#RSAC

RSAConference2017

San Francisco | February 13 – 17 | Moscone Center

POWER OF OPPORTUNITY

SESSION ID: EXP – T11

Advances in Cloud-Scale Machine Learning for Cyber-Defense

Mark Russinovich
CTO, Microsoft Azure
Microsoft Corporation
@markrussinovich

Intelligence in every software

Cortana Intelligence Suite	SQL Server + R	Microsoft R Server	Hadoop + R	Spark + R	Microsoft CNTK
Azure Machine Learning	R Tools/Python Tools for Visual Studio	Azure Notebooks (JuPyTer)	Cognitive Services	Bot Framework	Cortana
Office 365	HoloLens	Bing	Skype	Xbox 360	Dynamics 365

Microsoft's daily cloud security scale

10s of PBs
of logs

1.3+ billion
Azure Active Directory logons

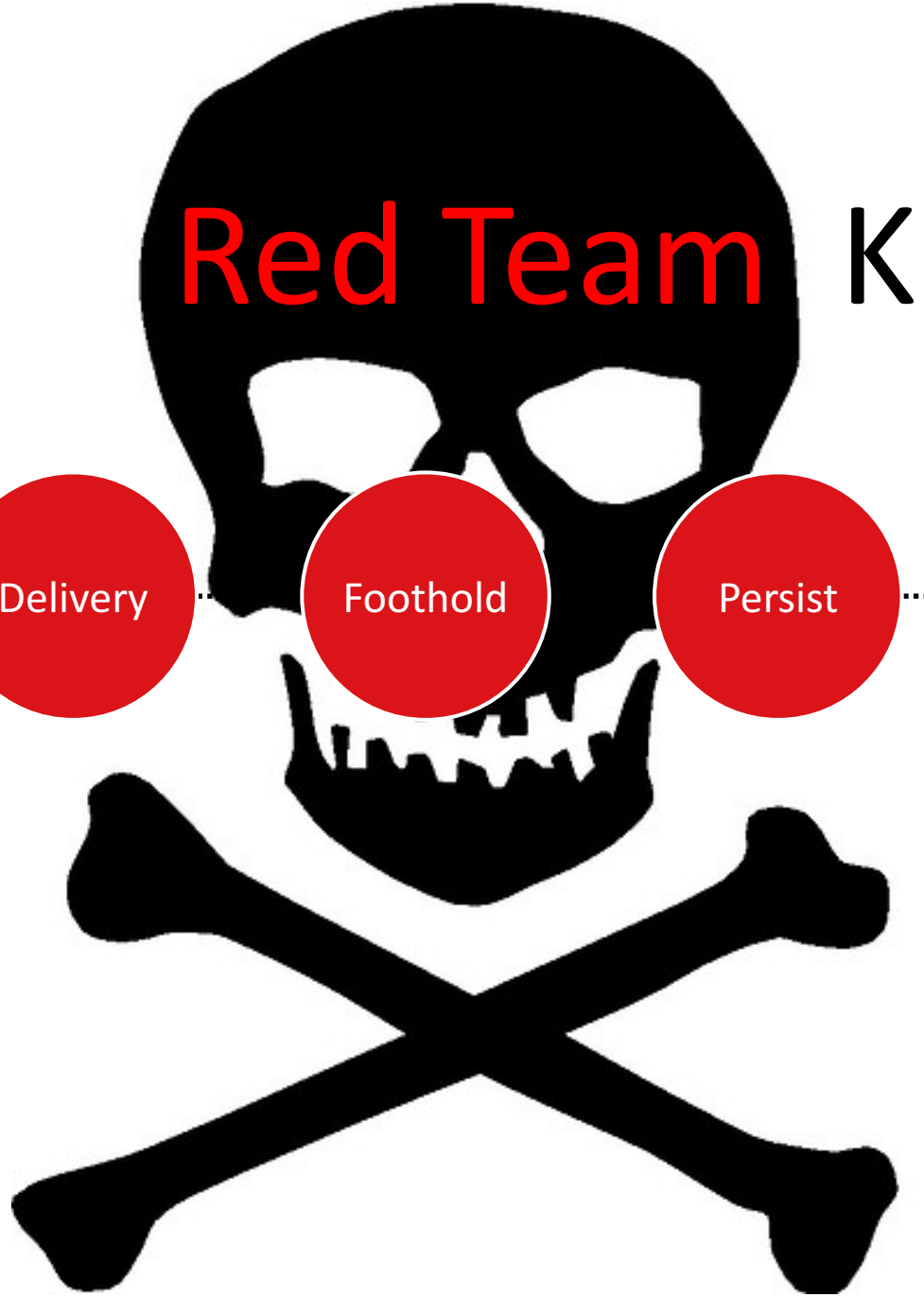
300+ million
active Microsoft Account users

1.5 million
compromise attempts deflected

Detected/ reflected attacks
>10,000
location-detected attacks

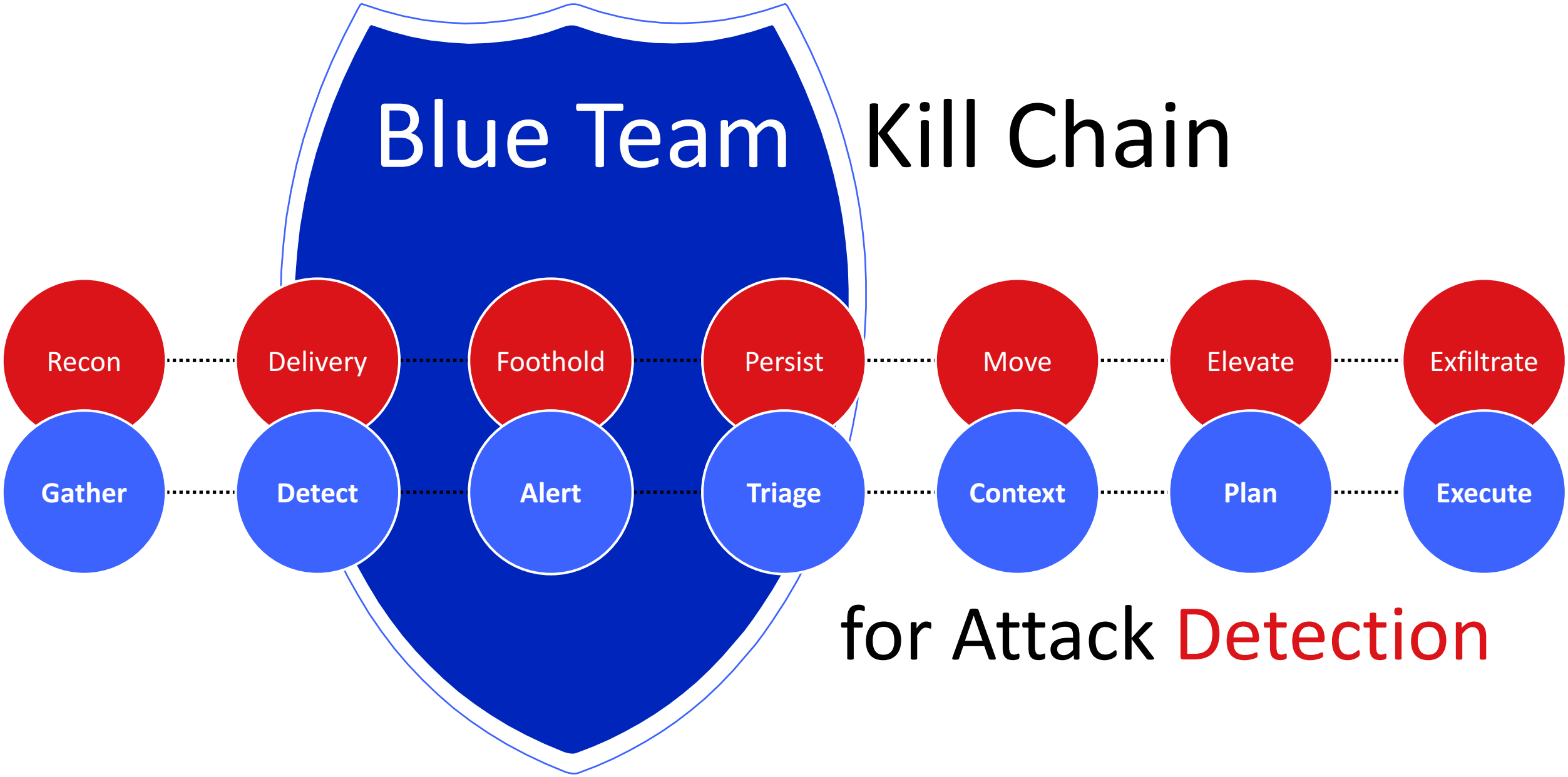
WHAT IS ATTACK DISRUPTION?

