RSAConference2016

San Francisco | February 29 – March 4 | Moscone Center

Connect to Protect

EXP-W04

Machine Learning and the Cloud: Disrupting Threat Detection and Prevention

Mark Russinovich

CTO, Microsoft Azure
Microsoft Corporation
@markrussinovich

#RSAC

Microsoft’s daily cloud security scale

10s of PBs
of logs

1+ billion
Azure Active Directory logons

300+ million
active Microsoft Account users

Detected/ reflected attacks

> 10,000
location-detected attacks

1.5 million
compromise attempts deflected

Security data explosion

Useful Data	Web server logs	Windows Event logs, Linux syslog	Network logs
	Fabric	Data center security token service	Cloud service logs

Challenges with Standard Security Detection Systems

Weak independent alert streams

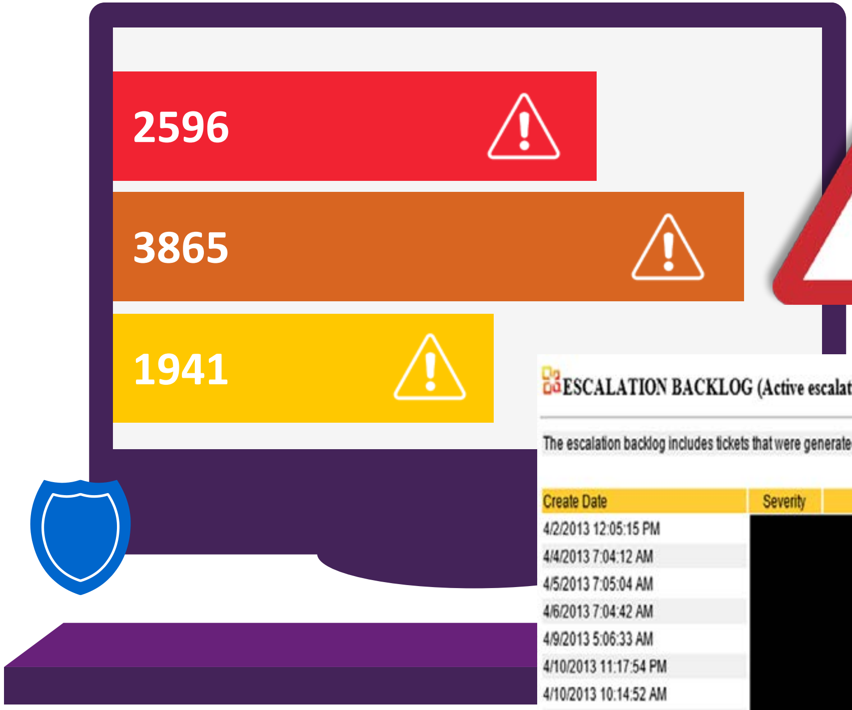
My Escalation Backlog

https://escalation-report-uri.cloudapp.net/escalation-backlog#sampleData

This escalation backlog includes tickets generated more than 8 hours ago. Please prioritize and triage the backlog to confirm the activity.

Created	Severity	Task	Assigned To	Category
2/27/2016				Sever Data Health
3/1/2016				Event Count Outliers
3/1/2016				Failed Logins
3/1/2016				Failed Logins
3/2/2016				Event Count Outliers
3/2/2016				Firewall Change

Burden of triage



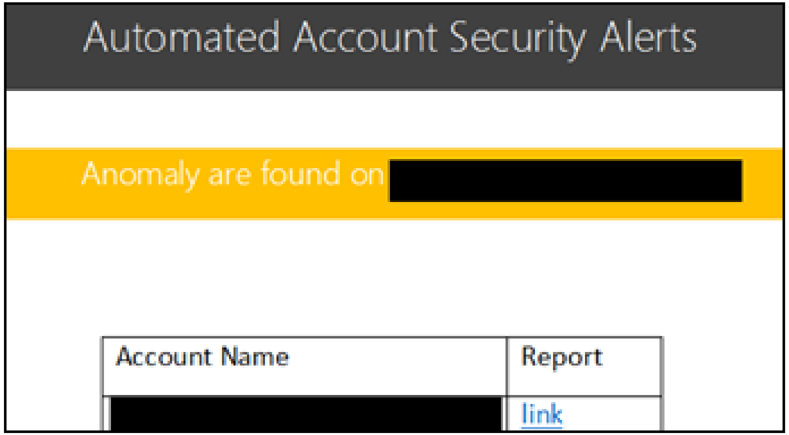
Active alerts to triage

ESCALATION BACKLOG (Active escalations older then 24 hours)

The escalation backlog includes tickets that were generated more then 48 hours ago. Your workload should never have security escalations that go unresolved for more then 48 hours.

