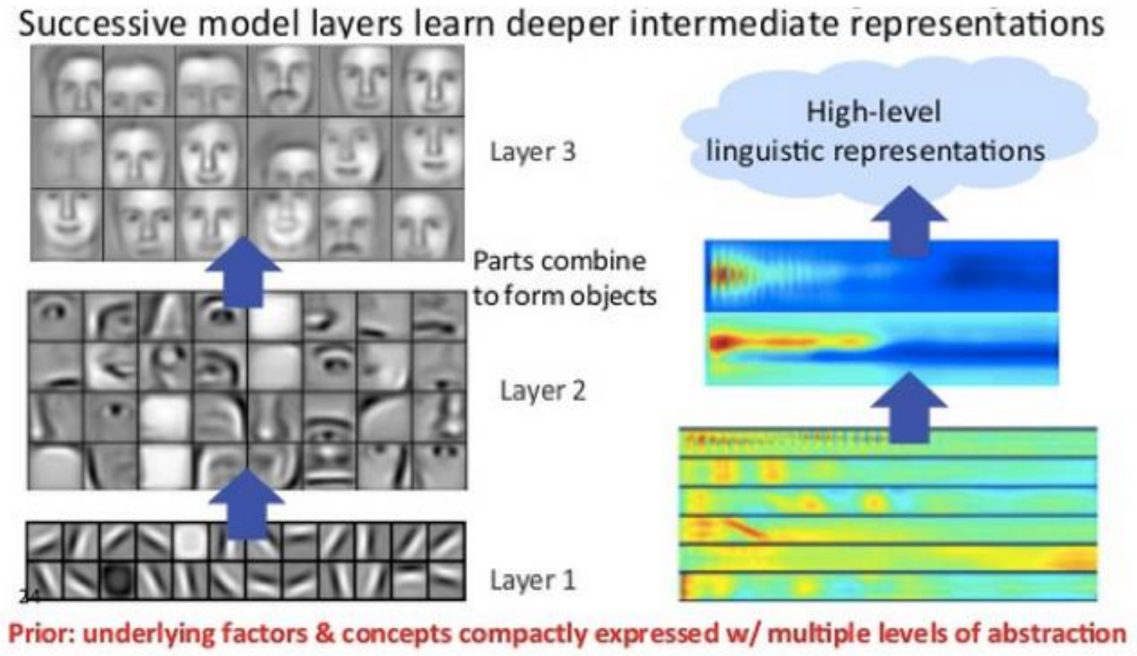
The weekend is coming up, and you've heard that there's going to be a cheese festival in your city. You like cheese, and are trying to decide whether or not to go to the festival. You might make your decision by weighing up three factors:

- Is the weather good?
- Does your boyfriend or girlfriend want to accompany you?
- Is the festival near public transit? (You don't own a car).





Deep Neural Network



Successive model layers learn deeper intermediate representations

Prior: underlying factors & concepts compactly expressed w/ multiple levels of abstraction

Source – deeplearning4j.org

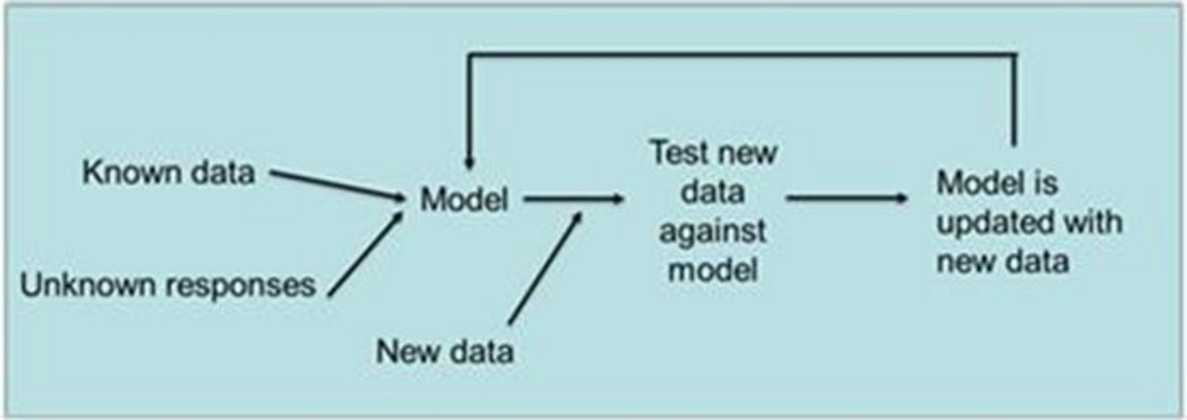
Machine Learning in the context of security

Modeling

Clustering

Pre-Execution

Post-Execution



Sources Of Features

- **Static Analysis** (file type, resources, meta-data etc.)
- **Fuzzy Hashing** (identical byte or checksum sequences)
- **Import Address Hash** (Function calls, Order of function calls)
- **Dynamic Analysis** (file system, registry, network behaviors)
- **Memory Analysis** (process or system memory analysis)

Leveraging Multiple Sources Of Knowledge

- ✓ Identify a suspicious characteristic or activity
- ✓ The object is given a reputation and confidence level if existing signatures based methods don’t detect
- ✓ **Pre-execution:** Static file feature extraction (file type, import hash, entry point, resources, strings, packer & compiler details, compile time, API’s, section names etc.)
- ✓ **Post-execution:** Behavioral features and Memory analysis (behavioral sequence, process tree, file system, registry events, network communication events, mutex, strings from memory etc.)

A hybrid approach provides the best classification rates!

Extracting Static Features

Ransomware: CTB-Locker (pre-execution)

- File type, resources & strings
- Packer & compiler details
- Compile time, Entry Point
- Import Address Hash,
- Function calls & API’s etc.

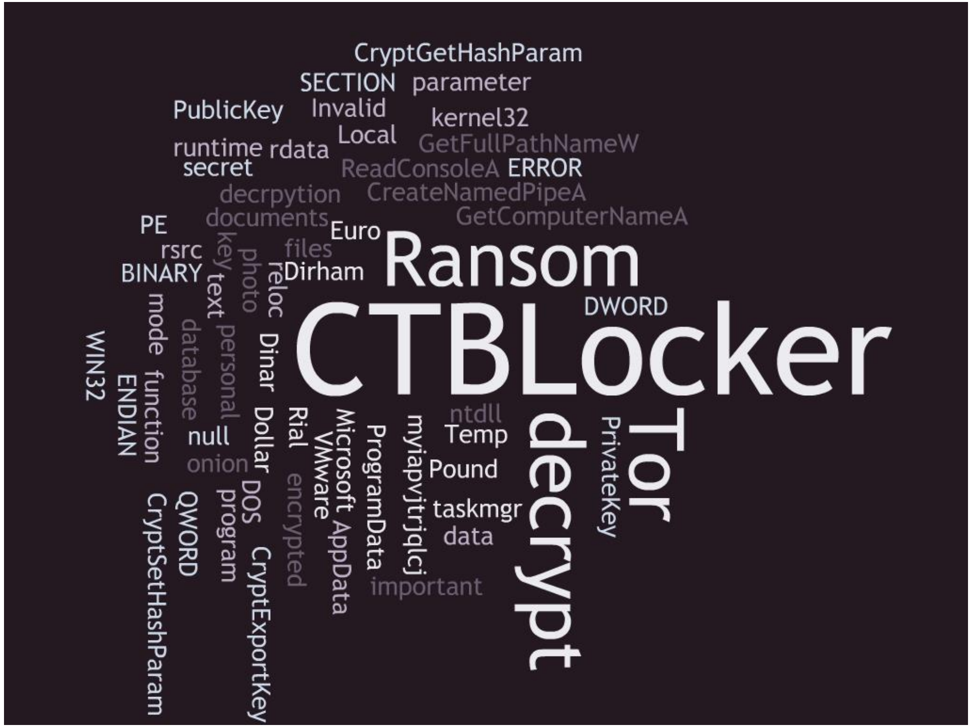
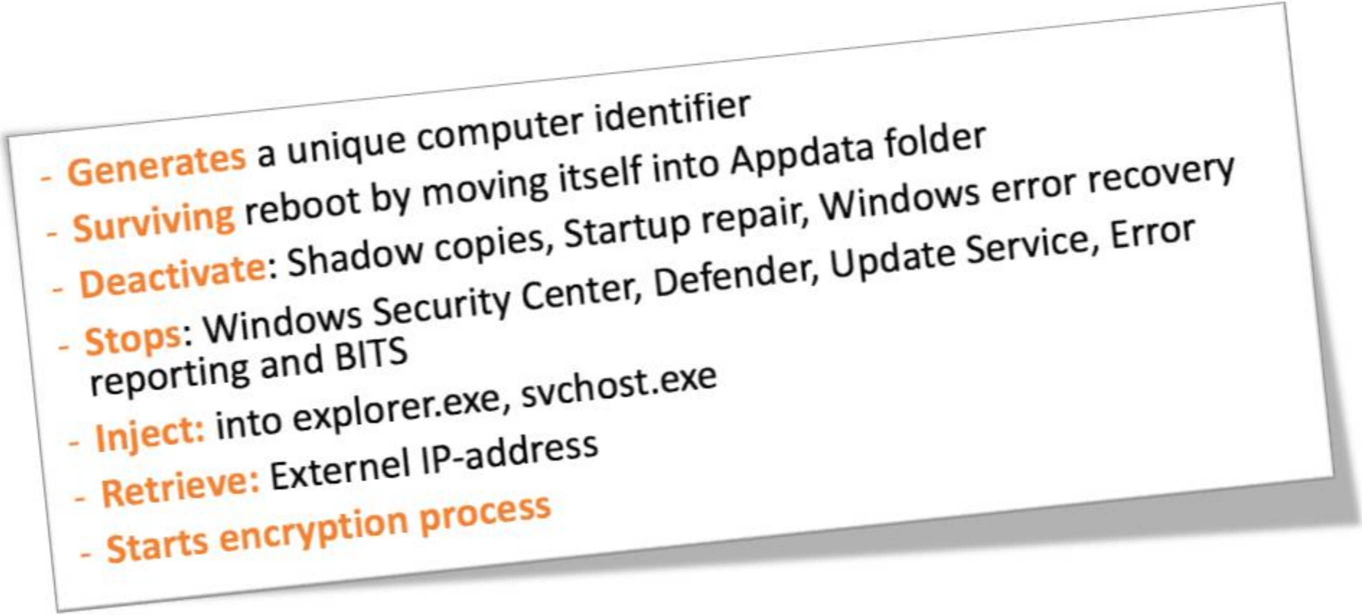


