How to Spot a Fake

Improve Your Security Operations with Real-world AI

Stephan Jou
CTO, Interset
Hey, I’m Stephan. I like Analytics.

- CTO at Interset, a security analytics company
- Previously: Cognos and IBM’s Business Analytics CTO Office
- Big data analytics, visualization, cloud, predictive analytics, data mining, neural networks, mobile, dash-boarding and semantic search
- M.Sc. in Computational Neuroscience and Biomedical Engineering, and a dual B.Sc. in Computer Science and Human Physiology, all from the University of Toronto
AI and Machine Learning is everywhere

I’m smart!
The best Bayesian!
Buy me!
Super machine learning!
I do it all!
So, this is **not** AI

if the mail is from the departing insider and the message was sent in the last 30 days and the recipient is not in the organization’s domain and the total bytes summed by day are more than a specified threshold then send an alert to the security operator

A Pattern for Increased Monitoring for Intellectual Property Theft by Departing Insiders, Andrew Moore, Carnegie Mellon 2011
Let’s talk about...

- Anomaly detection based approaches for security-related use cases
- Data selection and feature engineering
- Model and algorithm selection
- Mathematical and technical architecture
- People, process and technology
Motivating Examples
Example #1: Chip Manufacturer

- 2 Engineers stole data
- $1 Million Spent Large security vendor failed to find anything
- Easily identified the 2 Engineers
- Found 3 additional users stealing data in North America
- Found 8 additional users stealing data in China

Unusual day/time for source code access
Accessing an unusual source code project for this user
Taking an unusually large number of source code files
Checking out more files than expected
Example #2: Defense Contractor

High Probability Anomalous Behavior Models
- Detected large copies to the portable hard drive, at an unusual time of day
- Bayesian models to measure and detect highly improbable events

High Risk File Models
- Detected high risk files, including PowerPoints used to collect large amounts of inappropriate content
- Risk aggregation based on suspicious behaviors and unusual derivative movement

Coming in on an unusual day
Accessing a set of file shares unusual for him
Accessing a set of file shares unusual for his peers
Copying an unusual number of files to a USB hard drive
Example #3: ‘Fileless’ DNS Messenger

**Stage 1:**
Malicious Word Document
- Delivered to victim via phishing email
- Appears to be ‘secured’ by a trusted security vendor
- Observed anti-virus detection was low 48 out of 64 vendors failed to detect

**Stage 2:**
Persistence and PowerShell
- PowerShell spawned by Word
- Persistence established by modifying registry
- Persistence established via ‘onidle’ scheduled task
- On-going execution further modifies scheduled task

**Stage 3/4:**
PowerShell and DNS
- PowerShell queries C&C domains for command
- PowerShell queries C&C domains for Stage 4 payload
- Stage 4 payload executed using cmd.exe

Unusual parent/child process
Anomalous DNS frequency
Unusual DNS time of day
Unusual DNS payload
Unusual and rare process
Rare process time execution
Step 1: Starting with humans
Humans inspire the use case, data and math...

- What’s the use case?
- Has this happened before?
- What data and clues did we have?
- How can we detect this in the future?
- How can we reduce any noise?
A lot of useful human expertise...

- In-house Hackers and Historians
- CERT Insider Threat
- MITRE ATT&CK grid
- Mandiant APT Attack Lifecycle
- Lockheed Martin Cyber Kill Chain
- Security community
- ... and more!

https://attack.mitre.org
Step 2: Looking at the data
Mandiant APT Attack Lifecycle

- Initial Recon
- Initial Compromise
- Establish Foothold
- Escalate Privileges
- Access Targets
- Move Laterally
- Stage Data
- Internal Recon
- Maintain Presence
- Complete Mission
Standard Approach: Active Directory and Firewall