Create Date	Severity	Bug ID	Assigned To	Category
4/2/2013 12:05:15 PM	[redacted]	[redacted]	[redacted]	Server Data Health Issues
4/4/2013 7:04:12 AM	[redacted]	[redacted]	[redacted]	Event Count Outliers
4/5/2013 7:05:04 AM	[redacted]	[redacted]	[redacted]	Event Count Outliers
4/6/2013 7:04:42 AM	[redacted]	[redacted]	[redacted]	Event Count Outliers
4/9/2013 5:06:33 AM	[redacted]	[redacted]	[redacted]	Server Data Health Issues
4/10/2013 11:17:54 PM	[redacted]	[redacted]	[redacted]	Failed Logins - Internal Accounts
4/10/2013 10:14:52 AM	[redacted]	[redacted]	[redacted]	Failed Logins - Internal Accounts
4/10/2013 5:40:42 PM	[redacted]	[redacted]	[redacted]	Failed Logins - Internal Accounts

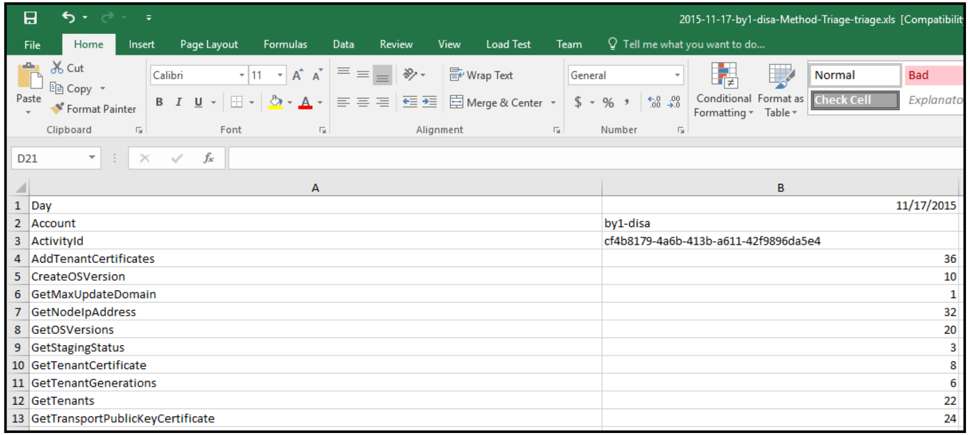
Interpretability of alerts



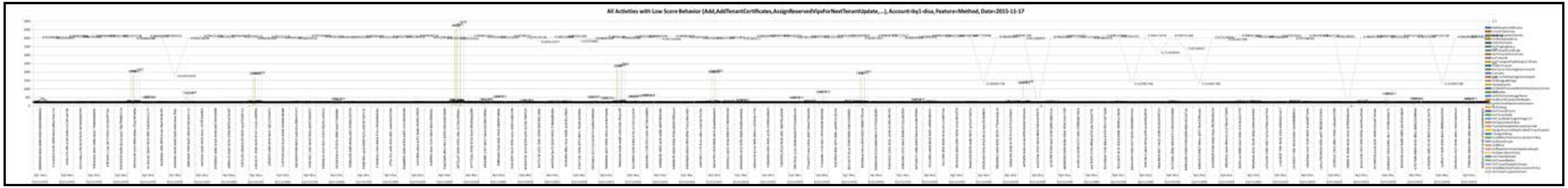
Automated Account Security Alerts

Anomaly are found on [redacted]

Account Name	Report
[redacted]	link



	A	B
1	Day	11/17/2015
2	Account	by1-disa
3	ActivityId	cf4b8179-4a6b-413b-a611-42f9896da5e4
4	AddTenantCertificates	36
5	CreateOSVersion	10
6	GetMaxUpdateDomain	1
7	GetNodeIpAddress	32
8	GetOSVersions	20
9	GetStagingStatus	3
10	GetTenantCertificate	8
11	GetTenantGenerations	6
12	GetTenants	22
13	GetTransportPublicKeyCertificate	24



Lack of feedback loop

How Machine Learning can help

Reduce triage of burden by PRIORITIZING ALERTS

COMBINING INDEPENDENT ALERT STREAMS and providing informed scoring

Account Name	Overall Triage Status
[redacted]	Triage-P1
	Triage-P1
	Triage-P1
	Not-For-Ticketing
	Not-For-Ticketing
	Not-For-Ticketing
	Not-For-Ticketing
	Not-For-Ticketing
	Not-For-Ticketing

Each alert combines multiple points:

- Is the sequence of API calls unusual for this account?
- Is the IP address unusual?
- Does the time of access look normal?