Image: http://www.abcya.com/word_clouds.htm

Extracting Behavioral Features

Ransomware: CTB-Locker (post-execution)

- File system, registry and network changes actions it begins encrypting files

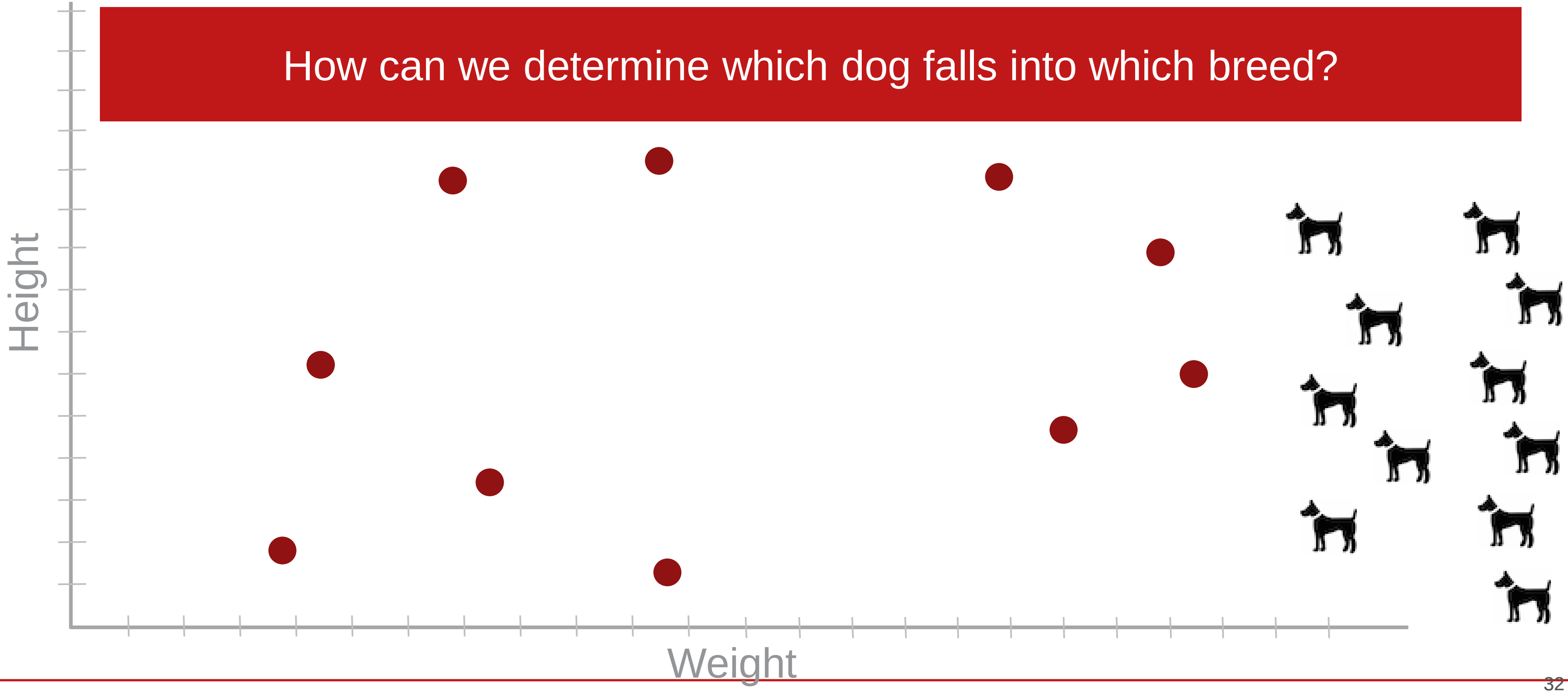


- **Generates** a unique computer identifier
- **Surviving** reboot by moving itself into Appdata folder
- **Deactivate**: Shadow copies, Startup repair, Windows error recovery
- **Stops**: Windows Security Center, Defender, Update Service, Error reporting and BITS
- **Inject:** into explorer.exe, svchost.exe
- **Retrieve:** Externel IP-address
- **Starts encryption process**