Red Team Kill Chain



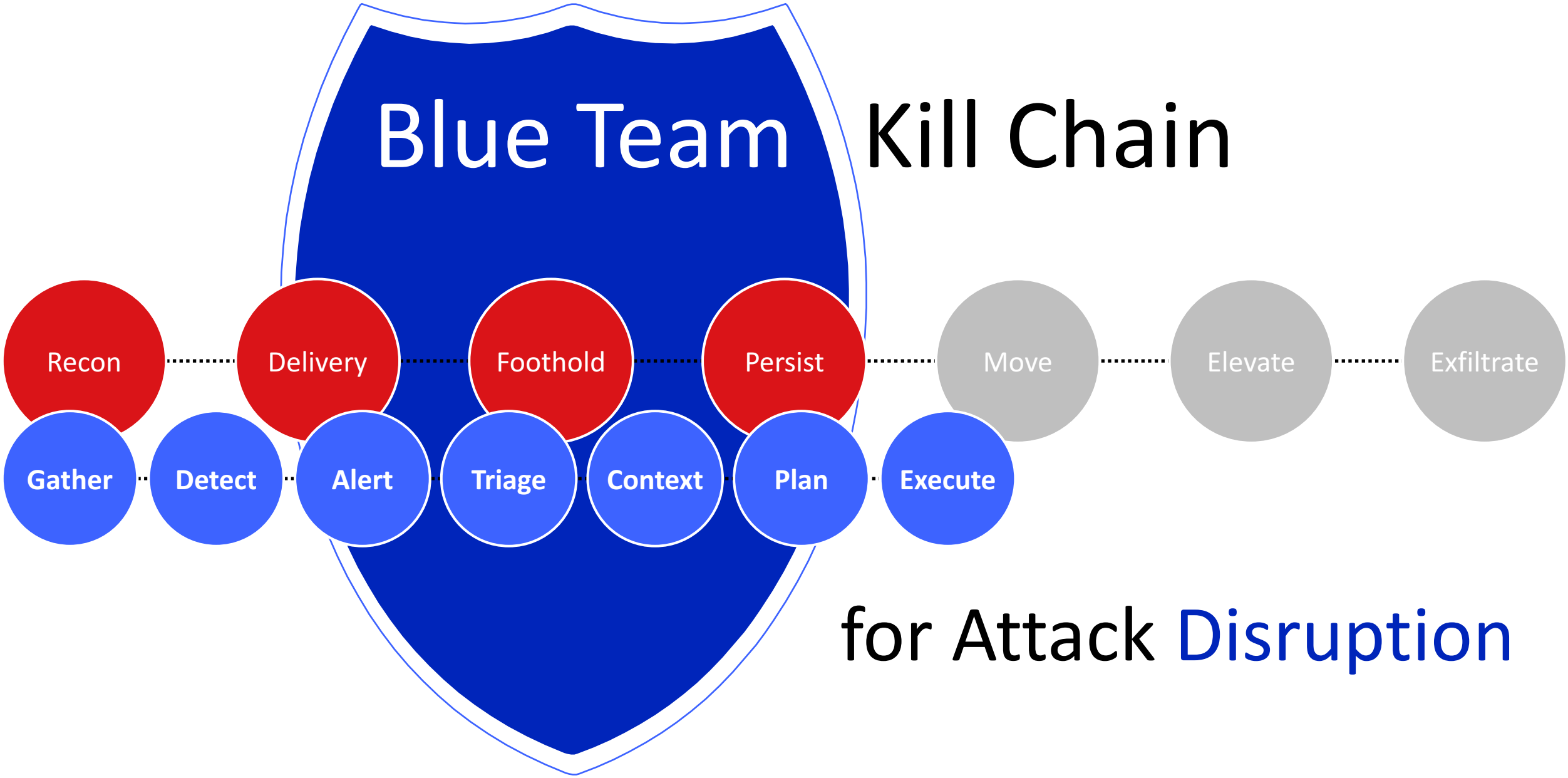
Recon → Delivery → Foothold → Persist → Move → Elevate → Exfiltrate

Blue Team Kill Chain



for Attack Detection

Blue Team Kill Chain



for Attack Disruption

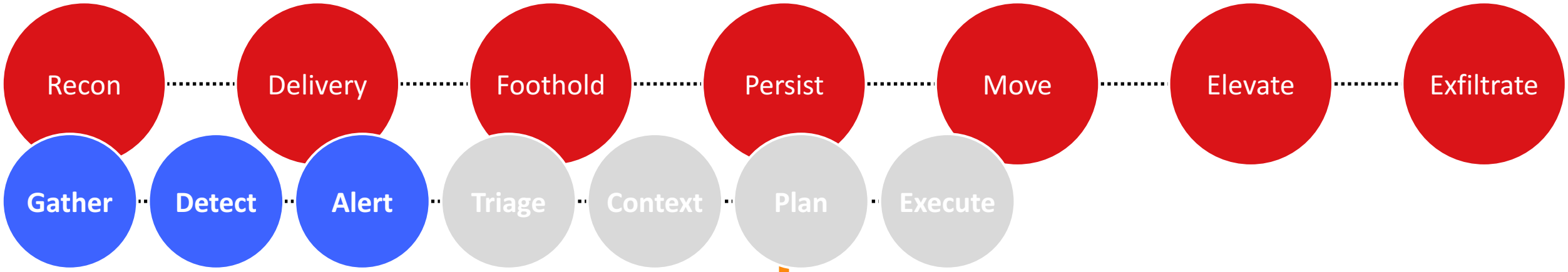
Challenges for Attack Disruption

False Positives

Manual Triage

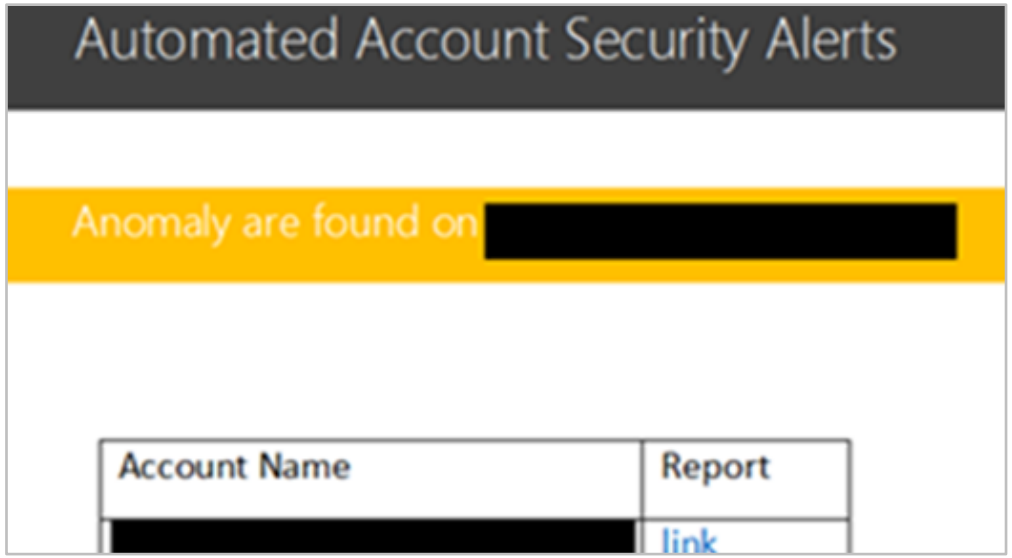
False Positives

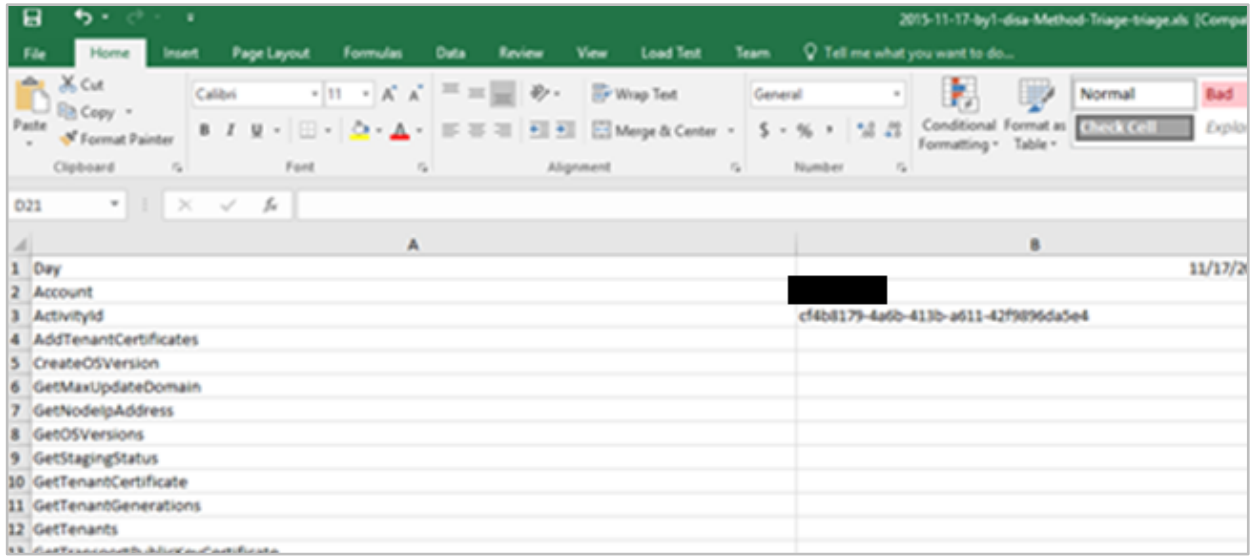
Lose ability to triage

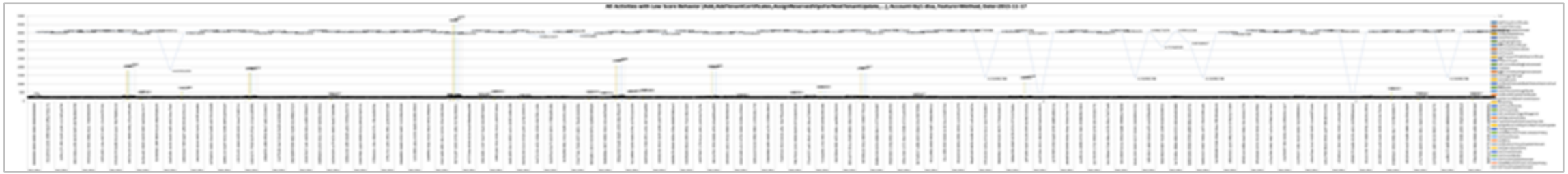


False Positives FACT

You cannot salvage a false positive with just Visualization. You need better solutions.