For our DevOps anomaly detection, we combine over 8 different weaker streams.

How Machine Learning can help

Incorporating analyst/user feedback TO IMPROVE THE SYSTEM SIGNAL

From: [redacted]
Sent: [redacted]
To: [redacted]
Subject: [ACTION REQUIRED] Please confirm your recent account activity

We detected the following activity [redacted] and [redacted] from [redacted]

Was this you?

Yes, this was me

No, something's not right

PROVIDING INTERPRETABLE RESULTS

When we get an alert, we're informed exactly why the ML system feels it is anomalous. Not a black box.

Unusual UserAgent	Logins Eval	Unusual Location	Failed Login	Unusual IP	Unusual Activity	Overall Score
1	1	0	0	37	324	197106
0	0	0	0	0	64	134460
0	5	0	0	25	0	521308
5	3	0	0	0	0	33648
0	0	0	0	3048	0	129
0	2	0	1	3	0	94

Machine Learning for Security

An Introduction

How ML is different

Traditional Programming

Data

Program

Computer System

Output

Machine Learning

Data

Output + Assumptions

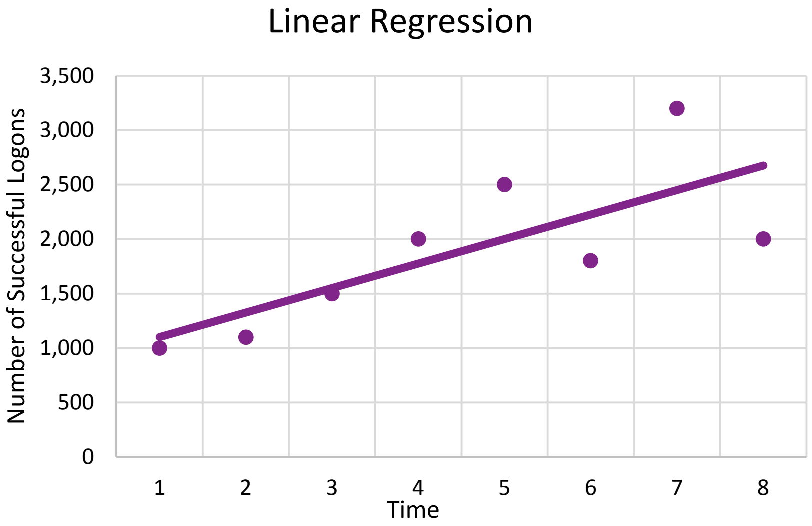
Computer System

Program

Source: Lecture by Prof. Domingos

Components of a ML system

TASK	E.g: Predict number of Logons in a end-system
LEARNER	Linear Regression
FEATURE	Count of logons over time
DATA	Security Event logs



Number of Successful Logons = 225 * Time + 875

Machine Learning for security is difficult

Lack of ground truth

Data labeled as an attack is rare

Datasets are imbalanced

Disproportionate cost of false negative (missing an attack)

Constantly changing environment

Adversarial setting: deliberately avoiding detection

The data labelling challenge

PROBLEM

You don't know what anomalous activity looks before hand

PROBLEM

Difficult to determine 'good' behavior

SOLUTION

CRAWL: Use publically available data sets to test

CON: Attacker has access to this too! Also, not every dataset is applicable

WALK: Set up Honeypots to collect attack data

CON: Data is not generated on-demand.

RUN: Have a Red Team validate your detection as part of an exercise

Microsoft has access to high quality attack data through MSRC, O365 Advanced Threat Protection, MMPC, DCU.