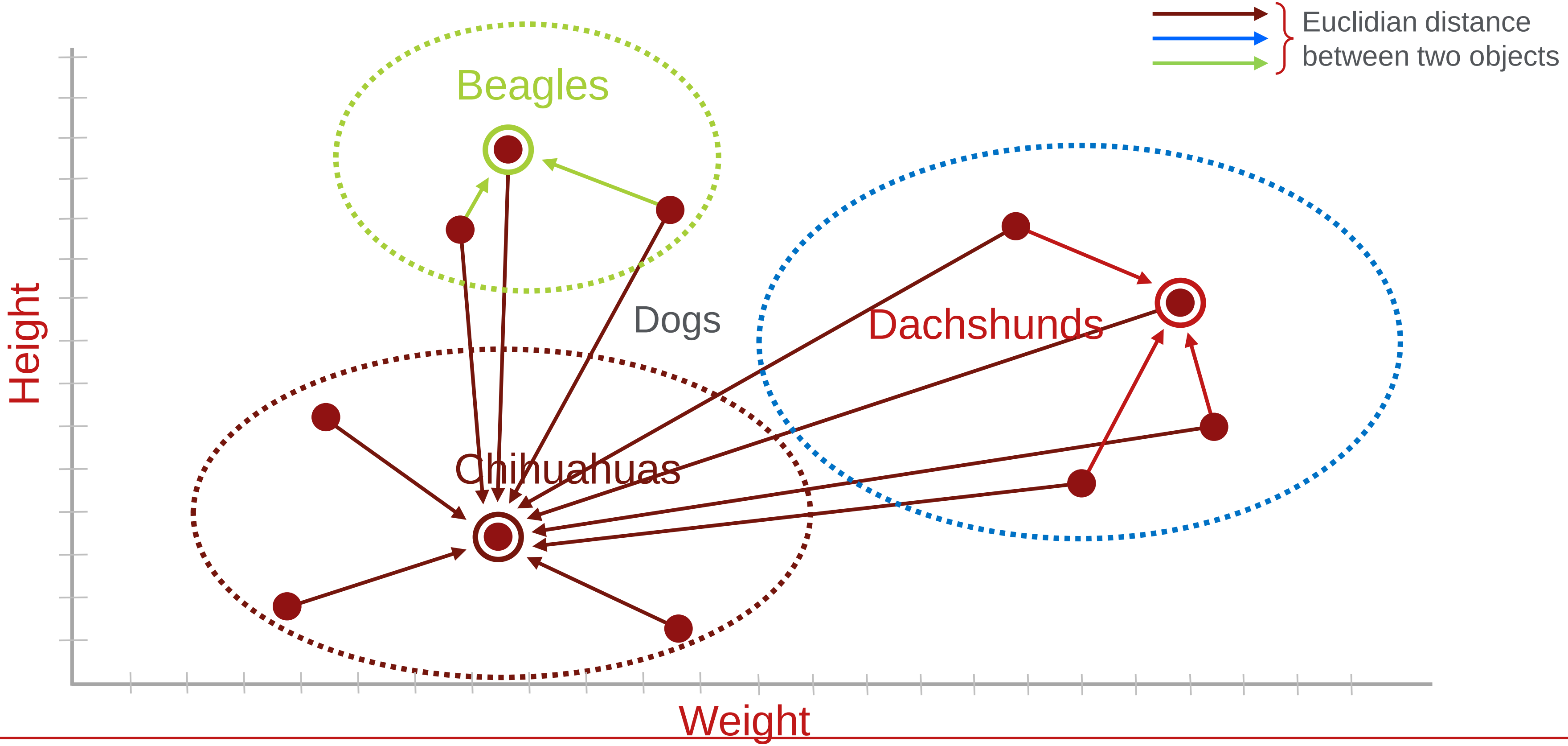
Unsupervised Machine Learning

- Lets start with an example

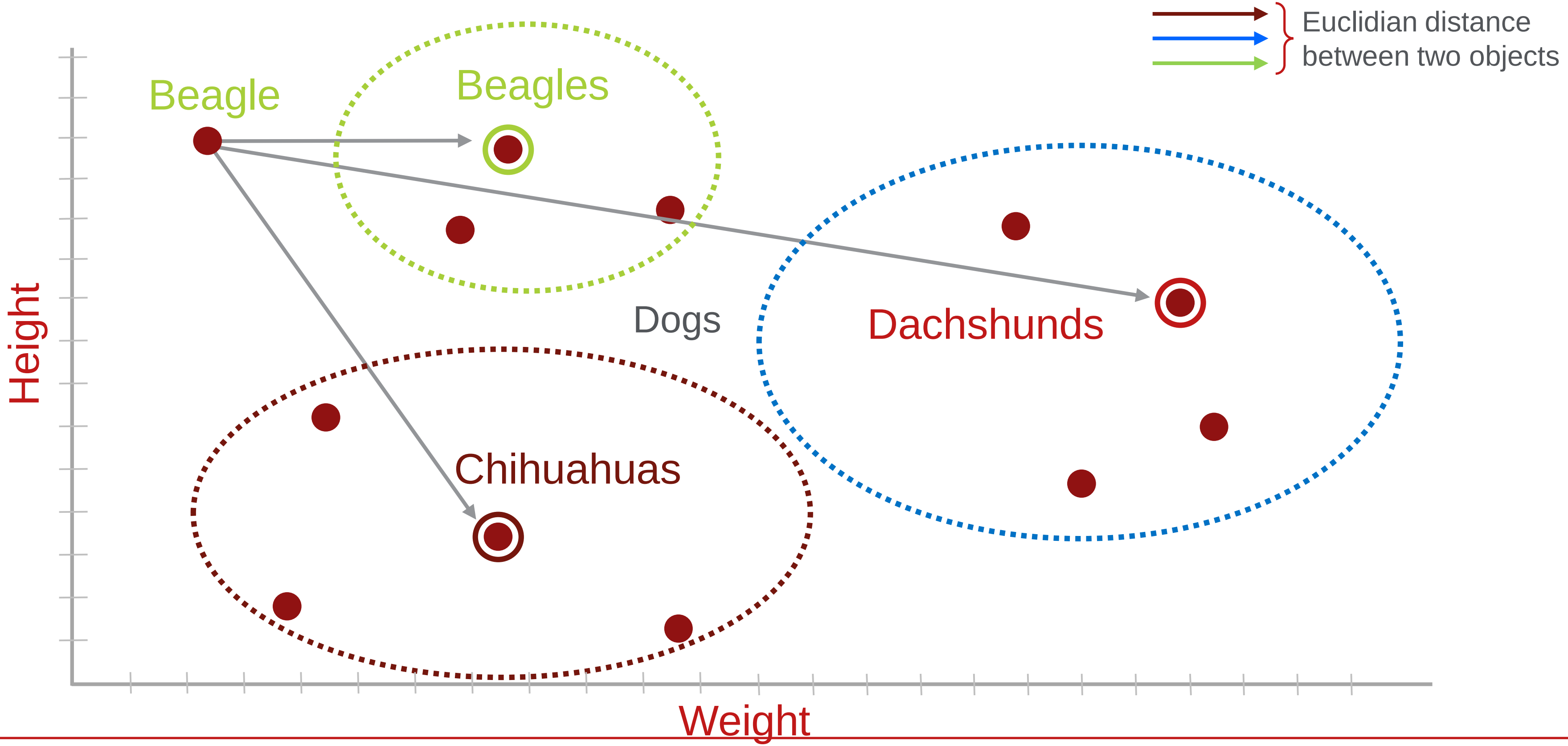
How can we determine which dog falls into which breed?



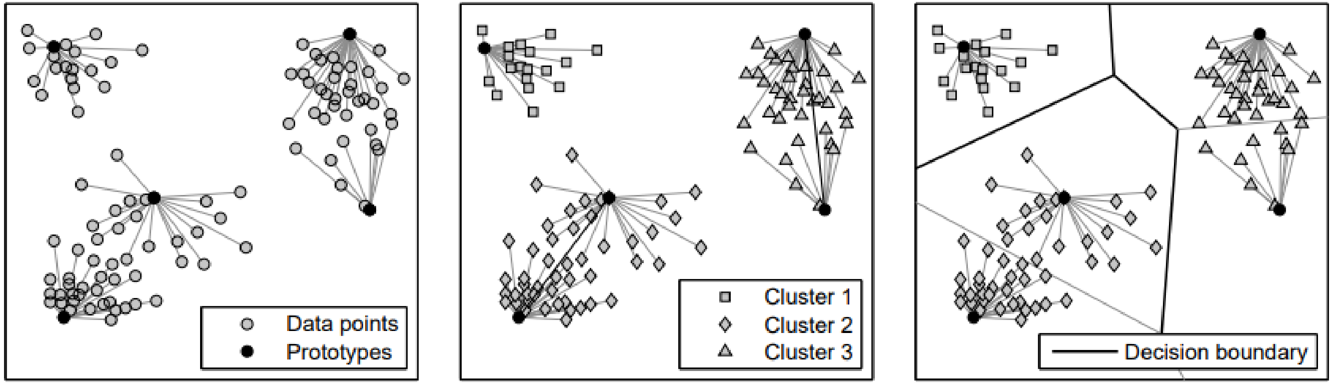
Similarity: Prototype Based Clustering



Similarity: Classification Based On Clustering



Clustering

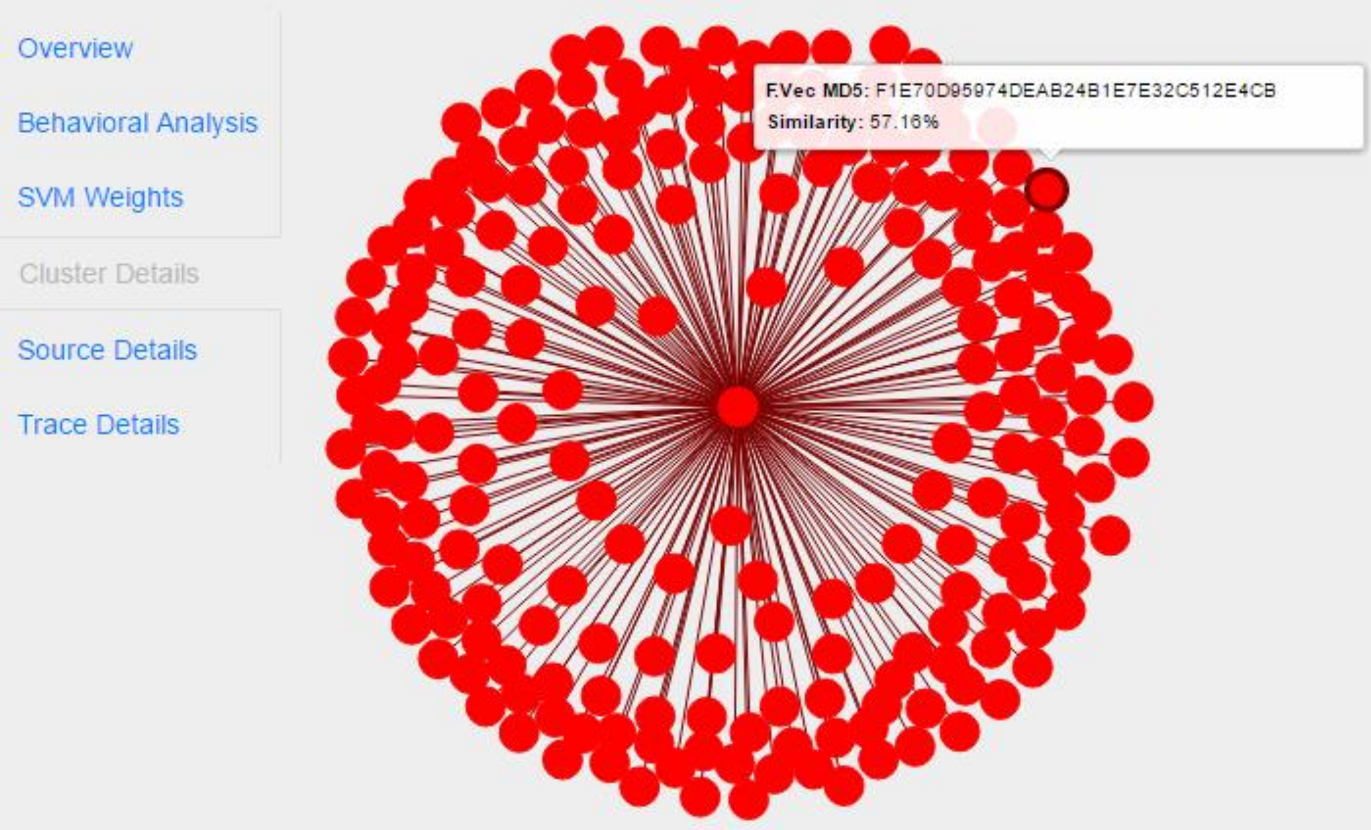


(a) Prototypes (b) Clustering (c) Classification

Figure 4: Behavior analysis using prototypes: (a) prototypes of data, (b) clustering using prototypes, and (c) classification using prototypes. Black lines in Figure 4(b) indicate prototypes joined by linkage clustering. Black lines in Figure 4(c) represent the class decision boundary.