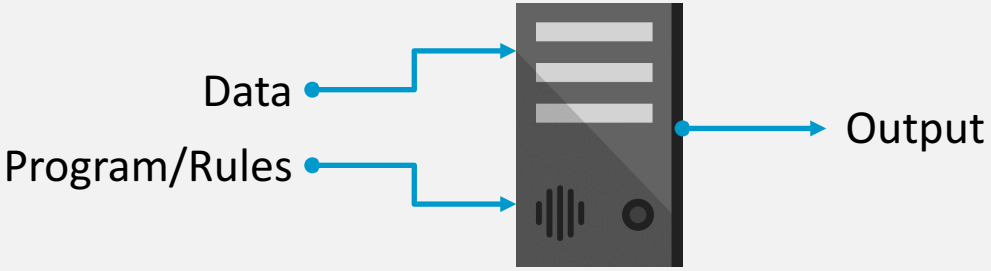




False Positives

Evolution of security detection techniques

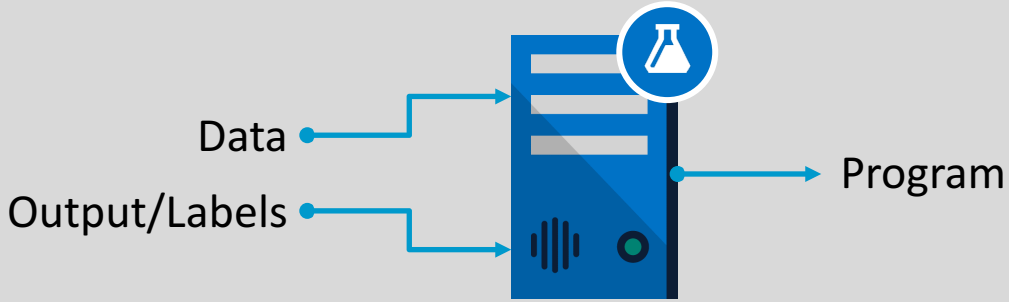
TRADITIONAL PROGRAMMING



Hand-crafted rules by security professionals

Con: Rules are static, and don't change with changes in environment => False Positives!

MACHINE LEARNING

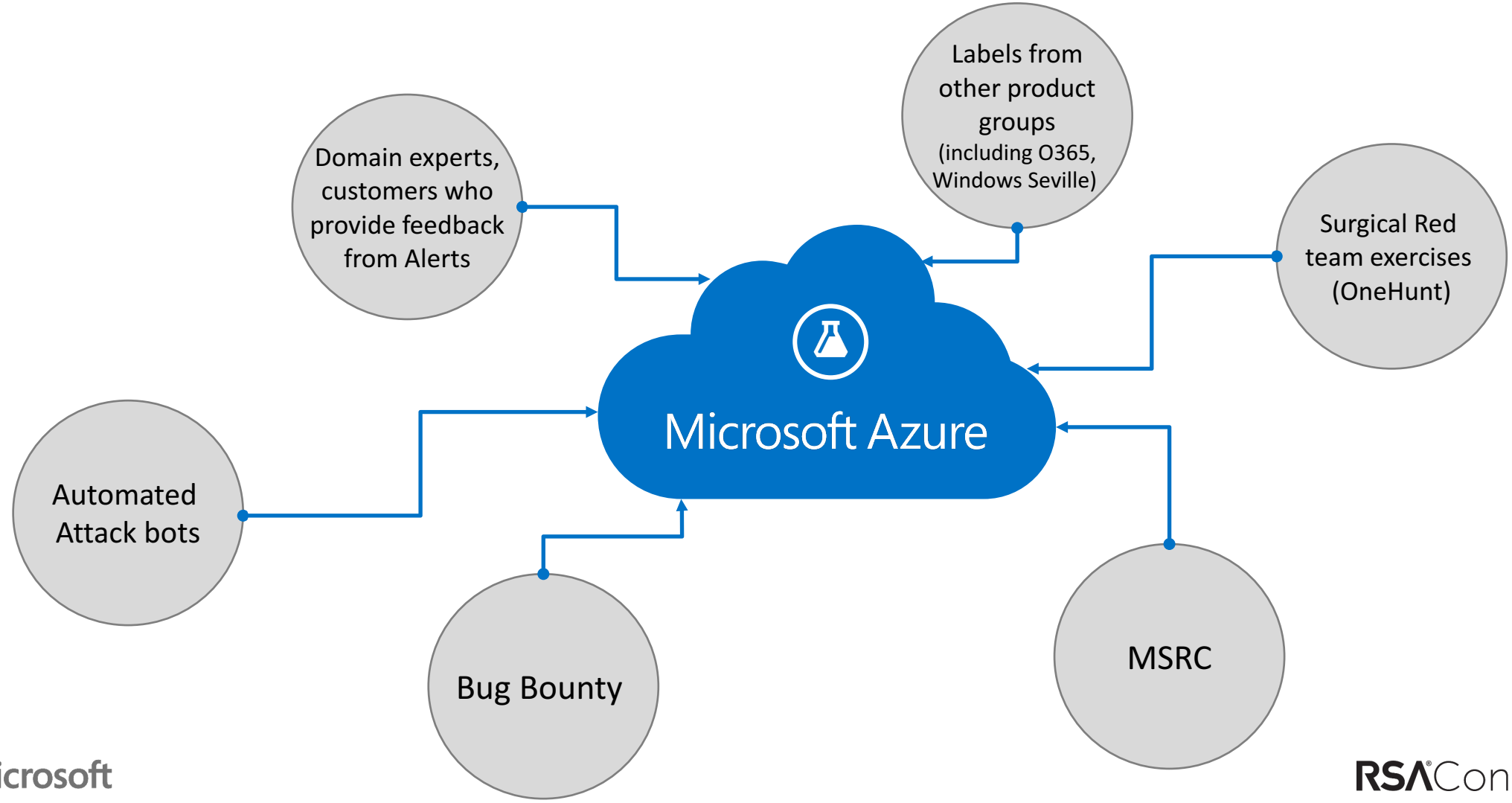


System adapts to changes in environment as new data is provided, and re-trained

Our **supervised learning** approach enables detection without generating many FPs

Labels in Microsoft

For supervised learning, Azure gets labeled data through:

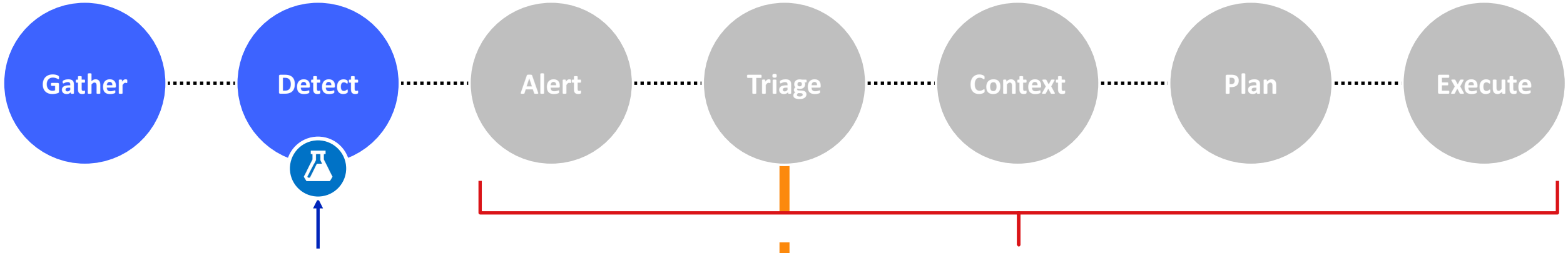


False Positives

Manual Triage

For Attack Disruption, we need to think beyond detection

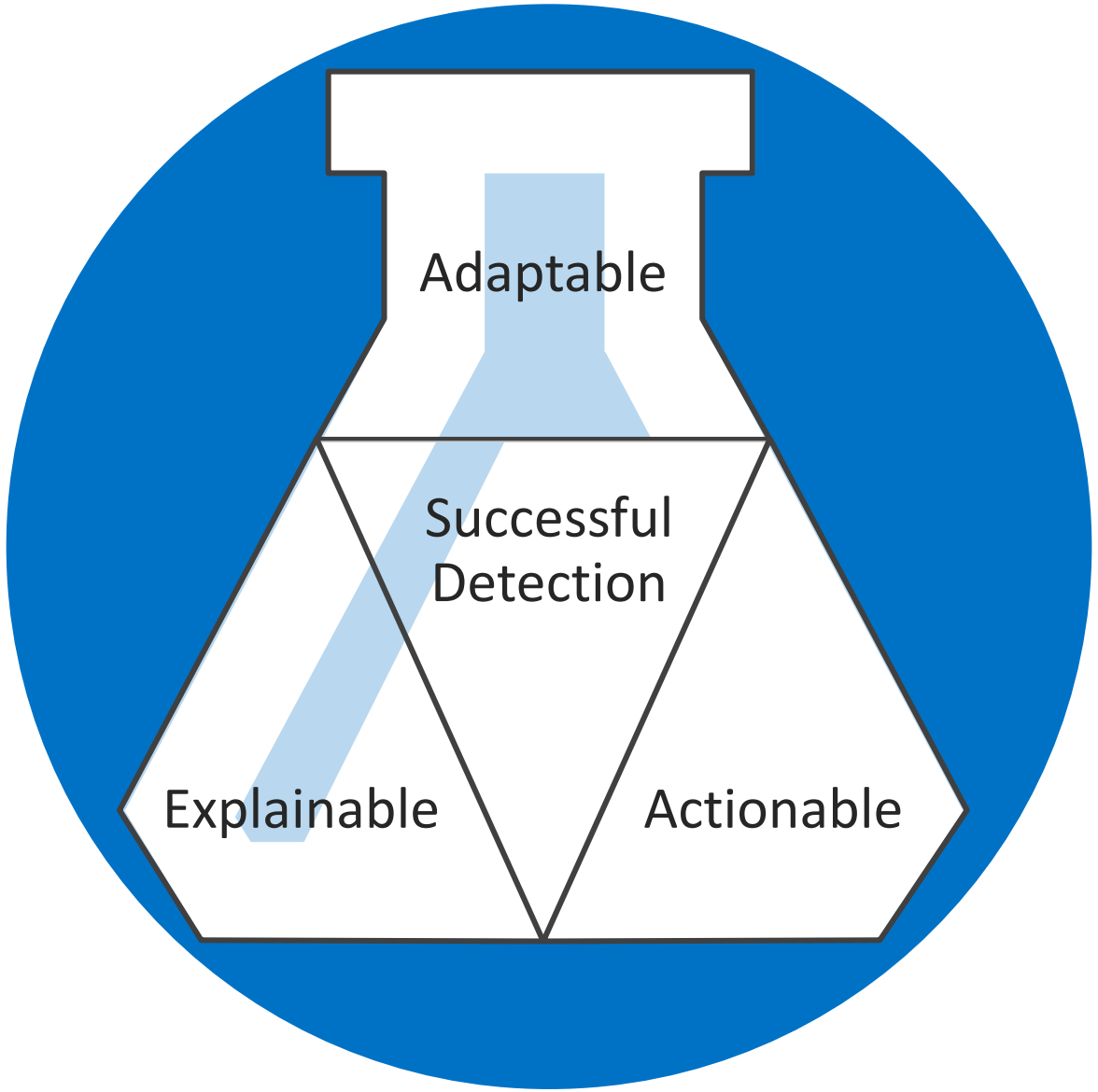
Manual Triage



Current use of Machine Learning

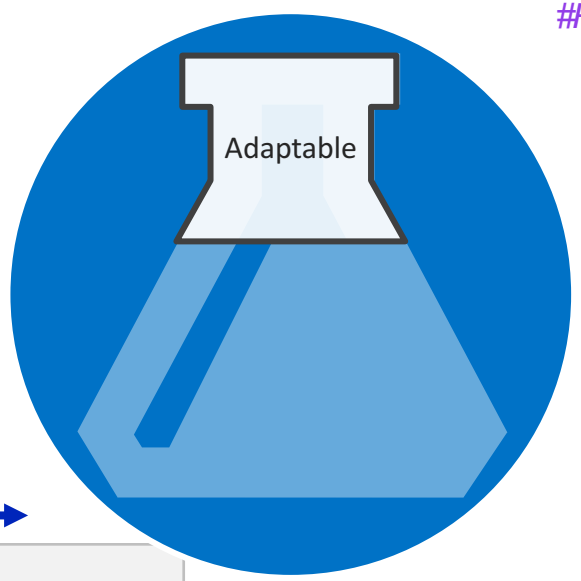
Everything is manual, with little or no intelligence – We need to change this

