Finding Hijacked Accounts

Anomaly Detection in User Behavior Analysis

László Kovács
laszlo.kovacs@balabit.com

I'm a Data Scientist

What my colleagues think I do

What my family thinks I do

What my boss thinks I do

What society thinks I do

What professional programmers or statisticians think I do

stack overflow

What I actually do

Machine learning effectively finds hijacked accounts beyond the limits of rule-based security

Limits of SIEM systems

- Strict rules
- Changing environment
- B.Y.O.D.
- Sophisticated threats
- Variety of attacks

Problems with investigation

- Data breach is hard to notice
- Investigation and drawing conclusion is time-consuming
- Log messages are noisy and unstructured

Flexible, unsupervised user behavior analysis can provide means of solution

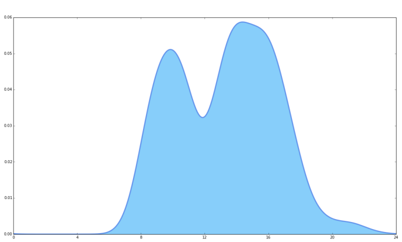
Unsupervised machine learning applied to a „label-less" problem

No examples to train on	Create models for the usual	Find the outliers and measure the anomalousness
• Limited knowledge about the attacks • Few well documented examples (not representative) • Custom-tailored attacks	• Assume that the bulk of a user's behavior is harmless, and normal • The normal behavior can be modelled in an unsupervised fashion	• Once the model is trained we can investigate the actions of the user • Every new action can be compared against the model • Outlier-ness can be objectively defined

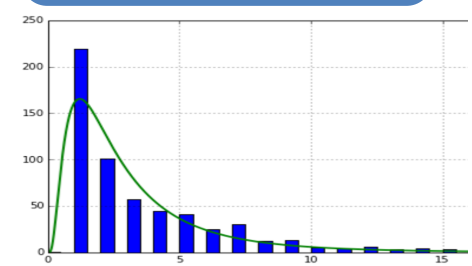
Combining several tools to inspect different aspects of the user activities

Algorithms for detecting anomalies

Analysis of one dimension		Analysis of multiple dimensions	Analysis of aggregated data
Time distribution modelling	Recommended hosts, based on peers	Customer basket analysis	Spotting unusual amounts of activities



	Host1	Host2	Host3	Host4	Host5
User1	1	1	0	1	1
User2	0	1	0	1	1
User3	1	1	1	1	0
User4	1	1	0	0	0
User5	1	1	0	0	1



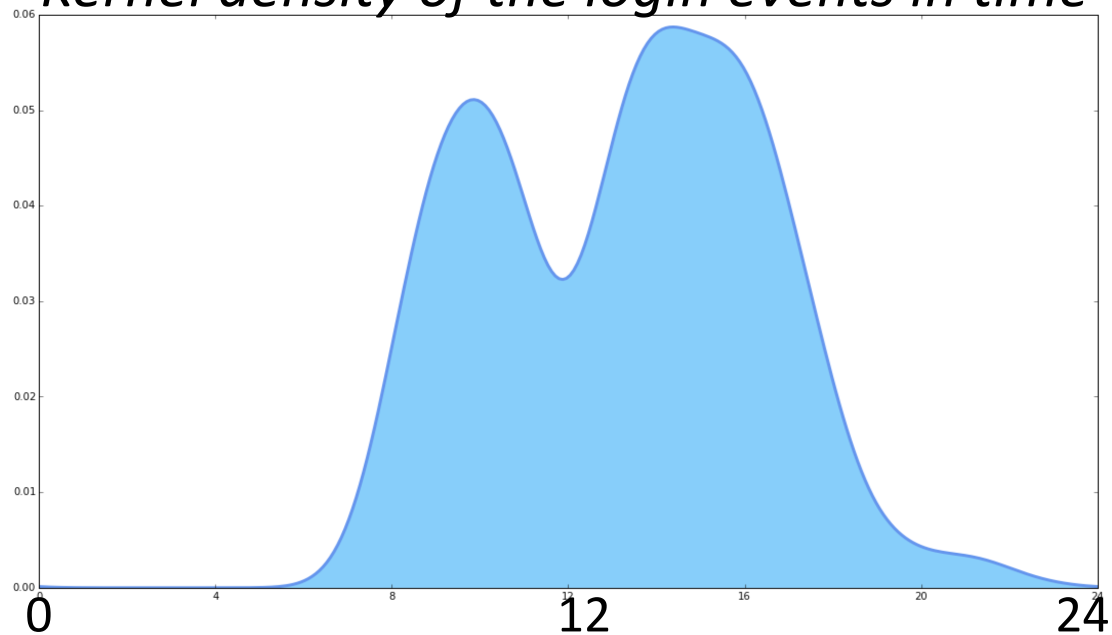
Complexity

Unusual log-in times can signal anomalous behaviour

The most obvious anomaly:

Somebody works when she does not work usually

Kernel density of the login events in time



- Easy to build a model based on the past
- Easy to measure the anomalousness of an event
- Easy to interpret the results

Non-recommended servers can point out suspicious activity

	Host1	Host2	Host3	Host4	Host5
User1	1	1	0	1	1
User2	0	1	0	1	1
User3	1	1	1	1	0
User4	1	1	0	0	0
User5	1	1	0	0	1

Amazon recommends products based on ones purchased items and the peers' transactions.

This approach can be used to calculate the unexpected-ness of a new connection.

No prior knowledge needed about the servers/ users.

The less recommended a server is
the more unexpected the activity will be.

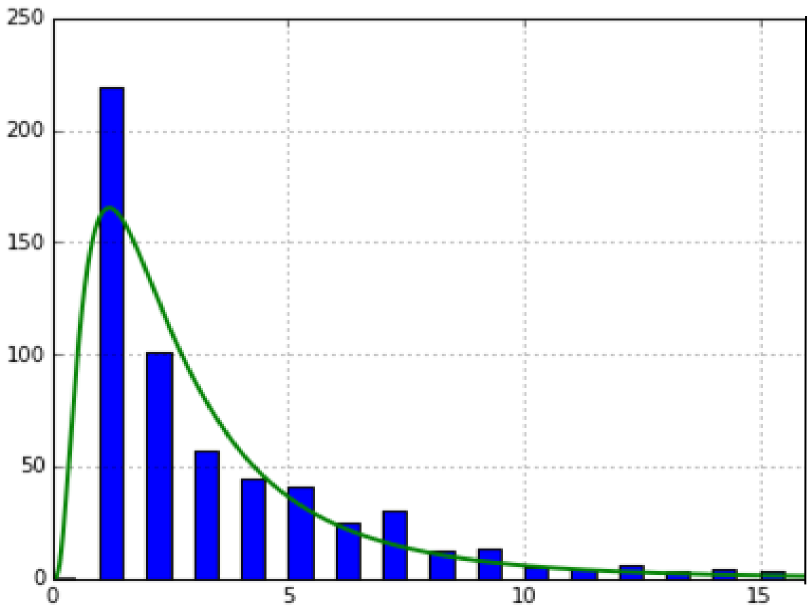
Basket analysis reveals othervise hidden anomalies

	Retail	Security
Data	Transactions	Log-in events
Entity	Set of products	Set of log-in metadata
Codified model	Common co-purchases	Common patterns
Gained insight	Taste of the customer	Structure of activities
Example finding	„Beer implies chips in 30% of the cases”	„SSH implies „weekends” in 10% of the cases”

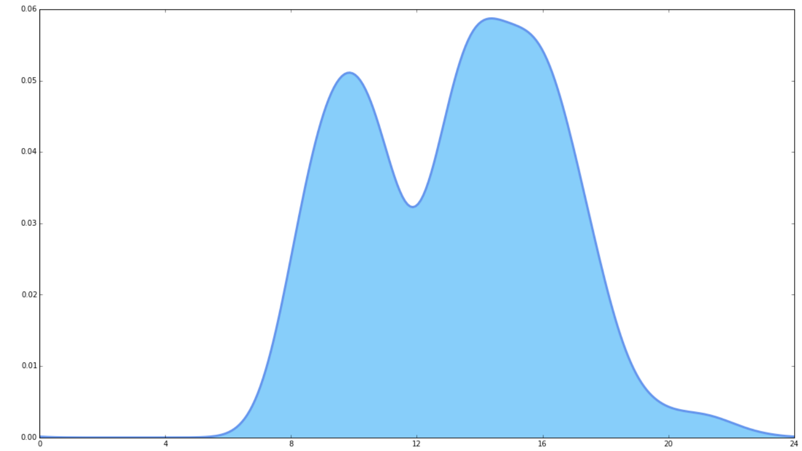
Frequency analysis of usual events can reveal anomalous behaviour

Unusually many events are important clues!