Classification With Real Protect

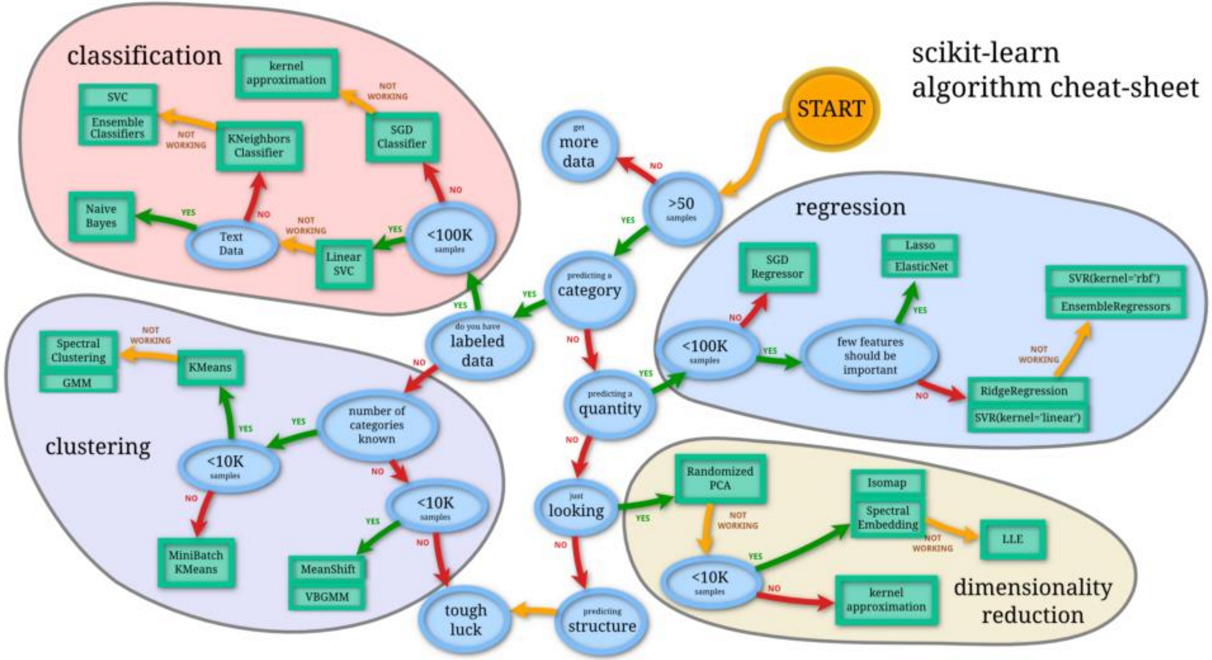
Graphic representation of clusters with samples which are similar



F.Vec MD5: F1E70D95974DEAB24B1E7E32C512E4CB

Similarity: 57.16%

Modeling Machine Learning Classifier

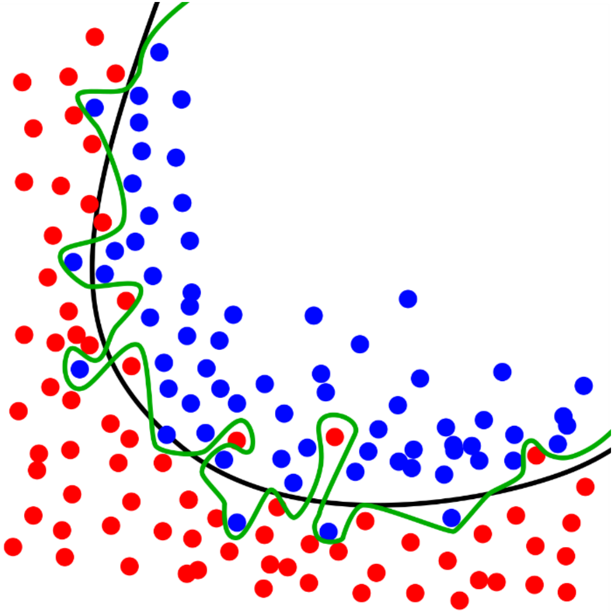

scikit-learn
algorithm cheat-sheet
START
>50 samples
get more data
predicting a category
do you have labeled data
predicting a quantity
just looking
predicting structure
tough luck
classification
<100K samples
Linear SVC
Text Data
Naive Bayes
KNeighbors Classifier
SVC
Ensemble Classifiers
SGD Classifier
kernel approximation
clustering
number of categories known
<10K samples
KMeans
Spectral Clustering
GMM
MiniBatch KMeans
<10K samples
MeanShift
VBGMM
regression
<100K samples
SGD Regressor
few features should be important
Lasso
ElasticNet
RidgeRegression
SVR(kernel='linear')
SVR(kernel='rbf')
EnsembleRegressors
dimensionality reduction
Randomized PCA
<10K samples
Isomap
Spectral Embedding
LLE
kernel approximation
YES
NO
NOT WORKING

Overfitting

- Training your model properly is important

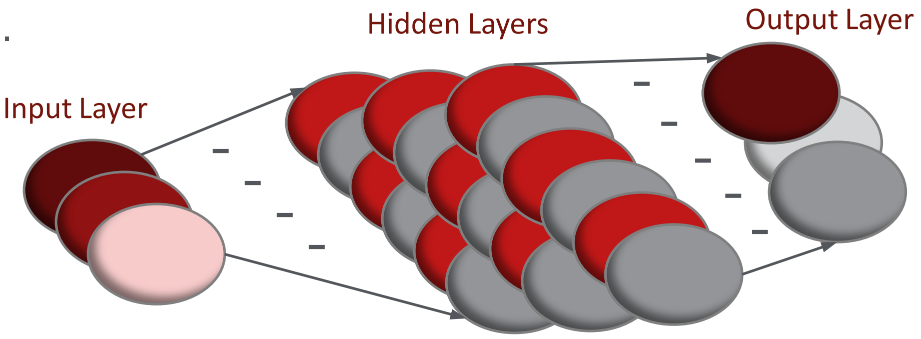
- Analogy- If you keep giving a student last week's test, he will get A+

- May perform horribly if you give him a new test



Modeling a Machine Learning Classifier

- **Input Data**
  - Executables, compiled code, documents etc..
- **Feature Engineering**
  - N-grams, entropy of sections etc..
- **Labels**
  - Is malicious or clean?
  - Belongs to a certain family of malware
  - Capabilities (keyloggers, backdoors..)
- **Model**
  - Assigns a sample to an output class
  - Support vector machines, Naïve Bayes, Random forests, neural networks etc