Properties of a Successful Machine Learning Solution



Adaptable in Cloud is Difficult

Why?



EVOLVING LANDSCAPE

Frequent deployments

New services coming online

Usage spikes

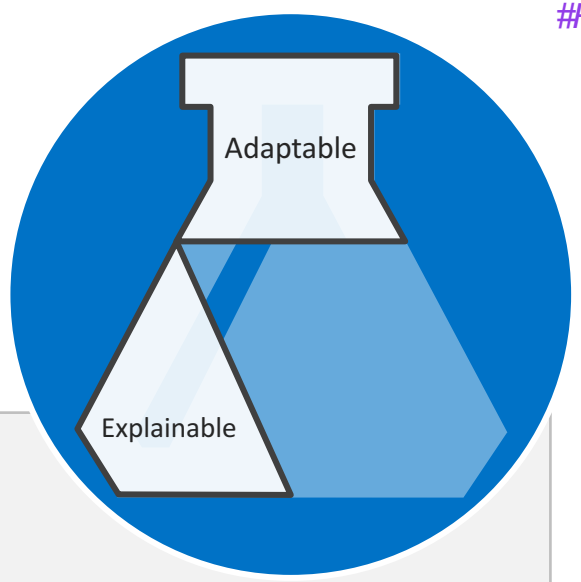
EVOLVING ATTACKS

Constantly changing environments leads to constantly changing attacks

- New services
- New features for existing services

Explainability

Why?



Surfacing a security event to an end-user can be useless if there is no explanation

Explainability of results should be considered at earliest possible stage of development

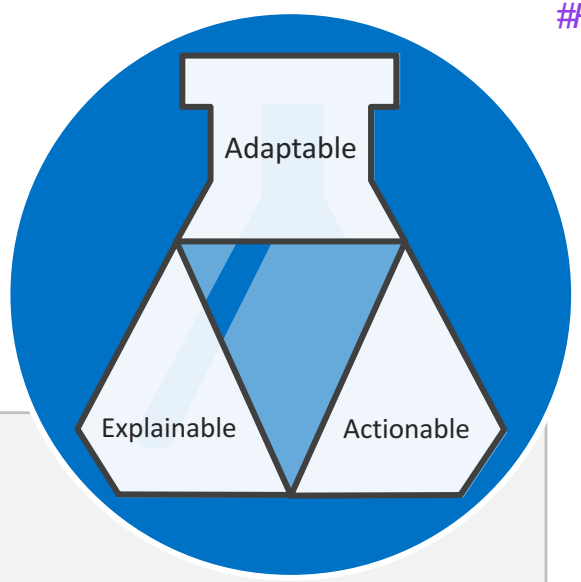
Results without explanation are hard to interpret

Best detection signal with no explanation might be dismissed/overlooked

<Example – How do you explain this to an analyst>

UserId	Time	EventId	Feature1	Feature2	Feature3	Feature4	…	Score
1a4b43	2016-09-01 02:01	4688	0.3	0.12	3.9	20	…	0.2
73d87a	2016-09-01 03:15	4985	0.4	0.8	0	11	…	0.09
9ca231	2016-09-01 05:10	4624	0.8	0.34	9.2	7	…	0.9
5e9123	2016-09-01 05:32	4489	2.5	0.85	7.6	2.1	…	0.7
1e6a7b	2016-09-01 09:12	4688	3.1	0.83	3.6	6.2	…	0.1
33d693	2016-09-01 14:43	4688	4.1	0.63	4.7	5.1	…	0.019
7152f3	2016-09-01 19:11	4688	2.7	0.46	3.9	1.4	…	0.03

Actionable Detections



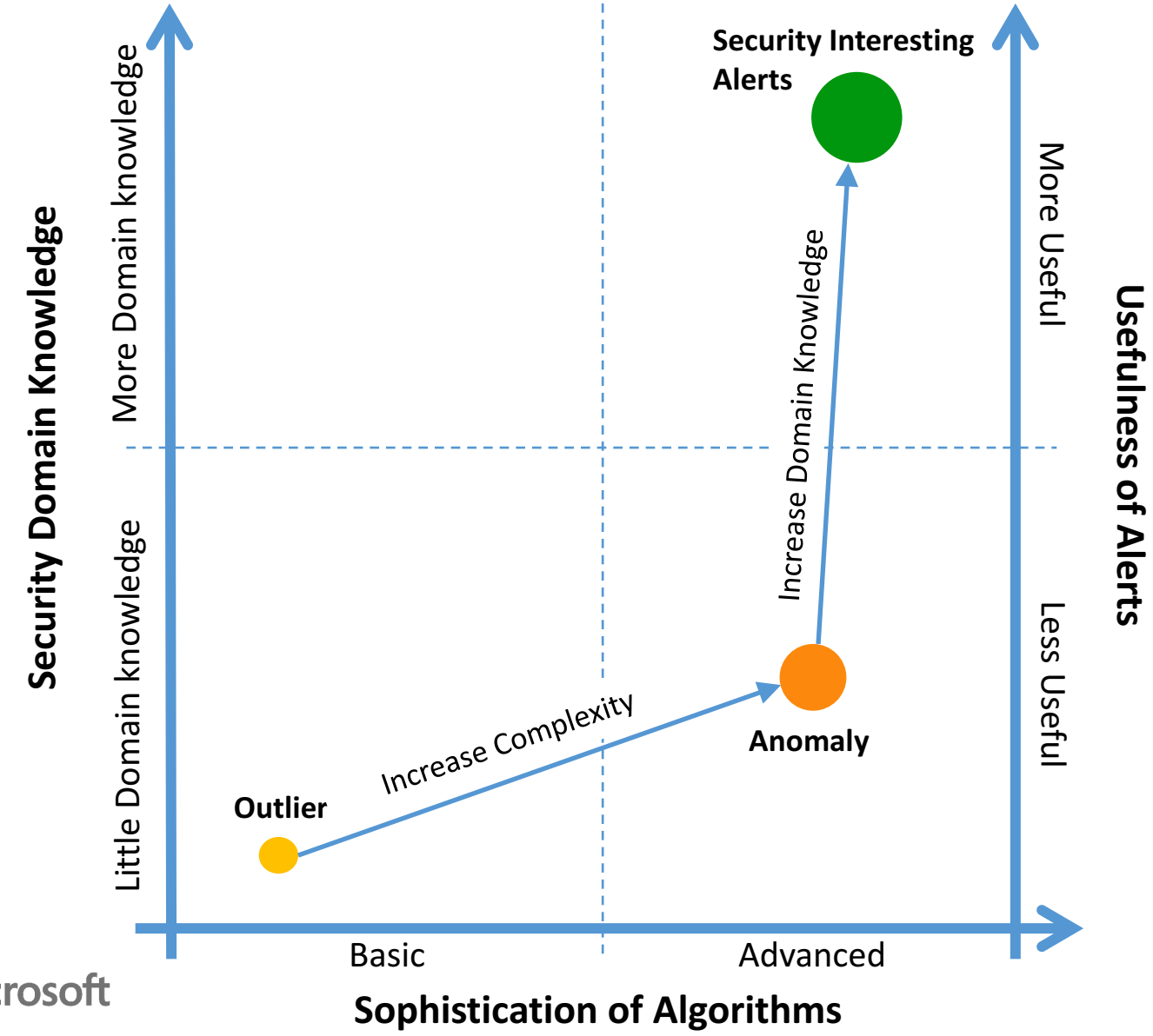
Detections must result in downstream action

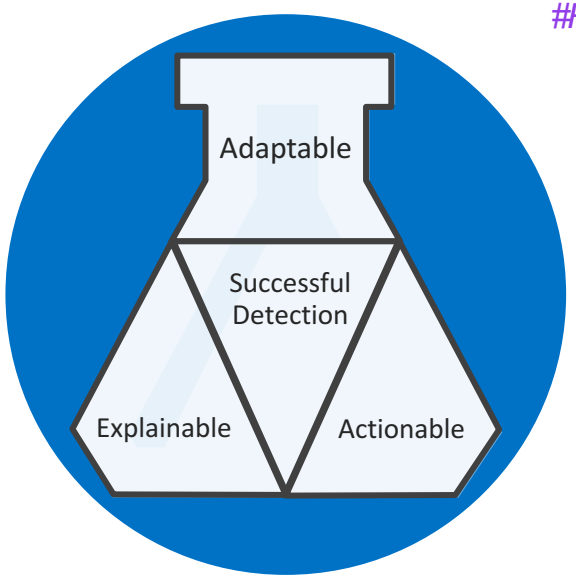
Good explanation without being actionable is of little value

EXAMPLES

- Policy decisions
- Reset user password

Framework for a Successful Detection





Successful Detections incorporate domain knowledge through disparate datasets and rules