ML Algorithms for Security

DevOps anomalies

- Identify user and service accounts
- Detect and alert on privileged access anomalies

DevOps / Engineers / Support Experts

Automated Service Identities

Identities

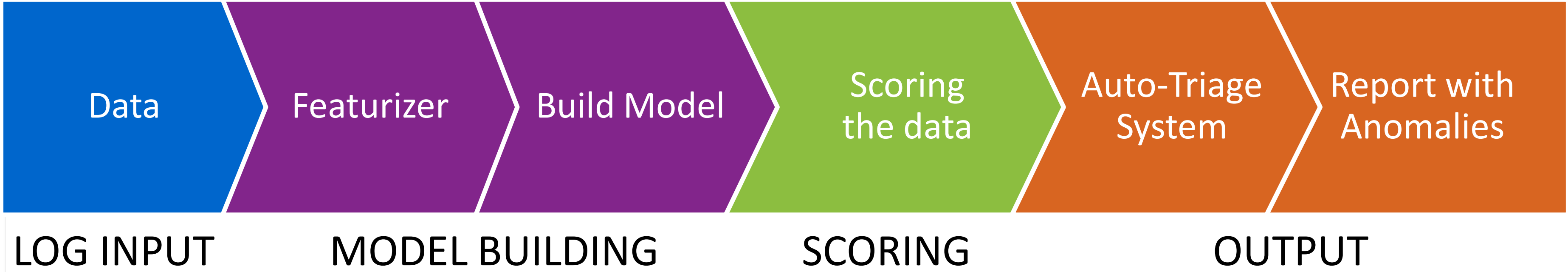
Security Tokens
Secrets & Certificates
RDP/SSH Sockets

Storage Accounts
SQL Databases
Table Store
Queues / Service Bus

Virtual Machines
Cloud Services
Resource Providers
Fabric

Cloud Infrastructure & Services

DevOps anomaly detection

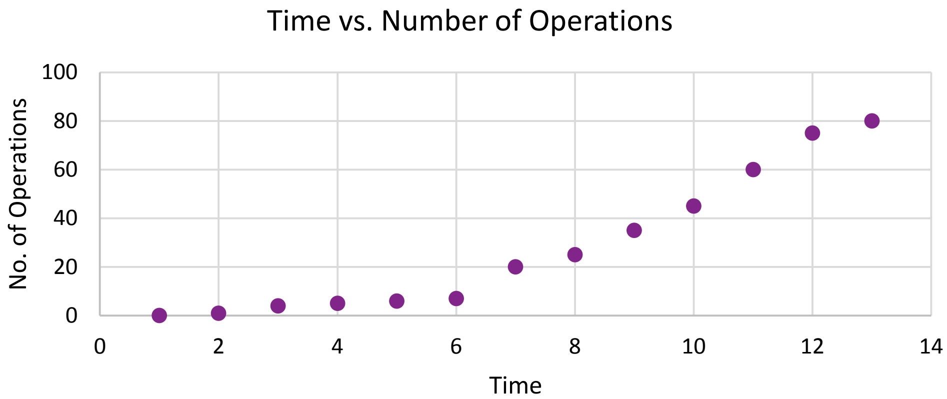


The anomaly detection problem

Two Features or Dimension

N=12
i.e. 12 examples

Time of Day	Number of Operations
1 AM	0
2 AM	1
3 AM	4
4 AM	5
5 AM	6
6 AM	7
7 AM	20
8 AM	25
9 AM	35
10 AM	45
11 AM	60
12 PM	75



- Given a new example, is it anomalous or not?

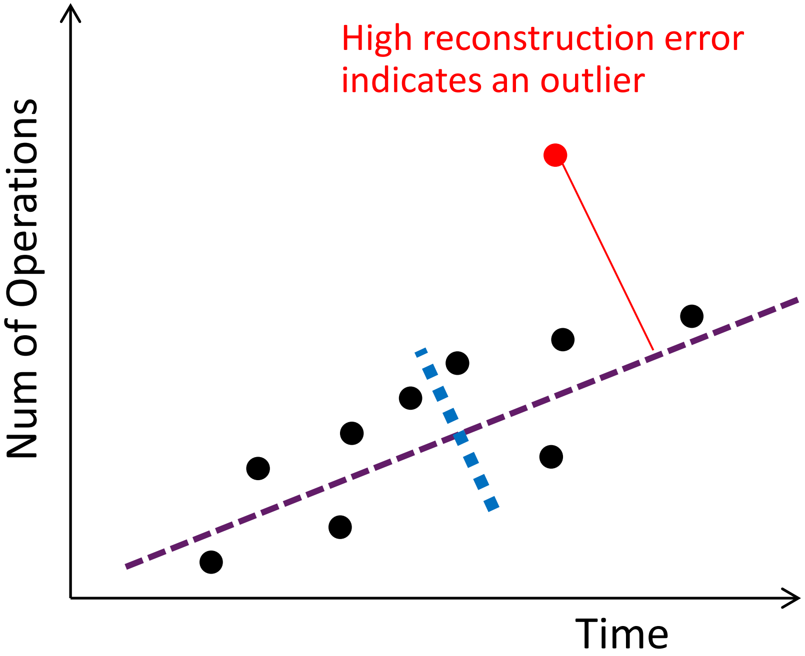
Principal Component Analysis (PCA)

#RSAC

Principal Components can intuitively be thought of as those directions that capture the most variation in the data.