Attacking Machine Learning Defenses

Exploratory: Obfuscate to Evade Detection

HypervisorDetection

SystemTampering

BranchConditionModification

Encryption

JunkInsertion

InstructionSubstitution

OSFingerprinting

RegisterRrenaming

InstructionReordering

Compression

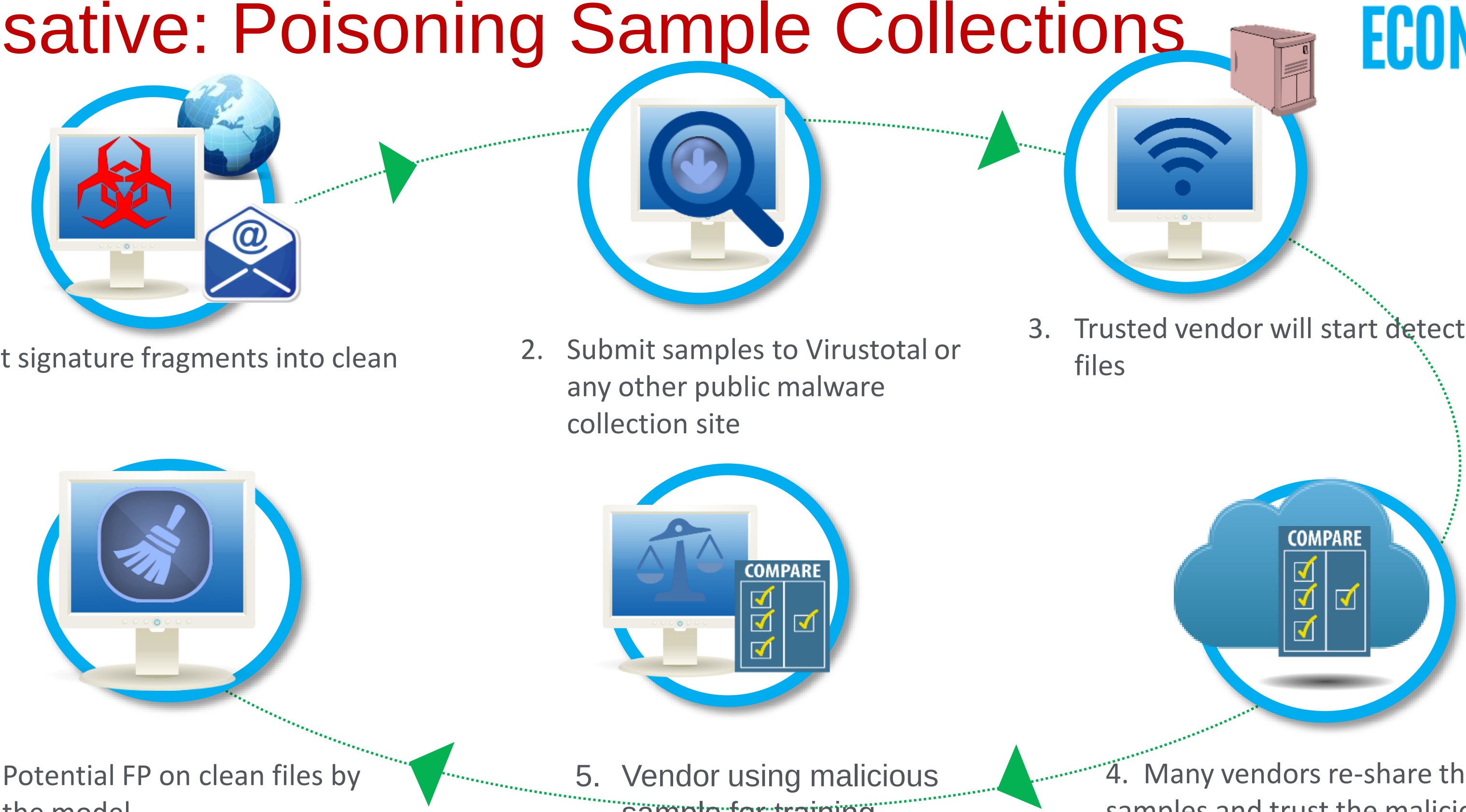
LatentExecution

NopInsertion

InteractionBased

BasicBlockReordering

Causative: Poisoning Sample Collections



1. Insert signature fragments into clean files

2. Submit samples to Virustotal or any other public malware collection site

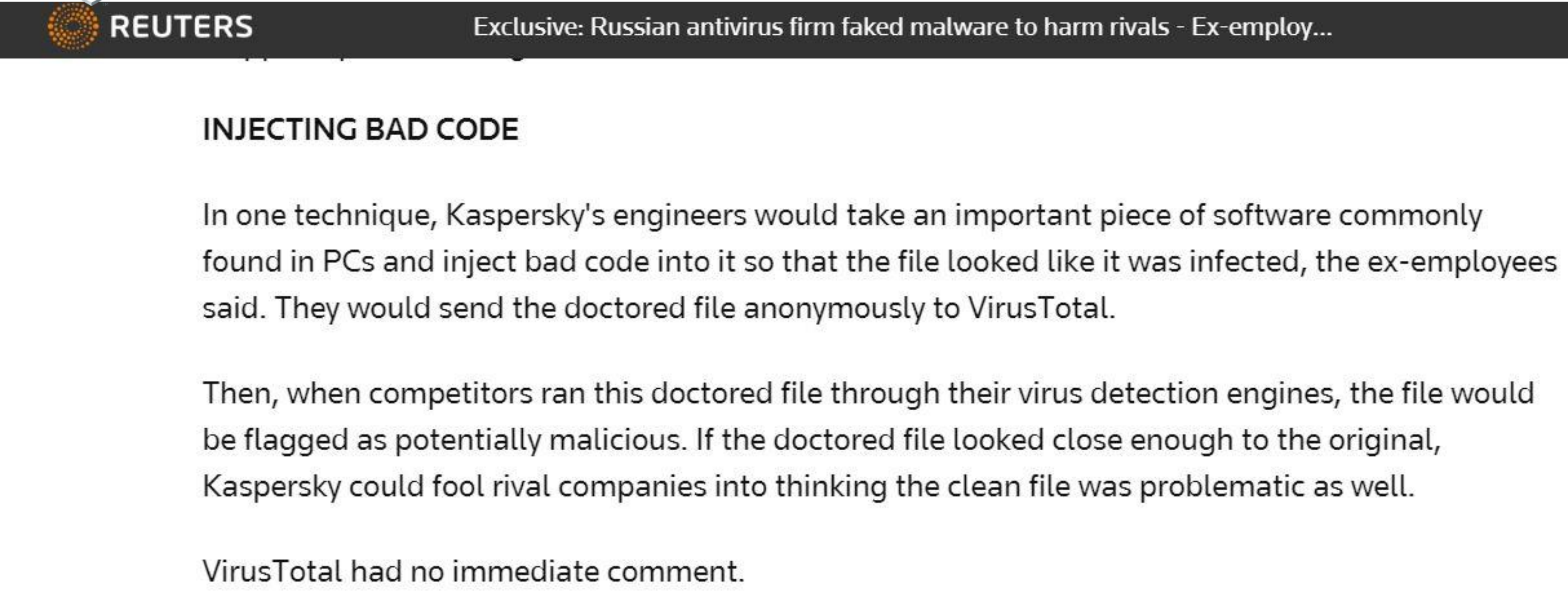
3. Trusted vendor will start detecting those files

4. Many vendors re-share the samples and trust the malicious classification

5. Vendor using malicious sample for training models

6. Potential FP on clean files by the model

Causative: Poisoning Sample Collections



REUTERS Exclusive: Russian antivirus firm faked malware to harm rivals - Ex-employ...

INJECTING BAD CODE

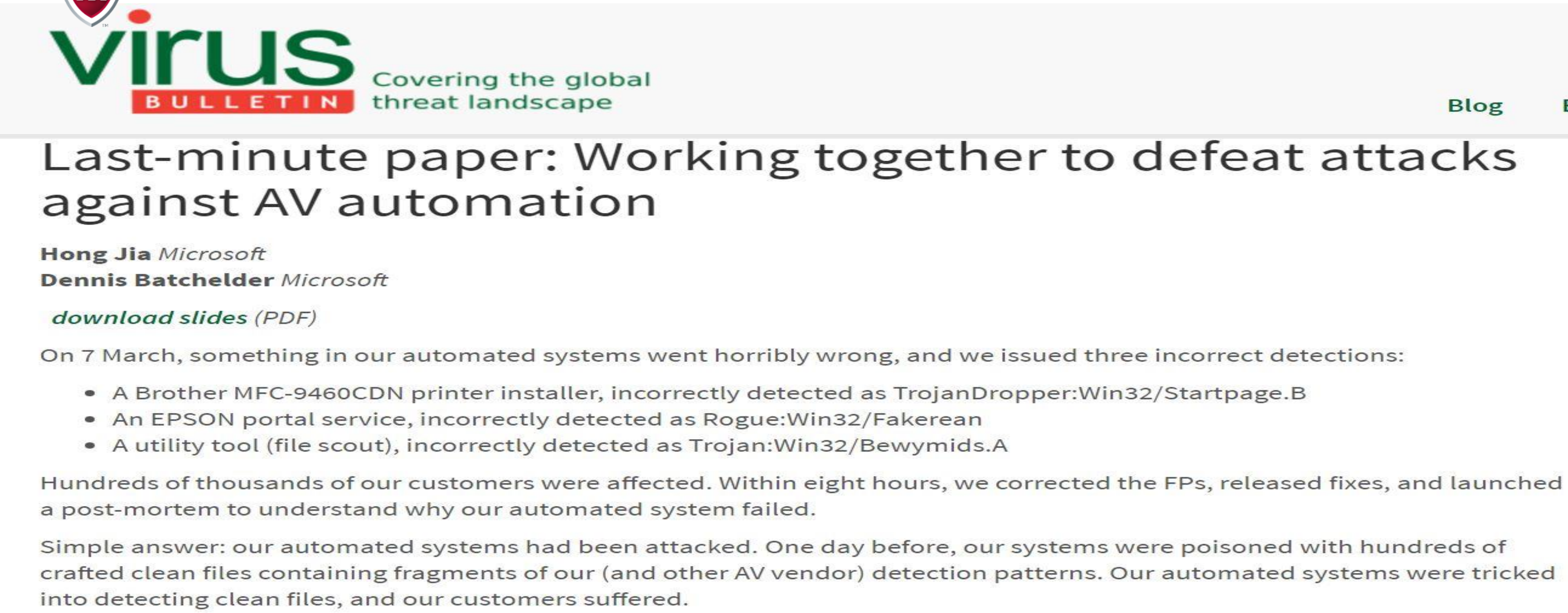
In one technique, Kaspersky's engineers would take an important piece of software commonly found in PCs and inject bad code into it so that the file looked like it was infected, the ex-employees said. They would send the doctored file anonymously to VirusTotal.