Case Study 1

Successful detection through combining disparate datasets

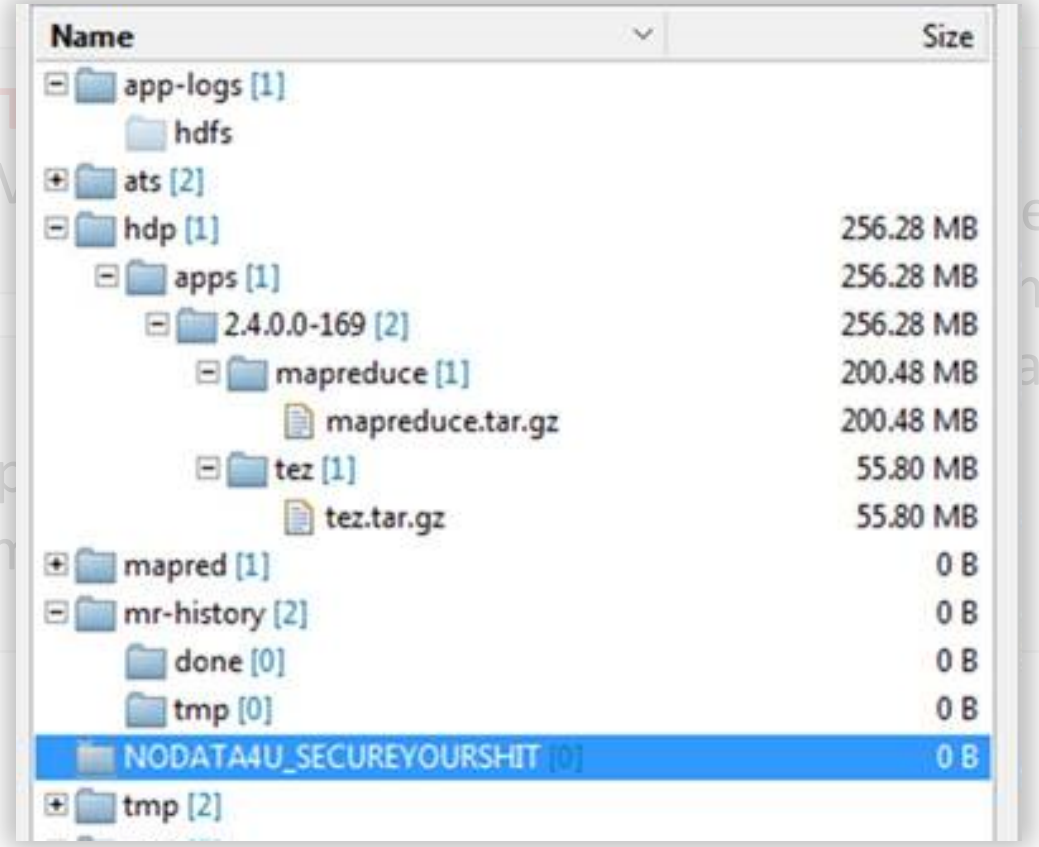
PROBLEM STATEMENT

Detect compromised V

HYPOTHESIS

If the VM is sending sp
is most likely compron

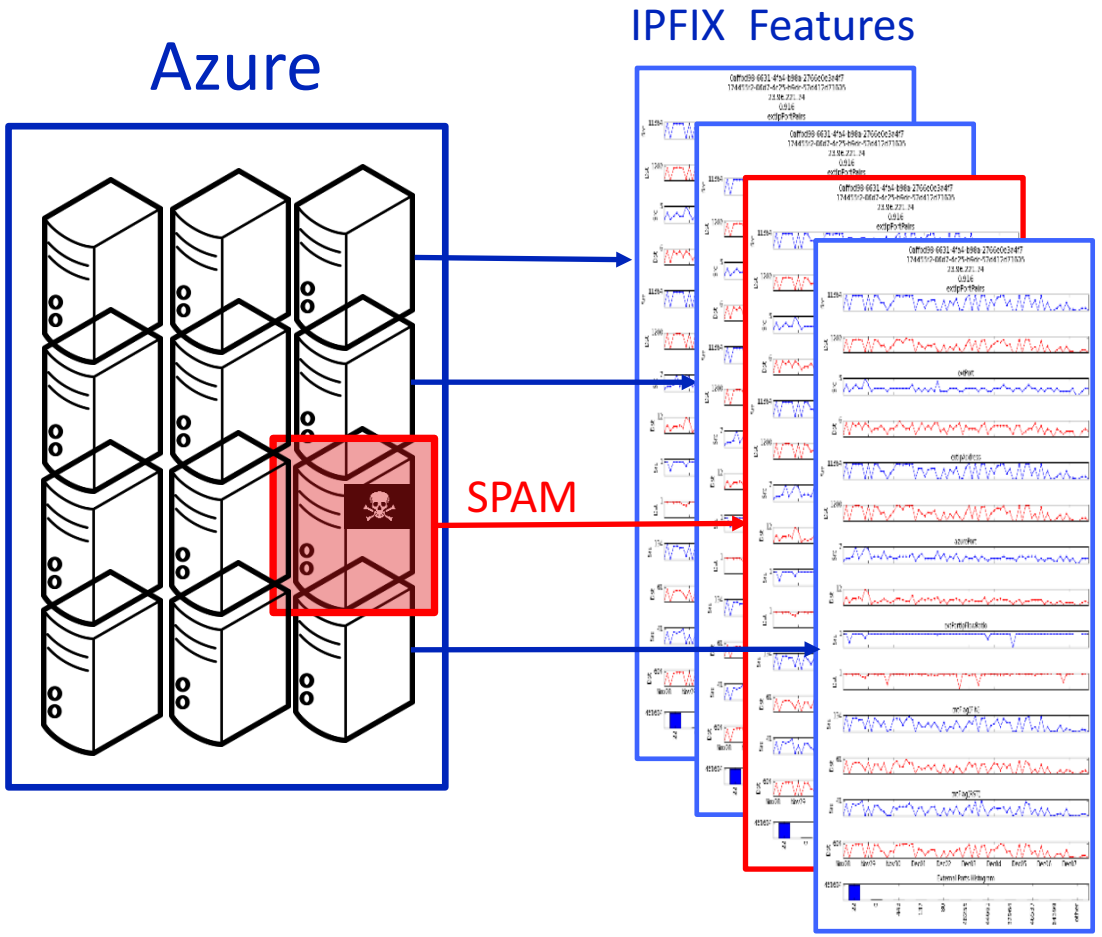
e Learning to leverage
n Office365 and
a from Azure.



Name	Size
app-logs [1]	
hdfs	
ats [2]	
hdp [1]	256.28 MB
apps [1]	256.28 MB
2.4.0.0-169 [2]	256.28 MB
mapreduce [1]	200.48 MB
mapreduce.tar.gz	200.48 MB
tez [1]	55.80 MB
tez.tar.gz	55.80 MB
mapred [1]	0 B
mr-history [2]	0 B
done [0]	0 B
tmp [0]	0 B
NODATA4U_SECUREYOURSHIT [0]	0 B
tmp [2]	

Case Study 1

Technique Overview



EXAMPLES

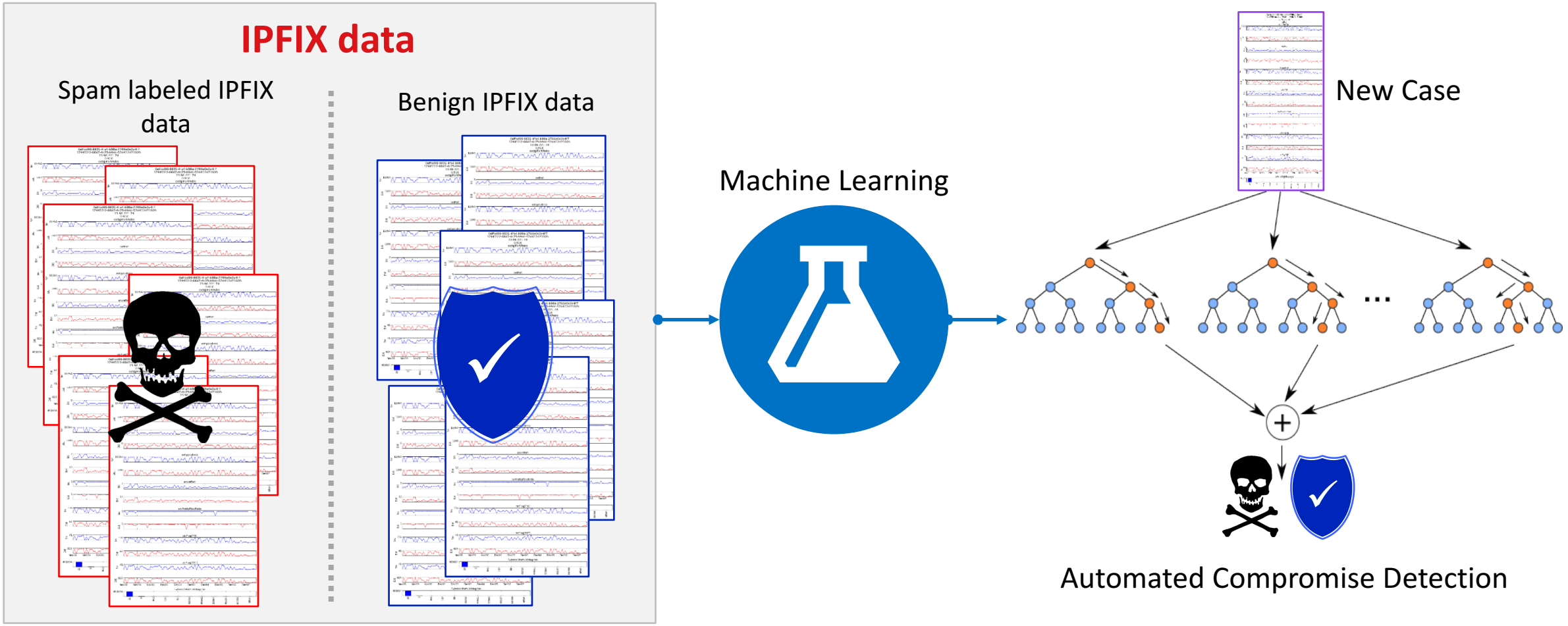
Automated

- All ports with traffic
- Number of connections
- Which TCP flags combination exist
- Many more…

Spam Tags come from O365!