Identity Store:
Unusual login behavior
Geo-velocity

Network:
High risk exfiltrations
Better Data = Faster and Better Detection

**Data Repositories:**
- Unusual or dangerous access
- Unusual lateral movement
- Dangerous behavioral changes

**Endpoint:**
- Unusual application

**Network:**
- Unusual IP Address
- Unusual payload size

**Identity & Access:**
- Unusual login behavior
- Unusual failed or successful access
- Geo-velocity

**Network:**
- Unusual VPN location
- Unusual remote access

**Endpoint:**
- Unusual command line tools
- Unusual registry reads

**Identity & Access:**
- Unusual service account activity
- Bot-like activity

**Identity Store:**
- Unusual machine/server access

**Endpoint:**
- Unusual data transfers or access
- Large local data storage

**Identity & Access:**
- Unusual machine logins
- Unusual share access
- Unusual Linux PAM activity

**Endpoint:**
- Unusual network share access

**Network:**
- High risk exfiltration
- Unusual network protocol

**Identity & Access:**
- Unusual print activity

**Endpoint:**
- Unusual web, USB or print activity
### Better data trumps better math

#### Endpoint
- CrowdStrike events
- Third-party DLP logs
- Windows Event logs
- Interset sensor events

#### Network
- Web proxy logs
- VPN logs
- NetFlow logs
- Firewall logs

#### Authentication
- Active Directory
- IAM logs
- Linux auditd logs
- Application access logs

#### Repository
- Source code logs
- Database logs
- Enterprise application logs
- Badge access logs
Remember these guys?

- 2 Engineers stole data
- $1 Million Spent Large security vendor failed to find anything
- Easily identified the 2 Engineers
- Found 3 additional users stealing data in North America
- Found 8 additional users stealing data in China
### The Data: Source Code Access Logs

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Engineer Account</th>
<th>IP Address</th>
<th>Action</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014/02/12 04:33:10</td>
<td>95a277da18f628a3bc9553eafdf82fe126a0ff45ea7d89895a38b18ed78</td>
<td>10.43.32.127/10.222.224.96</td>
<td>diff</td>
<td>279fe92803963ed2938b3ba9e76219d/1e4f0f1fa86a9f9b66b0c45c08b1bd8</td>
</tr>
<tr>
<td>2014/02/12 04:33:10</td>
<td>95a277da18f628a3bc9553eafdf82fe126a0ff45ea7d89895a38b18ed78</td>
<td>10.43.32.127/10.222.224.96</td>
<td>sync</td>
<td>279fe92803963ed2938b3ba9e76219d/1e4f0f1fa86a9f9b66b0c45c08b1bd8</td>
</tr>
<tr>
<td>2014/02/12 04:33:10</td>
<td>95a277da18f628a3bc9553eafdf82fe126a0ff45ea7d89895a38b18ed78</td>
<td>10.43.32.127/10.222.224.96</td>
<td>print</td>
<td>279fe92803963ed2938b3ba9e76219d/1e4f0f1fa86a9f9b66b0c45c08b1bd8</td>
</tr>
<tr>
<td>2014/02/12 04:33:10</td>
<td>95a277da18f628a3bc9553eafdf82fe126a0ff45ea7d89895a38b18ed78</td>
<td>10.43.32.127/10.222.224.96</td>
<td>difff</td>
<td>279fe92803963ed2938b3ba9e76219d/1e4f0f1fa86a9f9b66b0c45c08b1bd8</td>
</tr>
</tbody>
</table>

© 2018 Interset Software
Step 3: Building the math
There is no single best algorithm

Find the Right Tool For the Job...
First, observe and quantify normal behavior
Second, subtract current behavior from normal behavior
So, this is just curve fitting...

- **Modeling**: Pick a model that describes normal behavior
  - Probability density function
- **Learning**: Fit observations to a model of normal behavior
  - Usually per-user, per-machine, per-entity, ...
- **Scoring**: Compute distance function between current and normal
  - Euclidean, Bayes, String, ...
35 source code anomaly models