Essentially, any point can be reconstructed as a linear combination of the principal components

Outlier = Any point that has high reconstruction error



PCA at Azure's scale

$O(Nd^2)+O(d^3)$

Traditional PCA:

- *Nxd* matrix of data, *N* examples, *d* features

Azure scale:

- N = 100,000,000+ data points and d = 1,000,000 features
- Order of 10^{23} operation

$O(dkN)$

Azure uses Distributed PCA with Random Projection

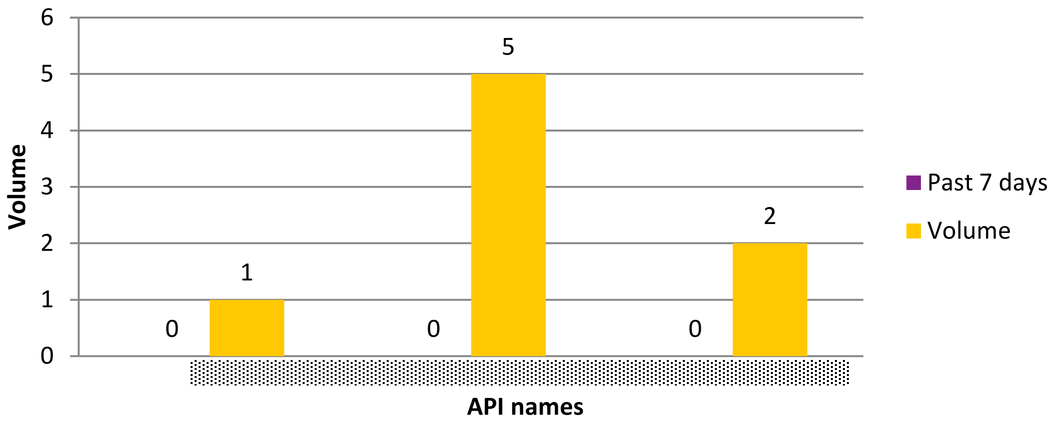
- Random Projection: We pick the directions/degrees of freedom to find interesting data
- Time complexity becomes *O(dkN), where k < d*
- **Model builds in 8 minutes**

Red team detection

Red Team abused 3 APIs in from a DevOps account in Azure Service

Machine learning model threw a P1-alert in the order of minutes with reason **"Suspicious API activity"**

Rank	Account Name	IP Address	Triage-Status	Reason	Report
1	[redacted]		Triage-P1	Suspicious API Activity	LINK



Geo-Anomaly Detection

Timestamp	Country	City, State	Service	State
Tue Nov 26, 13 21:45	US	New York, NY	Storage	Normal
Tue Nov 26, 13 22:57	US	New York, NY	Storage	Normal
Tue Nov 26, 13 23:24	US	New York, NY	Storage	Normal
Tue Nov 27, 13 01:27	IE	Dublin	Storage	Normal
Tue Nov 27, 13 07:31	CN	Shanghai	Storage	Suspicious
Tue Nov 27, 13 08:32	CA	Vancouver, BC	SQL	Suspicious

Intuitive geo anomaly detection

- Cache the last 10 locations of the user
- For current location:
  - If current location != cached locations, challenge user
  - If false positive, add current location to cache

Problems with rules only system

NOISY RESULTS

Company Proxy

Cellphone Networks

Vacations/Travel

A former rules-based Microsoft system scored

28% of logins as suspicious

1 billion logins per day =

280 million

"suspicious" logins

After applying

Machine Learning

the rate dropped to less than

0.001%

Accurate geo anomaly detection

SECURITY DUALITY

Maximize calling suspicious login behavior, Minimize friction/false positives caused by normal business routines (e.g. conferences, VPN's)

SOLUTION

Simple rules for determining suspicious login, large graph based machine learning approach for determining normal behavior

- Build up expected behavior by incorporating behavior of users similar behavior (but not all users)
- Model travel heuristics and device familiarity requirements
- Flag unexplainable remainder