Then, when competitors ran this doctored file through their virus detection engines, the file would be flagged as potentially malicious. If the doctored file looked close enough to the original, Kaspersky could fool rival companies into thinking the clean file was problematic as well.

VirusTotal had no immediate comment.

Source: Reuters

Causative: Poisoning Sample Collections



Last-minute paper: Working together to defeat attacks against AV automation

Hong Jia *Microsoft*
Dennis Batchelder *Microsoft*

download slides (PDF)

On 7 March, something in our automated systems went horribly wrong, and we issued three incorrect detections:

- A Brother MFC-9460CDN printer installer, incorrectly detected as TrojanDropper:Win32/Startpage.B
- An EPSON portal service, incorrectly detected as Rogue:Win32/Fakerean
- A utility tool (file scout), incorrectly detected as Trojan:Win32/Bewymids.A

Hundreds of thousands of our customers were affected. Within eight hours, we corrected the FPs, released fixes, and launched a post-mortem to understand why our automated system failed.

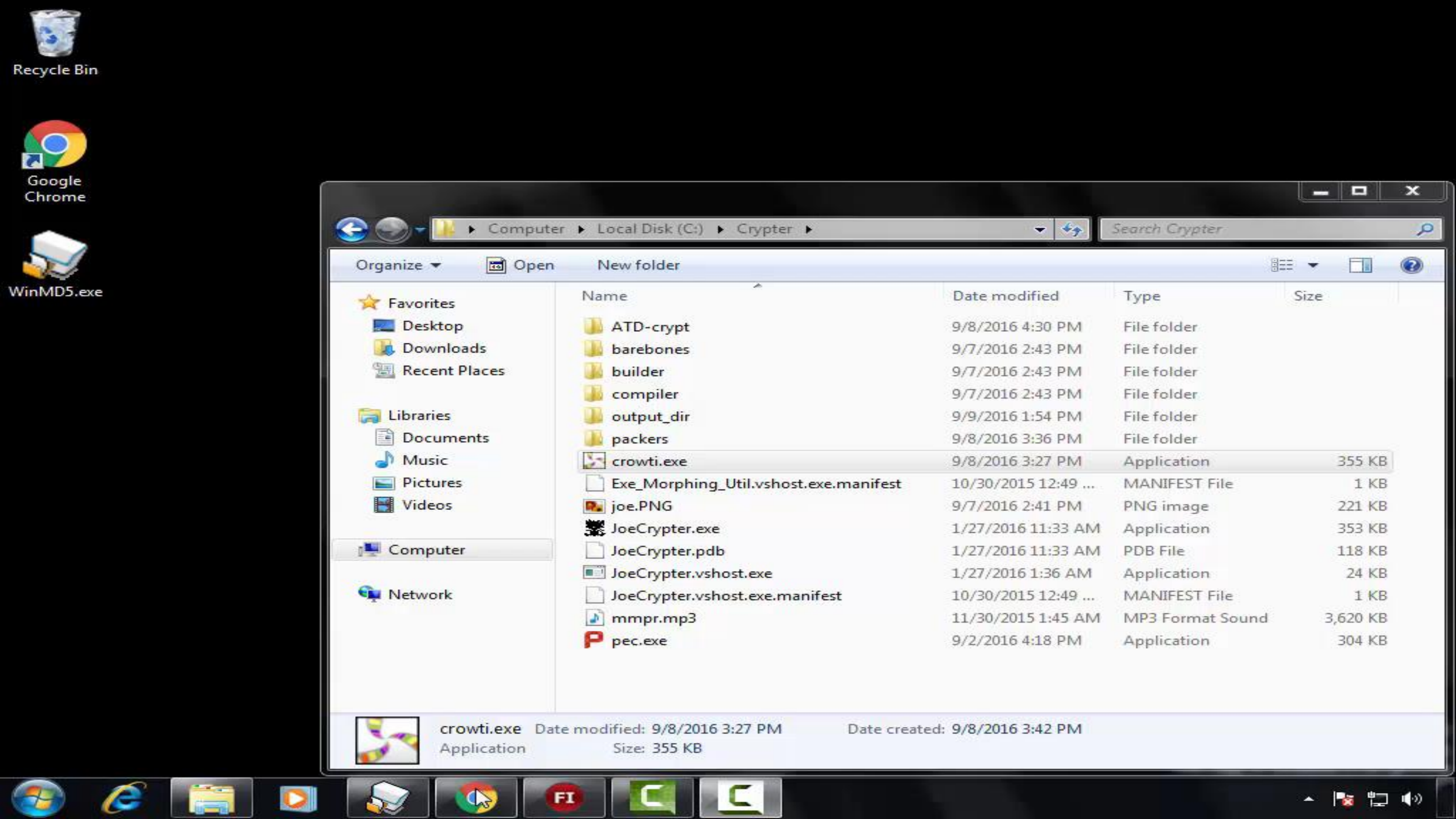
Simple answer: our automated systems had been attacked. One day before, our systems were poisoned with hundreds of crafted clean files containing fragments of our (and other AV vendor) detection patterns. Our automated systems were tricked into detecting clean files, and our customers suffered.

Source: Virus Bulletin

Defenses Against Machine Learning Attacks

- ✓ Exploratory attack
  - ✓ Training data: Prevent the attacker from knowing training data
  - ✓ Feature Selection: Harden classifiers against attack by using multiple features

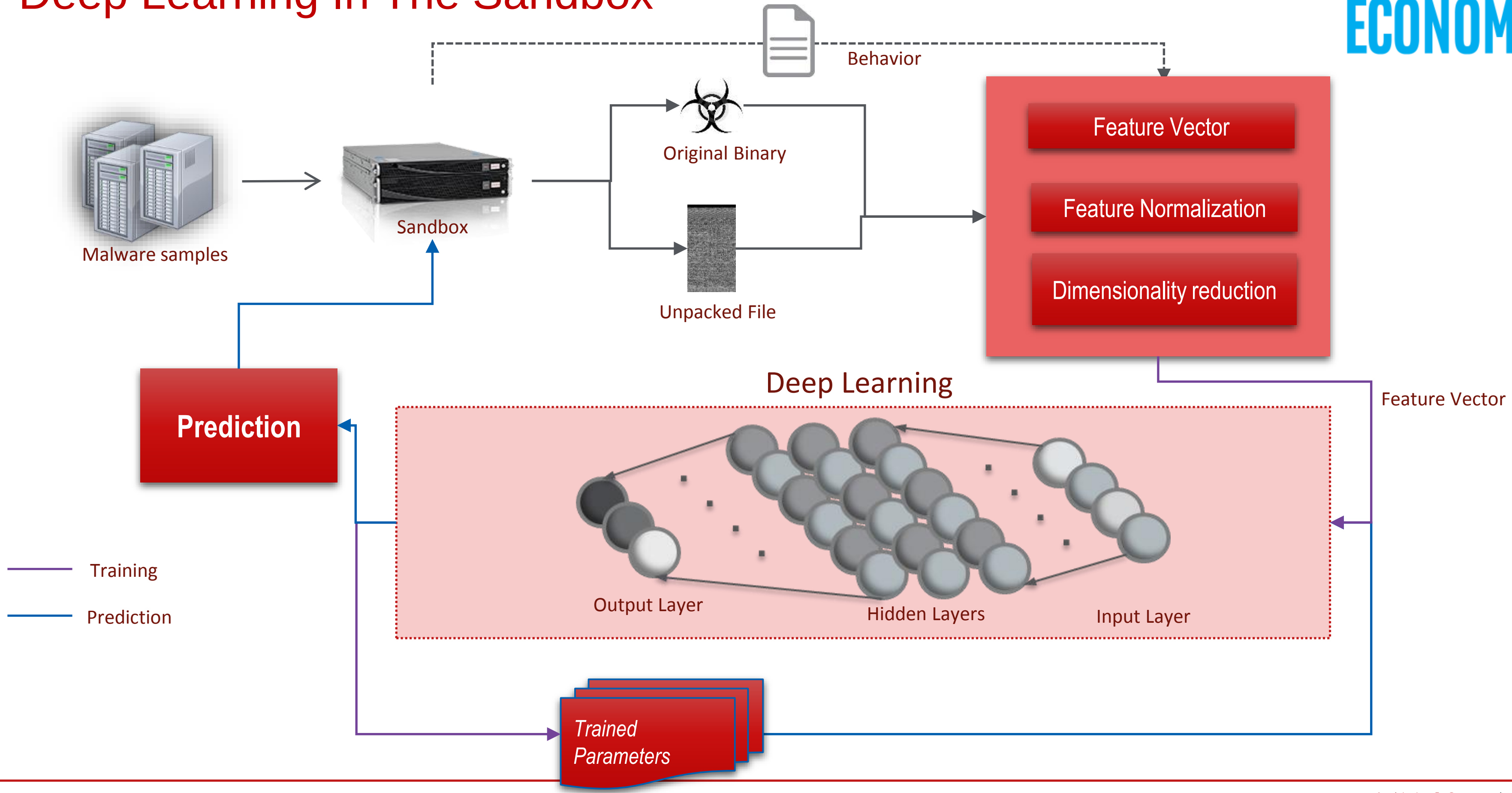
- ✓ Causative attack: Attacker has some degree of control over the training data. Learning should be resilient to poisoning attacks
  - ✓ Do empirical analysis of training instances to make it more resilient
  - ✓ Human in loop approach