Case Study 1

Technique Overview



Case Study 1

Dataset

WHY IS NETWORK DATA GOOD FOR DETECTION?

- ✓ No installation required – running on all Azure tenants
- ✓ No overload on the VM
- ✓ Resilient – cannot be maliciously turned off
- ✓ OS independent

FEATURE SOURCES

External IPs

External Ports

TCP flags

FEATURE TYPES

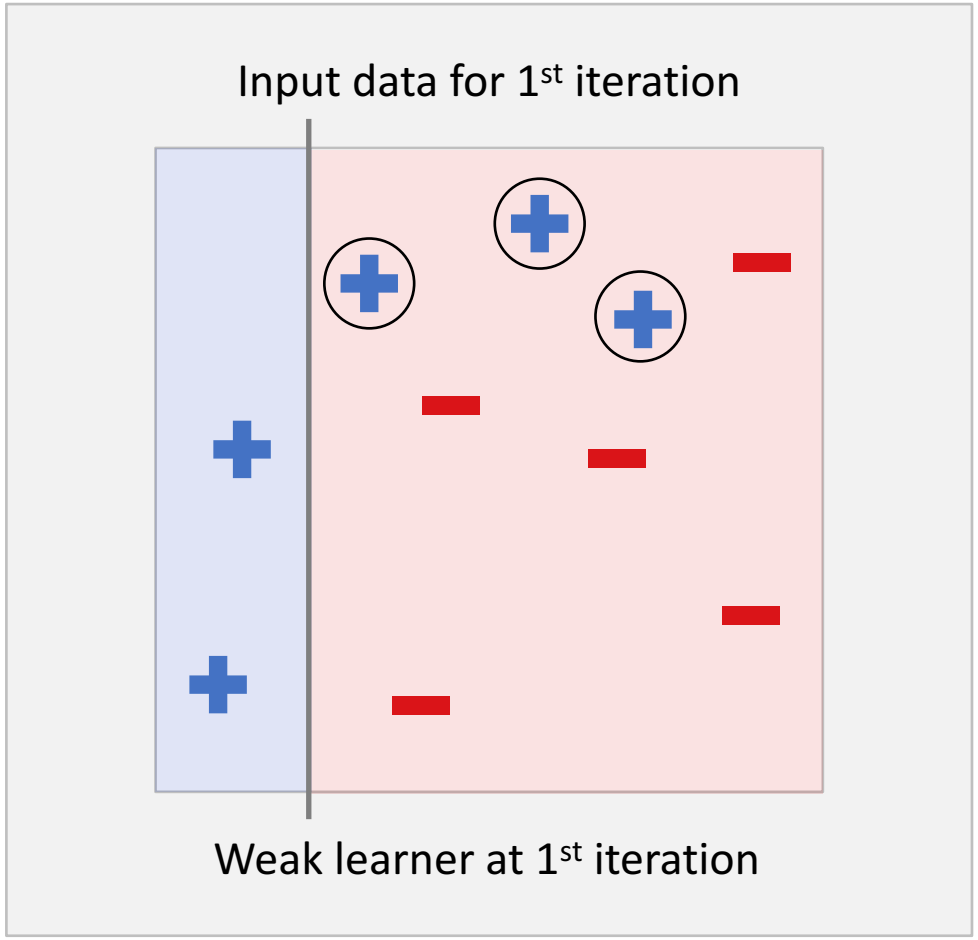
Existence (binary)

Counts

Normalized counts

Case Study 1

Machine Learning Deep Dive: **Gradient Boosting**

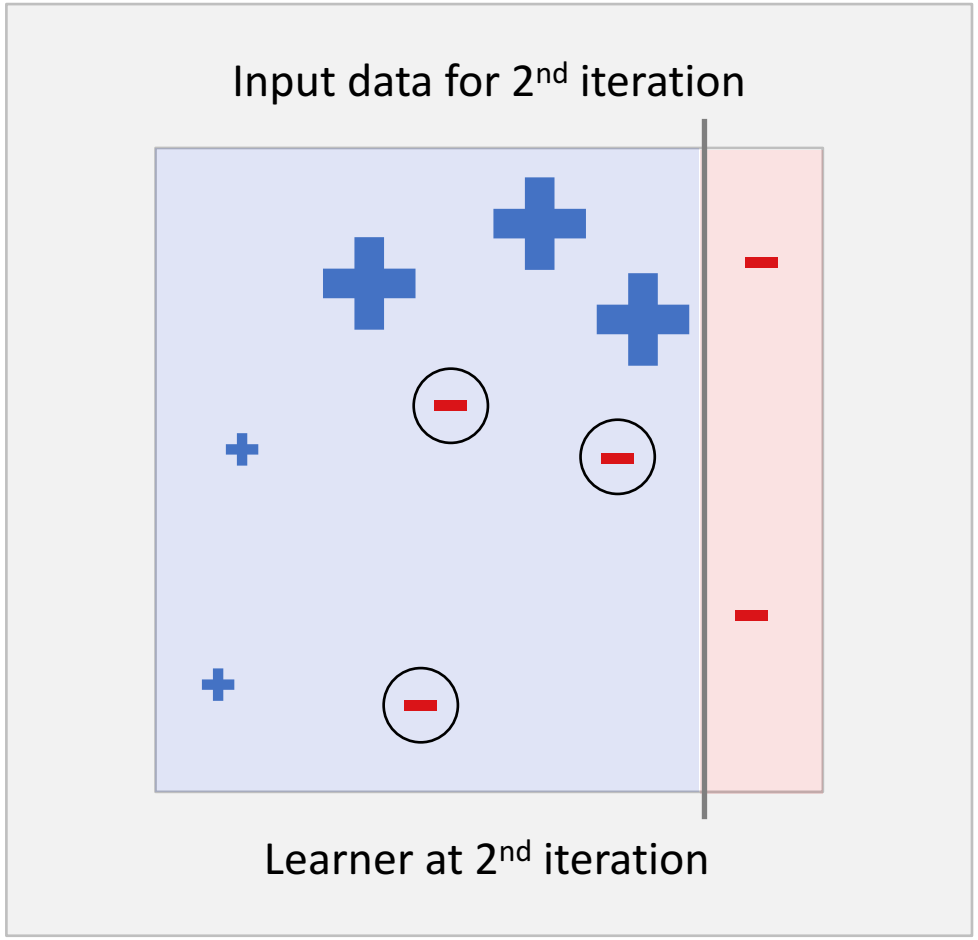


Case Study 1

Machine Learning Deep Dive: **Gradient Boosting**

The data points that were incorrectly categorized by the weak learner in the first iteration (the positive examples) are now weighted more.

Simultaneously, the correct points are down weighted.



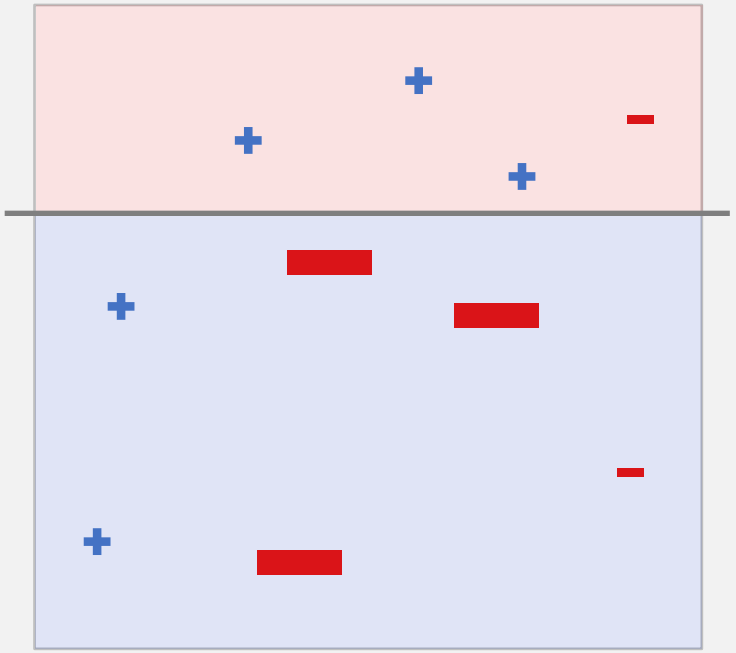
Case Study 1

Machine Learning Deep Dive: **Gradient Boosting**

The data points that were incorrectly categorized in the second iteration (the negative examples) are now weighted more.

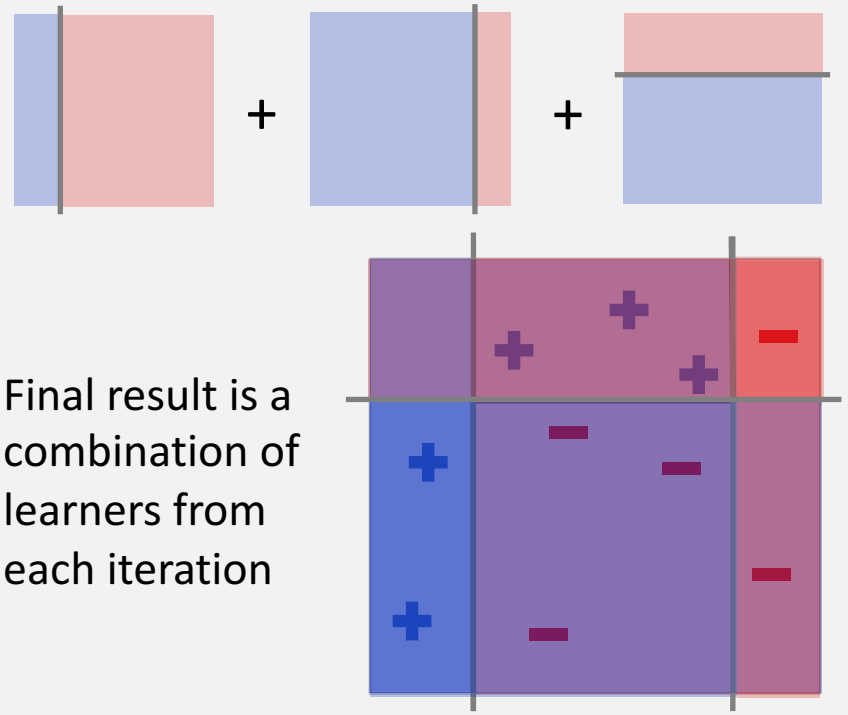
Simultaneously, the correct points are down weighted.