<table>
<thead>
<tr>
<th>Sneaking</th>
<th>Hoarding</th>
<th>Wandering</th>
<th>Mooching</th>
</tr>
</thead>
<tbody>
<tr>
<td>Models to look for the user doing something</td>
<td>Models to see if a user is taking</td>
<td>Models to quantify when a user accesses an</td>
<td>Models to detect dangerous changes in how</td>
</tr>
<tr>
<td>out of pattern, unusual time</td>
<td>more than expected</td>
<td>unexpected source code folder/area</td>
<td>much a user check-ins versus checks-out</td>
</tr>
</tbody>
</table>

© 2018 Interset Software
Model: Unusual time for a user

- Monitor, for each user, activity of interest (e.g. start times of when a file or window is brought into focus)
- Active times used as input into Gaussian kernel density estimators
- Times that contain 95% of activity deemed to be “normal”
- P(y is bad) at a given time is ratio of expected activity to 95% activity line
- Adjusting KDE bandwidth for different time scales (e.g. hourly versus daily versus weekly)
Model: Accessing an unusual source code area

- **Nodes**: source code projects
- **Edges**: user accessing projects
- **Edge length**: probability of project access

- Louvain clustering surfaced four large software groups or teams
- Access between team clusters is quantifiably abnormal
Third, assemble multiple anomalies together to find Waldo

- ... and moves a significantly high volume of data than normal for her
- ... and takes from a folder an unusual number of times for that folder
- ... and accesses repositories that she and her peers do not usually access
- ... VPN’s in from China, unusual for her
- Ann Funderburk works at an unusual hour for her
A mathematical architecture

Data
- Repository Logs
  - Ann moves a significant volume of data
  - Ann access and takes from file folders (other features)
- Active Directory Logs
  - Ann accesses anomalous repositories (other features)
- VPN Logs
  - Ann logs in from anomalous location
  - Ann logs in at unusual time of day (other features)

Feature Extraction

Anomaly Detection
- Volumetric Models
- File Access & Usage Models
- Auth./Access Anomaly Model
- VPN Anomaly Models

Entity Risk Aggregation

Entities
- Account
- Machine
- File
- Application
People, Process and Technology
Six personas of a cybersecurity data science team

- **The Hacker** has done it and seen it all
  - Security experts
  - Pen testers
  - Whitehats

- **Historians** bring subject matter expertise
  - SOC analysts
  - Threat hunters
  - Forensic investigators
Six personas of a cybersecurity data science team

- **Coder** wrangles data, parses records and writes code
  - R, Python, bash
  - Turn messy data into tidy data for analysis

- **Visualizers** build visual insights
  - Accessible visualizations for trends and patterns
  - Timeline plots
Six personas of a cybersecurity data science team

- **Modelers** turns “unusual” into math
  - Turn words into statistics
  - Models generalize in your environment

- **Storytellers** connect data to models to results to threats
  - Explains models and results
  - Drives understanding from the SOC analyst to the Board
Process pipeline from Use Case to Production

Use Case
- Acquisition
- Cleanup
- Normalization
- Semantics

Data
- Feature engineering
- Model design
- Model validation

Exploratory Data Analysis

Production Deployment
- Object model
- Data ingest
- Model development
- Test and performance

Results
Technology for Exploratory Data Analysis

- 30 days of historical log data from a new data source
- Interview and validate with stakeholders
- Validate or tune existing models
- Design and apply new models

- **R and R Studio**
  - To design, test and visualize models

- **R Markdown**
  - To document and version models
Technology for Production Deployment

- Update ingest and object model
- R to Spark and Phoenix code
- Models to batch and real-time
- Model efficacy
- Scalability and performance

- **NiFi, Flume and Kafka**
  - To move data around
- **Spark, Phoenix and HBase**
  - To learn and score models at scale
  - Uses Scala, Java and SQL
- **Elasticsearch and Kibana**
  - To store and search results
  - Integrates with ELK
Interset’s Behavioral Analytics Engine – by the Numbers

- **100%** unsupervised, online machine learning
- **450** threat detection algorithms, and growing
- **13** data types analyzed
- **100** person years of development, hardening and refinement
- **5** years of security analytics in the market
- **3** years as an In-Q-Tel portfolio company

**FIND THE THREATS THAT MATTER**

Distill billions of events into a handful of actionable leads