Understanding user login patterns

Capture past login history

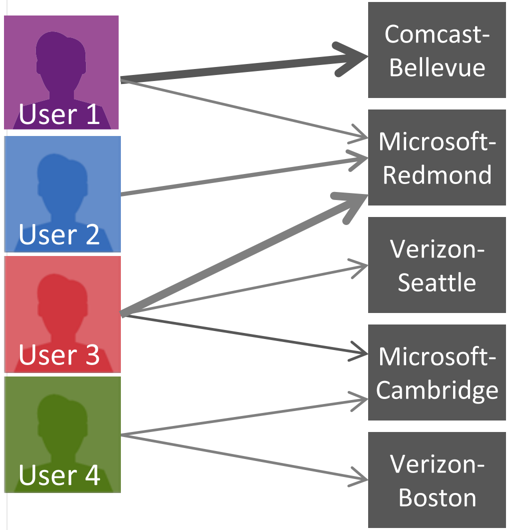
- 45 day window
- Weighted based on frequency/time last seen

Calculate user-user similarity

- Partial mapping between locations
- Constrained within tenants

Enumerate possible locations

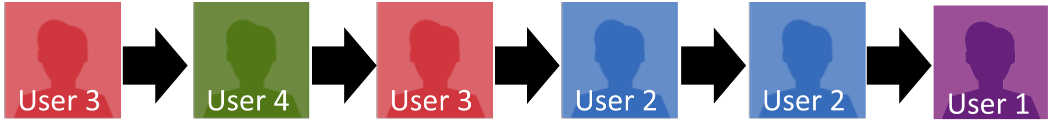
- Random walk with restarts
- Partial mapping to other similar Geo locations



	User 1	User 2	User 3	User 4
User 1	1.0	0.8	0.7	
User 2	0.8	1.0	0.7	
User 3	0.7	0.7	1.0	0.3
User 4			0.3	1.0

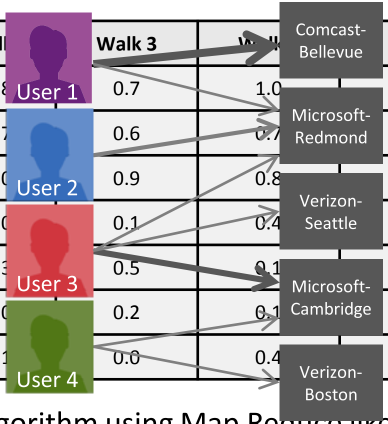
User	Location	Reachability
User 3	Comcast-Bellevue	965.0
User 3	Comcast-Redmond	875.0
User 3	Microsoft-Redmond	978.0
User 3	Verizon-Seattle	425.0
User 3	Verizon-Bellevue	350.0
User 3	Microsoft-Cambridge	275.0
User 3	Verizon-Boston	152.0
…	…	…

Random walk example



	User 1	User 2	User 3	User 4
User 1				
User 2				
User 3				
User 4				

Location	Walk 1	Wal	Walk 3	W	....	Walk 1000	Reachability
Comcast-Bellevue	0.7	0.8	0.7	1.0		0.9	965.1
Comcast-Redmond	0.6	0.7	0.6	0.7			
Microsoft-Redmond	0.9	1.0	0.9	0.8			
Verizon-Seattle	0.4	1.0	0.1	0.4			
Verizon-Bellevue	0.3	0.3	0.5	0.1			
Microsoft-Cambridge	0.5	0.0	0.2	0.1			
Verizon-Boston	0.2	0.1	0.0	0.4		0.7	132.0