Input data for 3rd iteration



Learner at 3rd iteration

Results



Final result is a combination of learners from each iteration

Case Study 1

Model Performance and Productization

Model trained in regular intervals

Size of data: 360GB per day

Within minutes

Classification runs multiple times a day

Completed within seconds

Dataset	True Positive Rate	False Positive Rate
Only using Azure IPFIX data	55%	1%
Using Azure IPFIX and O365 data	**81%**	**1%**

26 points improvement

Microsoft

Possible outgoing spam activity detected

eventconsumer-prod-01-p-cus

DESCRIPTION	Network traffic analysis detected suspicious outgoing traffic from eventconsumer-prod-01-p-cus. This traffic may be a result of a spam activity. If this behavior is intentional, please note that sending spam is against Azure Terms of service. If this behavior is unintentional, it may mean your machine has been compromised.
DETECTION TIME	Wednesday, August 24, 2016, 9:00:00 AM
SEVERITY	Medium
STATE	Active
ATTACKED RESOURCE	
SUBSCRIPTION	
DETECTED BY	Microsoft
ACTION TAKEN	Detected
REMEDIATION STEPS	1. Login to the machine in question to check if the SMTP connections are legitimate. 2. Check which local application is communicating with an external SMTP service (port 25) using netstat/tcpdump. 3. Make sure the machine has all the latest security patches and an updated Anti-Virus installed 4. Escalate the alert to the information security team and consider disconnecting the VM from the network until proper investigation has been made For Windows VMs: 1. Install and run Microsoft's Malicious Software Removal Tool (see https://www.microsoft.com/en-us/download/malicious-software-removal-tool-details.aspx) 2. Run Microsoft's Autoruns utility and try to identify unknown applications that are configured to run at login (see https://technet.microsoft.com/en-us/sysinternals/bb963902.aspx) 3. Run Process Explorer and try to identify unknown running processes (see

Case Study 2

Successful detection through combining rules and machine learning

PROBLEM STATEMENT

Rule based malware detection place hard constraints if something is a malware or not. While they are specific, they have a lot of False Positives, False negatives and are not adaptable

HYPOTHESIS

Can we combine the hard logic of rule based detections with the soft - logic of machine learning systems?

SOLUTION

Build two ML models:

1) Model 1 that baselines malware behavior

2) Model 2 that incorporates rules as features

Combine result of two models

Case Study 2

MALWARE DETECTION BACKGROUND

ATP Architecture

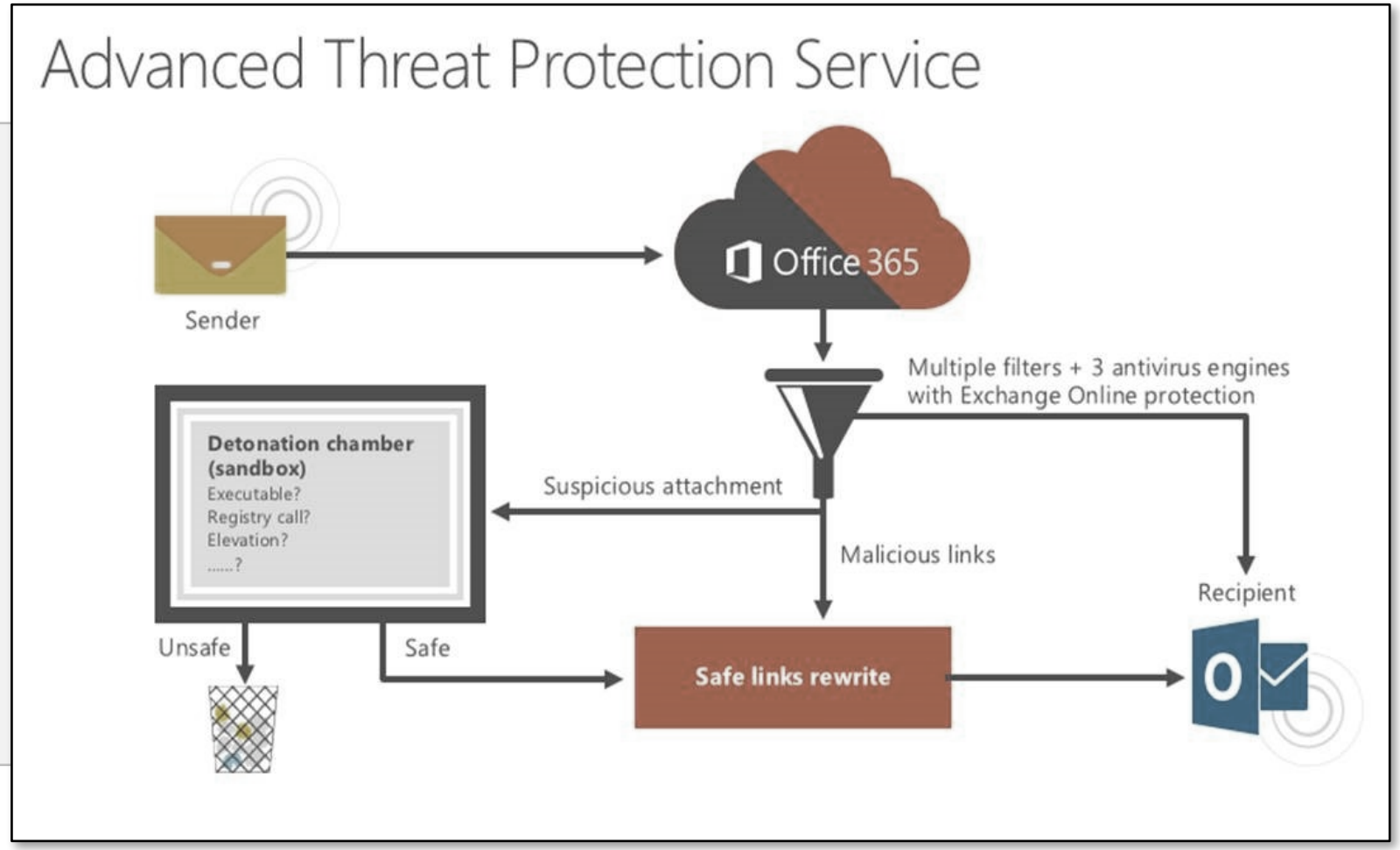
Conventional A/V

Detonation Chamber

- Spin up multiple VMs
- Multiple OS and Office versions
- Instrument attachment behavior

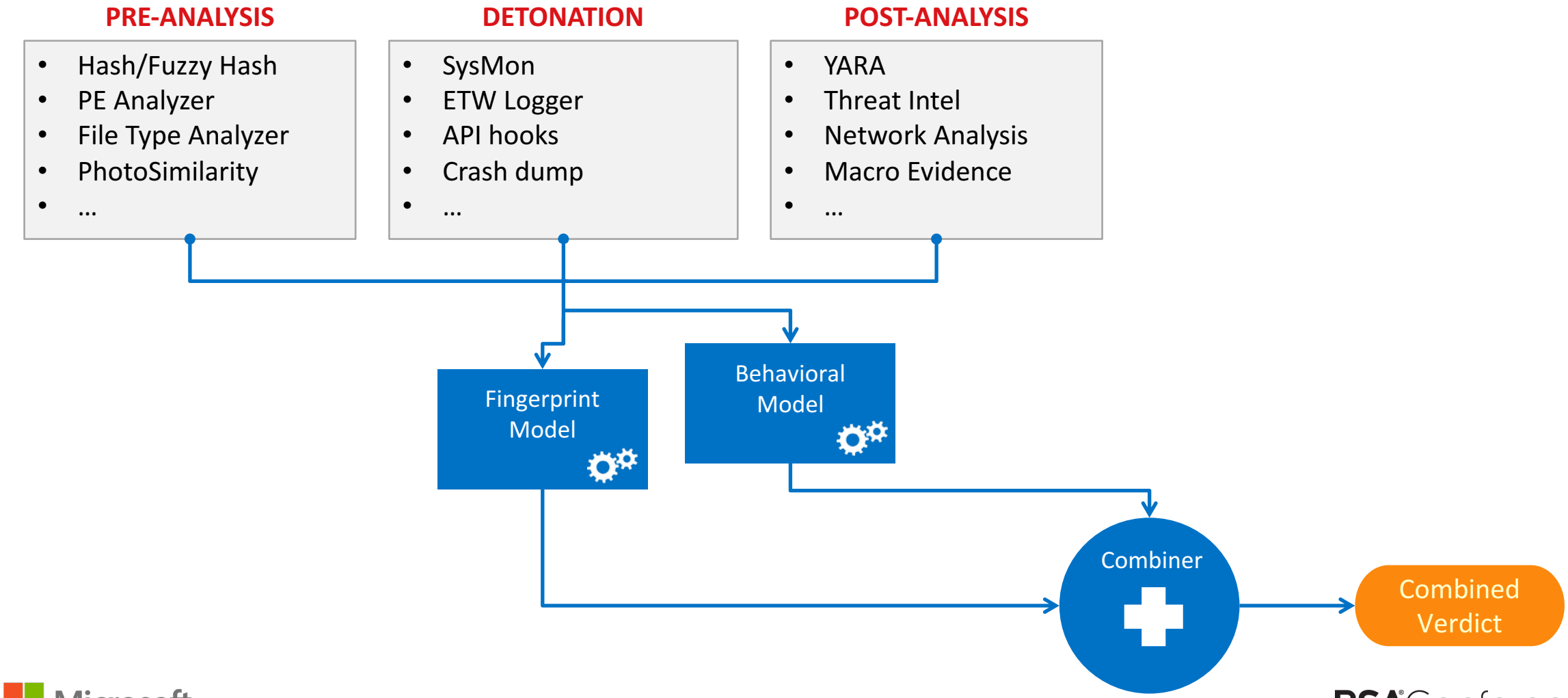
Safelinks

- Protects against malicious URLs in Real Time (on click)