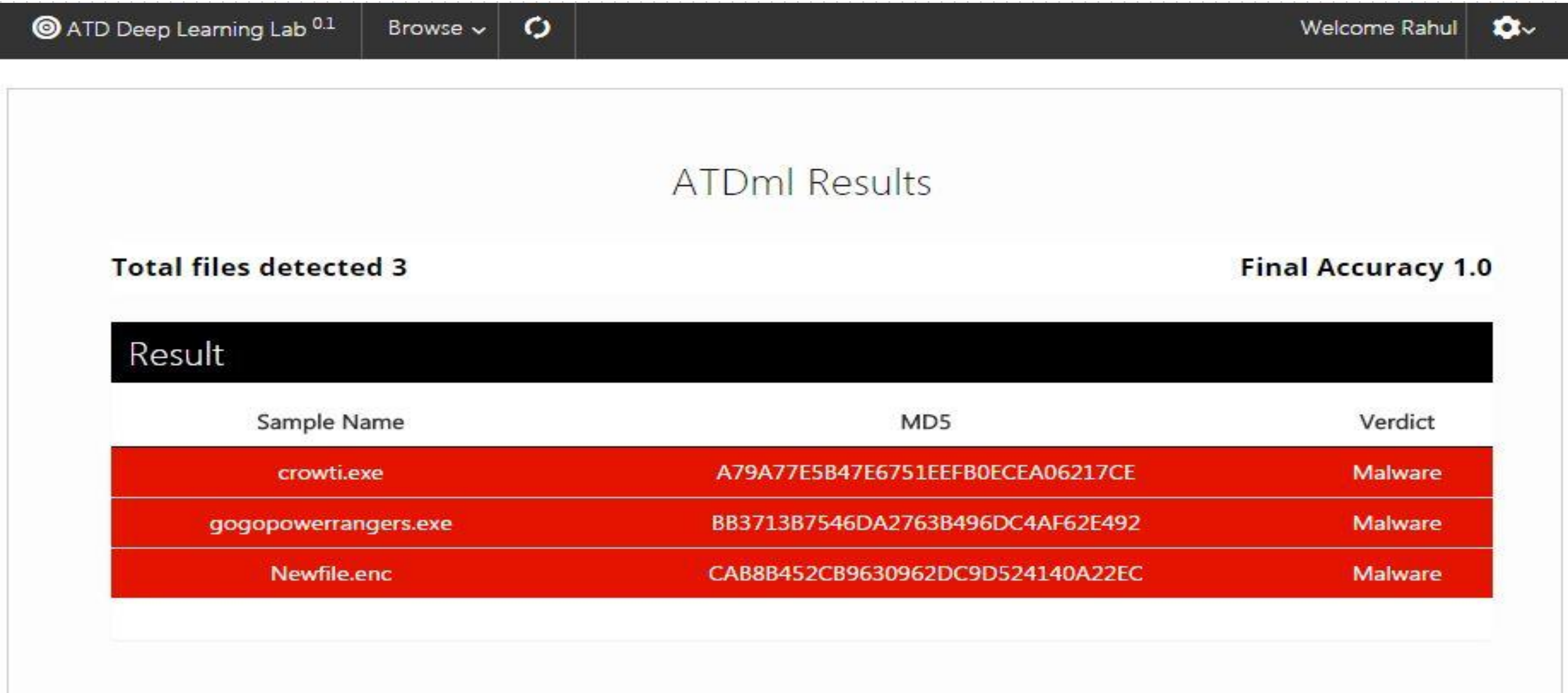
Name	Date modified	Type	Size
ATD-crypt	9/8/2016 4:30 PM	File folder	
barebones	9/7/2016 2:43 PM	File folder	
builder	9/7/2016 2:43 PM	File folder	
compiler	9/7/2016 2:43 PM	File folder	
output_dir	9/9/2016 1:54 PM	File folder	
packers	9/8/2016 3:36 PM	File folder	
crowti.exe	9/8/2016 3:27 PM	Application	355 KB
Exe_Morphing_Util.vshost.exe.manifest	10/30/2015 12:49 ...	MANIFEST File	1 KB
joe.PNG	9/7/2016 2:41 PM	PNG image	221 KB
JoeCrypter.exe	1/27/2016 11:33 AM	Application	353 KB
JoeCrypter.pdb	1/27/2016 11:33 AM	PDB File	118 KB
JoeCrypter.vshost.exe	1/27/2016 1:36 AM	Application	24 KB
JoeCrypter.vshost.exe.manifest	10/30/2015 12:49 ...	MANIFEST File	1 KB
mmpr.mp3	11/30/2015 1:45 AM	MP3 Format Sound	3,620 KB
pec.exe	9/2/2016 4:18 PM	Application	304 KB

crowti.exe Date modified: 9/8/2016 3:27 PM Date created: 9/8/2016 3:42 PM
Application Size: 355 KB

Deep Learning In The Sandbox



ATDml Detection



ATDml Results

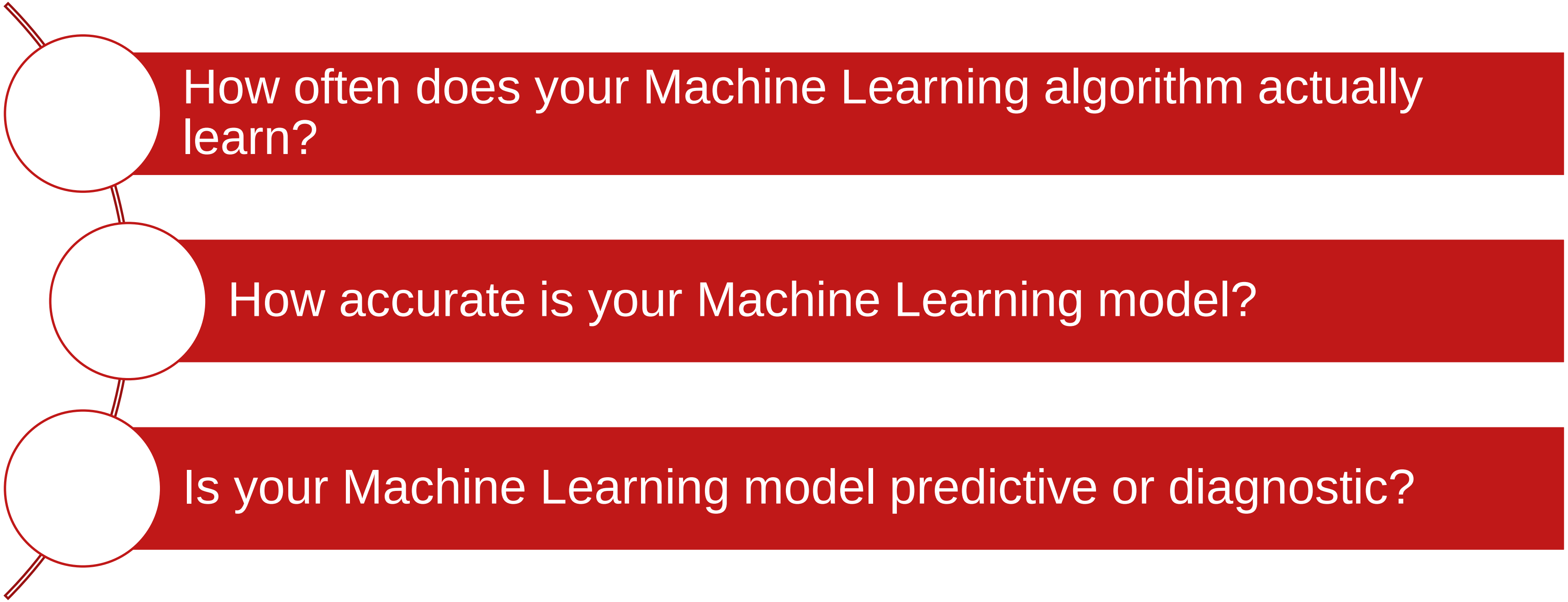
Total files detected 3

Final Accuracy 1.0

Result

Sample Name	MD5	Verdict
crowti.exe	A79A77E5B47E6751EEFB0ECEA06217CE	Malware
gogopowerrangers.exe	BB3713B7546DA2763B496DC4AF62E492	Malware
Newfile.enc	CAB8B452CB9630962DC9D524140A22EC	Malware

When to call BS on Machine Learning



How often does your Machine Learning algorithm actually learn?