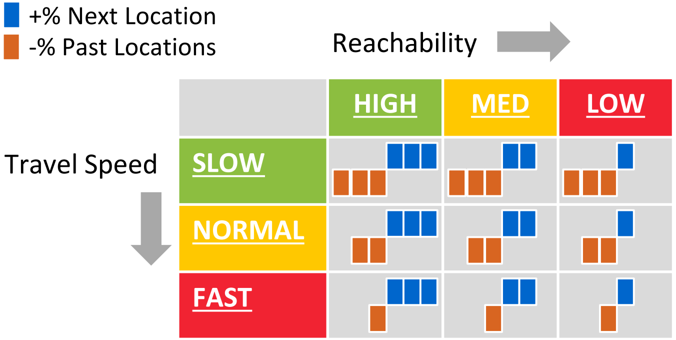


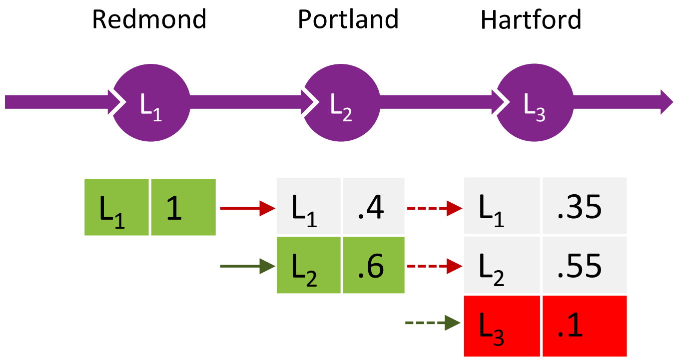
	User 1	User 2	User 3	User 4
User 1	1.0	0.8	0.7	
User 2	0.8	1.0	0.7	
User 3	0.7	0.7	1.0	0.3
User 4			0.3	1.0

- Training: Training of algorithm using Map-Reduce like framework (2 days)
- Evaluation: Approximations using Spectral Clustering and Linear Models allows fast evaluations (individual evaluation ~8ms)

How likely is a user in a location?

- Logging into location increases likelihood of being in a location, decreases likelihood of being in past locations
- Amplitude of change affected by speed of travel and the reachability of location
- Users logging into unlikely location with low probability are flagged as suspicious





Location	Time	Reachability
Seattle, WA -Comcast	Mar 02, 11:31 AM	800.1
Portland, OR-Verizon	Mar 02, 3:15 PM	119.2
Hartford, CT - Comcast	Mar 02, 4:16PM	45.8

Case study: phishing campaign

TimeStamp	Application	ClientIP	Country	City/State	Reachability	Call	Device
8/21/2015 1:21	Other	86.139.x	GB	Oundle	607.8	Normal	Windows 8.1;outlook.exe(Tablet PC)
8/23/2015 23:20	Other	5.148.x	GB	Kensington	279.2	Normal	Windows 8.1;outlook.exe(Tablet PC)
8/24/2015 7:23	Other	5.148.x	GB	Kensington	357.3	Normal	Windows 8.1;outlook.exe(Tablet PC)
8/24/2015 23:15	Other	5.148.x	GB	Kensington	357.3	Normal	Windows 8.1;outlook.exe(Tablet PC)
8/24/2015 23:22	Other	5.148.x	GB	Kensington	375.8	Normal	Windows 8;winword.exe(Tablet PC)
8/25/2015 1:17	Office 365	5.148.x	GB	Kensington	375.8	Normal	Windows 8.1;IE 11.0
8/25/2015 3:42	Office 365	41.206.x	NG	Lagos	44.5	Suspicious	Windows 7;Firefox 40.0
8/25/2015 7:18	Other	5.148.x	GB	Kensington	691.1	Normal	Windows 8.1;outlook.exe(Tablet PC)
8/25/2015 8:14	Other	5.148.x	GB	Kensington	691.1	Normal	Windows 8;excel.exe(Tablet PC)
8/25/2015 23:19	Other	5.148.x	GB	Kensington	691.1	Normal	Windows 8.1;outlook.exe(Tablet PC)
8/25/2015 23:58	Other	5.148.x	GB	Kensington	709.6	Normal	Windows 8.1;outlook.exe(Tablet PC)
8/26/2015 7:21	Other	5.148.x	GB	Kensington	709.6	Normal	Windows 8.1;outlook.exe(Tablet PC)
8/26/2015 7:34	Other	5.148.x	GB	Kensington	709.6	Normal	Windows 8;excel.exe(Tablet PC)

DEMO

Azure Active Directory anomaly detection

Summary and Next Steps

Rules versus Machine Learning

RULES

- Use when you know *exactly* what known-bad looks like
  - IOCs,
  - Signatures
  - Known-bad techniques
- Use when your detection strategy is atomic
  e.g: Look for xp_cmdshell in SQL logs

Rules decay quickly over time

MACHINE LEARNING

- Helps identify bad activity when simple heuristics fail
- Must have sufficient historical datasets, including labeled attack data
- Requires security experts who can provide feedback on quality of results
- Use when detection strategy is computation and behavioral
  e.g. Detecting unusual processes running on a host