Case Study 2

Technique Overview



Case Study 2

Dataset

(SAMPLE)


```
1  <?xml version="1.0" encoding="utf-8"?>
2  <Journal Types [redacted] TargetProcessId="2100" TargetFileName [redacted] Tar
3    <TargetProcessesByName />
4    <Events>
5      <LoadImage TaskName="ImageLoad" ProcessId="2100" ProcessName="Process:wscript" ParentPro
6        <ImageName>\Device\HarddiskVolume2\Windows\System32\wscript.exe</ImageName>
7        <ImageSize>155648</ImageSize>
8        <TimeDateStamp>1247528568</TimeDateStamp>
9        <ImageCheckSum>176852</ImageCheckSum>
10     </LoadImage>
11     <CallFunction TaskName="CallFunction" ProcessId="2100" ProcessName="Process:wscript" Par
12       <FunctionName>CreateMutexA</FunctionName>
13       <Param0>Local\_!MSFTHISTORY!_</Param0>
14     </CallFunction>
15     <CallFunction TaskName="CallFunction" ProcessId="2100" ProcessName="Process:wscript" Par
16       <FunctionName>CreateMutexW</FunctionName>
17       <Param0>Local\!IETld!Mutex</Param0>
18     </CallFunction>
19     <SetRegistryValue TaskName="SetValueKey" ProcessId="2100" ProcessName="Process:wscript"
20       <KeyName>\REGISTRY\MACHINE\SOFTWARE\Microsoft\Tracing\WScript_RASAPI32</KeyName>
21       <ValueName>EnableFileTracing</ValueName>
22       <Data>0x00000000</Data>
23       <DataType>REG_DWORD</DataType>
```

Case Study 2

Machine Learning Deep Dive: **Fingerprint Model**

Information gets more granular

Call Order	Level 1	Level 2	Level 3	Level 4	Level 5
1	Process	LoadImage	SYSTEM	.exe	wscript
2	Api	CallFunction	CreateMutexA	_!MSFTHISTORY!_	
3	Api	CallFunction	CreateMutexW	!IETld!Mutex	
4	Registry	SetRegValue	Tracing	wscript_rasapi32	EnableTracing
5	Registry	DeleteRegValue	InternetOption	internet settings	ProxyBypass
6	Process	CreateProcess	NOT_SANDBOX_CHECK	LaunchedViaCom	
7	Network	AccessNetWork	Wininet_Getaddrinfo		
8	Api	CallFunction	CreateMutexW	RANDOM_STR	
9	Network	ResolveHost	piglyeleutqq.com	UNKNOWN	
10	Api	CallFunction	Connect	UNKNOWN	

Case Study 2

Machine Learning Deep Dive: **Fingerprint Model Observations**

Benefits of the Action-Chain prototype

- It can be **RESILIENT** to malware obfuscation because it captures the runtime semantics by considering the more **IMPORTANT** details
- Feature extraction is **NON-PARAMETRIC**
  - Would generalize to many situations

Model

Current: L1 Logistic Regression followed by L2 Logistic Regression; weighted samples through cross-validation

Case Study 2

Machine Learning Deep Dive: **Behavioral Model**

Incorporates security domain knowledge into the model

Source of features

- YARA rules
- Static analysis
- Aggregates from Data:
  - Registry keys/values that are changed/created/deleted
  - Mutexes created
  - Number of spawn processes per process detail info

The model works well to detect new types malware

Case Study 2

Model Performance and Productization

Model trained in regular intervals

Size of data: 270GB per day

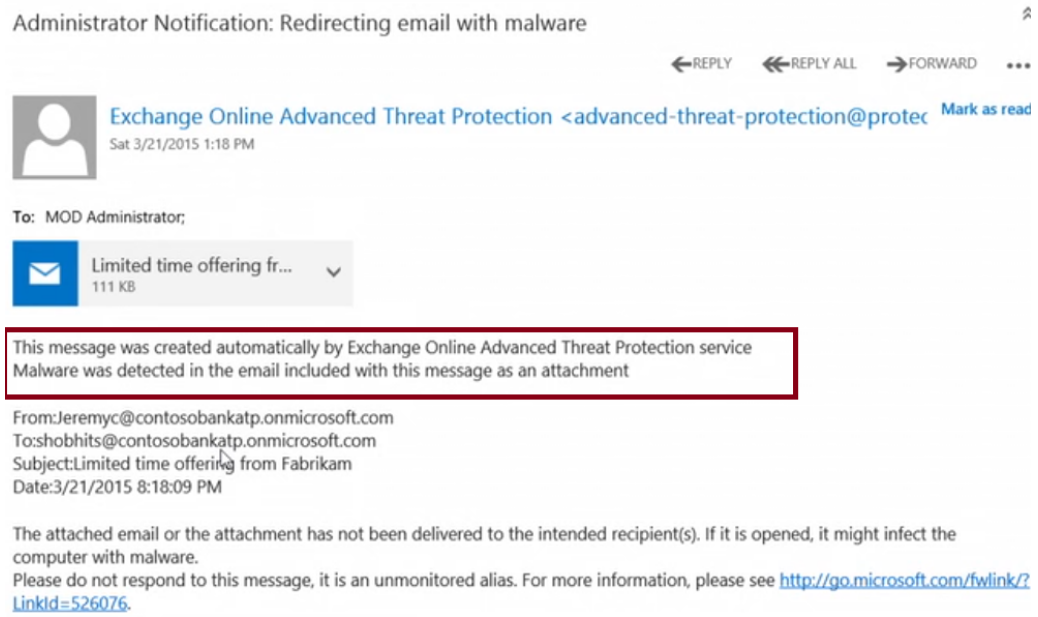
Completed within minutes

Classification runs multiple times a day

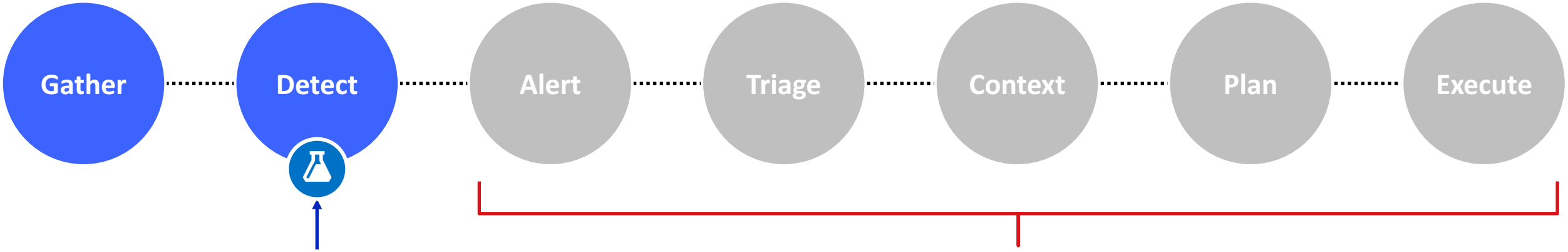
Completed within milliseconds

Dataset	True Positive Rate	False Positive Rate
YARA rules only	82.6%	0.0178%
Machine Learning Model 1 + Model 2	**93.6%**	**0.0127%**

10 points improvement



For Attack Disruption, We Need to Think Beyond Detection



Current use of Machine Learning

Everything is manual, with little or no intelligence – We need to change this

Triage incidents, not alerts

Anomalous DLL: rundll32.exe launched as sposql11 on CFE110095

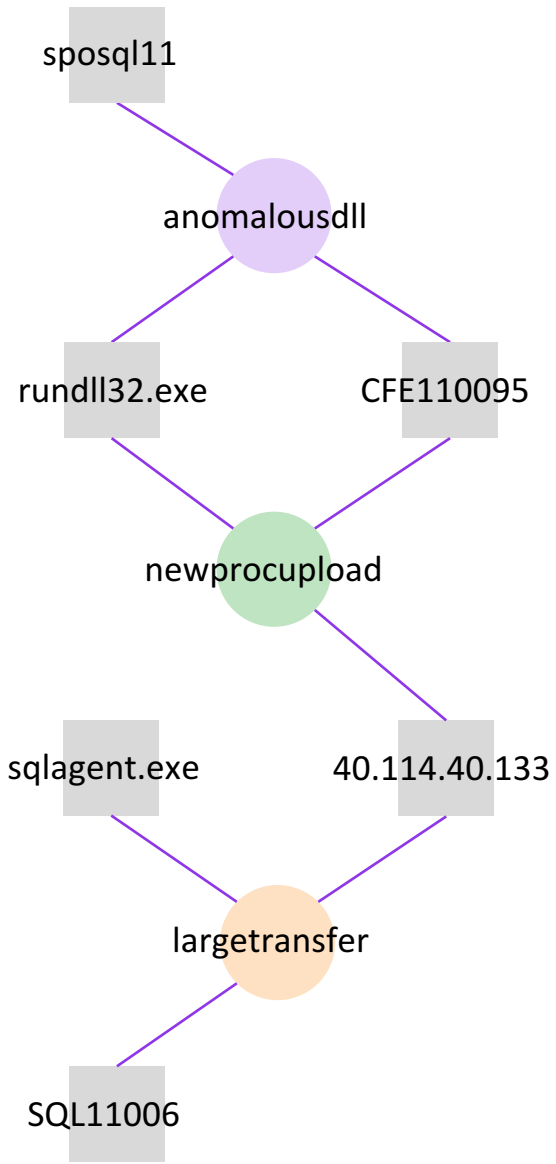
alert type *process* *user* *host*

New process uploading: rundll32.exe to 40.114.40.133 on CFE110095

alert type *process* *remote host* *host*

Large transfer: 50MB to 40.114.40.133 from sqlagent.exe on SQL11006

alert type *remote host* *process* *host*



Triage incidents, not alerts

Demo

Conclusion

Attack Disruption means to shorten blue team kill chain

Attack Disruption Checklist

- Combine different datasets
- Labels, Labels, Labels
- Scalable ML solution and expertise
- Example Azure services you can leverage:

Azure Event Hubs	Azure Data Lake	Azure Machine Learning

Thank you