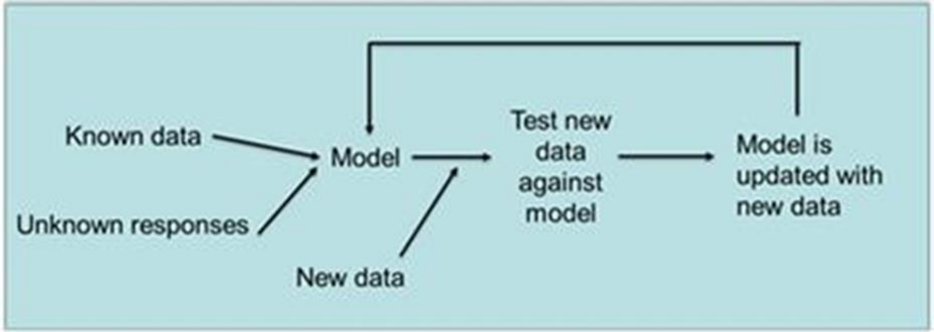
How accurate is your Machine Learning model?

Is your Machine Learning model predictive or diagnostic?

How often does your Machine Learning algorithm actually learn?

Why is it important:

- Learning is fundamental to ML. It means that as new information comes in, the model is updating itself (typically automatically) to be more relevant (and perhaps, more accurate) than it was to begin with. Developing a ML model, putting it out to the market, and then never updating it may increase false positives and false negatives. It may not consider all the new processes a customer (or the vendor) have that may ‘decay” the model, making it potentially irrelevant. A simple pictorial that shows a continuously learning model:



How often does your Machine Learning algorithm actually learn?

Good Answer:

- The algorithm learns at the rate that was determined at the time of model development and is updated periodically and applied to new signals. For example: “For this particular model, it is updated continuously.” “For other models, they are updated based on statistically significant changes to inputs to the model (e.g., the False Positive and False Negative rates, changes to filters, changes to down-select/down sampling, changes to customer or vendor processes, etc.).”

How often does your Machine Learning algorithm actually learn?

Bad Answer:

- “Routinely” – press the responder to drill down to an answer without “leading the witness”, such as “Routinely? How often?” “Is it different or the same for different models?” “What signals does your company use to update the model?” Or even the very basic “How do you know the model is working?”

How accurate is your Machine Learning model?

Why is it important:

- Accuracy is the measure of error. So if your ML model is not as accurate as it could be, it means it has more error and giving you incorrect decisions. Error is measured differently for different models. False positives (FP), false negatives (FN), true positives (TP), and true negatives (TN) are NOT the only way of measuring error. In fact, you can have these ROC (Received Operating Curve) measurements of FP, FN, TP, and TN be very good BUT your error rate, measured in “R^2” can be bad, misleading the data scientist. So, sometimes, you must do a few accuracy checks to find that “sweet spot” that meets the customer need and the math.
- Verification determines whether you have done the model right (mathematically). Validation determines whether you have done the right model (for the data set, for the customer requirements). While these overlap somewhat, the idea is that you want to try a few models in order to optimize your model, not just one and run with it. Many models do not scale with data volume or data variety.

How accurate is your Machine Learning model?

Good answer:

- The specific ML model for this purpose has a 95% accuracy. We also measure other forms of accuracy, such as Mean Square Root Error, Generalized R^2, and additional ROC metrics such as Recall, F1 Score, and Matthew’s Correlation Coefficient.

How accurate is your Machine Learning model?

Bad answer:

- 100%. (Nothing is ever 100%!!.)
- We haven’t had a complaint about inaccuracies.

Is your Machine Learning model predictive or diagnostic?

Why is it important:

- ML can be applied to data that is descriptive (what happened?), diagnostic (why did it happen?), predictive (what will happen?) and prescriptive (here are the recommendations for avoiding what will happen). You must have descriptive to do diagnostic, and you must have diagnostic to do predictive, and, lastly, you must have predictive to do prescriptive. Currently, in almost all industries (including security), ML is descriptive and/or diagnostic. It relates to what has happened in the past. A signal occurs, and the model is applied. With predictive ML, we are predicting what will happen in the future: “we can predict that you have a 95% chance of being hacked tomorrow.”

Is your Machine Learning model predictive or diagnostic?

Good answer:

- In this product X, our ML is diagnostic; that is, we sense a signal and apply the ML model to it in order to tell whether it is adversarial or not. We combine a number of different proprietary mathematical algorithms, including ML models, to arrive at an accurate assessment.

Is your Machine Learning model predictive or diagnostic?

Bad answer:

- It’s definitely predictive! If this answer occurs, dig deeper: How far in the future does it predict? How accurate is it (see Question 2, above)? If the vendor is predictive, ask if they are also prescriptive: Do you provide recommendations on how to avoid this attack from happening in the future?

Unified Defense Architecture

Analytics-driven threat and data defense

Orchestrated

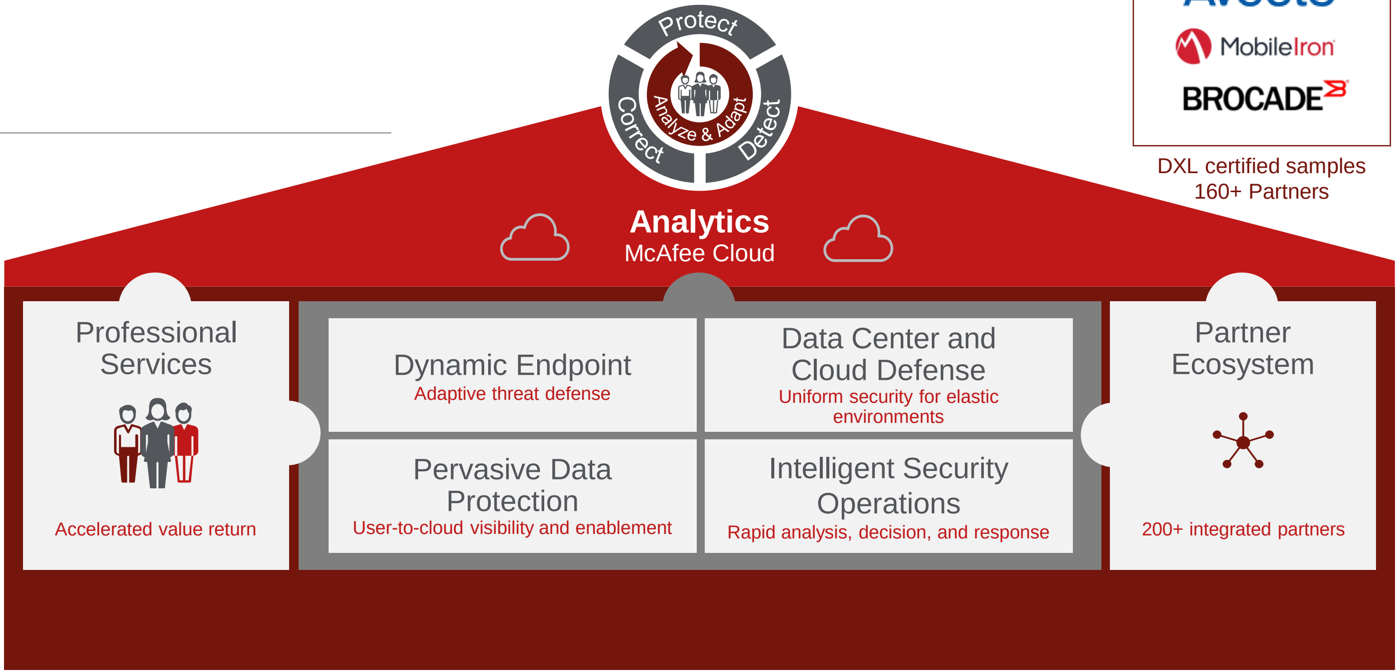
Lifecycle defense

Automated

Solution workflows

Integrated

Core and ecosystem technologies



Protect

Detect

Correct

Analyze & Adapt

Analytics

McAfee Cloud

Professional Services

Accelerated value return

Dynamic Endpoint

Adaptive threat defense

Data Center and Cloud Defense

Uniform security for elastic environments

Pervasive Data Protection

User-to-cloud visibility and enablement

Intelligent Security Operations

Rapid analysis, decision, and response

Partner Ecosystem

200+ integrated partners

FORCEPOINT

Check Point
SOFTWARE TECHNOLOGIES LTD.

RAPID7

Avecto

MobileIron

BROCADE

DXL certified samples
160+ Partners

McAfee™
by Intel