ML systems, when periodically retrained, do not decay over time

Many solutions will incorporate *both* rules & ML

Security ML requirements

Machine Learning expertise to think beyond standard toolkits

Data across the stack

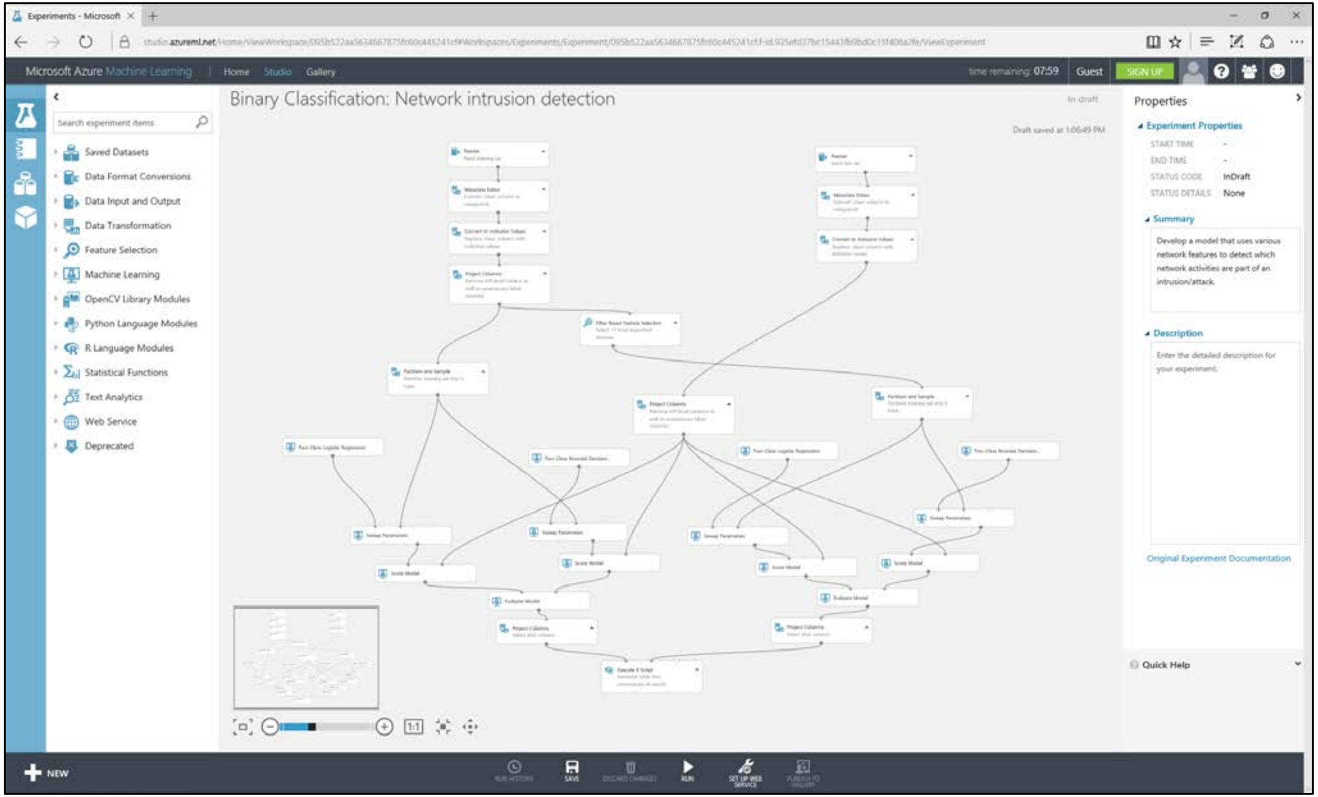
- Host (Event logs, syslog, AV logs)
- Network logs
- Service & application logs

Secure and scalable platform

Eyes on glass

Testing with real attack data

Azure Machine Learning



Next steps

Next week	Next month	In 3-6 months
DO Tinker around with this ML Network Intrusion Detection system! **EXPLORE** ML as a service and security-as-a-service solutions	Plan and collect logs from all layers: host, network, service & application	Develop an architecture to collect high-quality attack data Make it a habit to identify and investigate security anomalies

Summary

Recall ML for security improves	Interpretability	Actionability	Burden of triage
Keys to successful detection	Data is key	Secure and scalable platform	Specialized investment beyond standard machine learning toolkits

It is important to establish the credibility of the system by testing its simulated adversaries

Mark Russinovich
CTO, Microsoft Azure
markruss@microsoft.com
@markrussinovich