Models are made to represent the distribution of the aggregates.



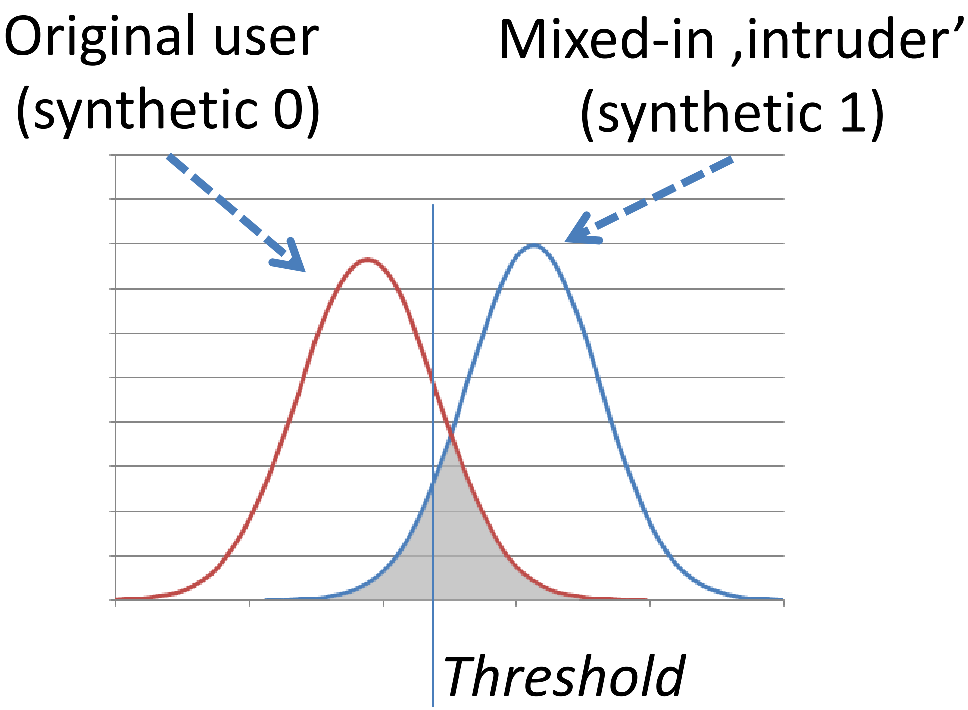
Histogram of time-based aggregates

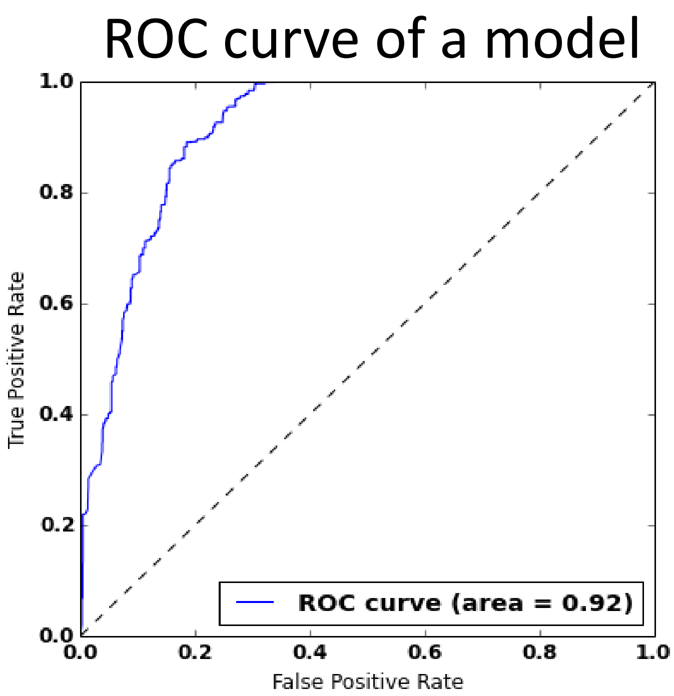


The data is not evenly distributed, but we can use the log-in time curve to **estimate the expected number of events** for any given time.

The problem of measurement can be tackled by synthetic labels

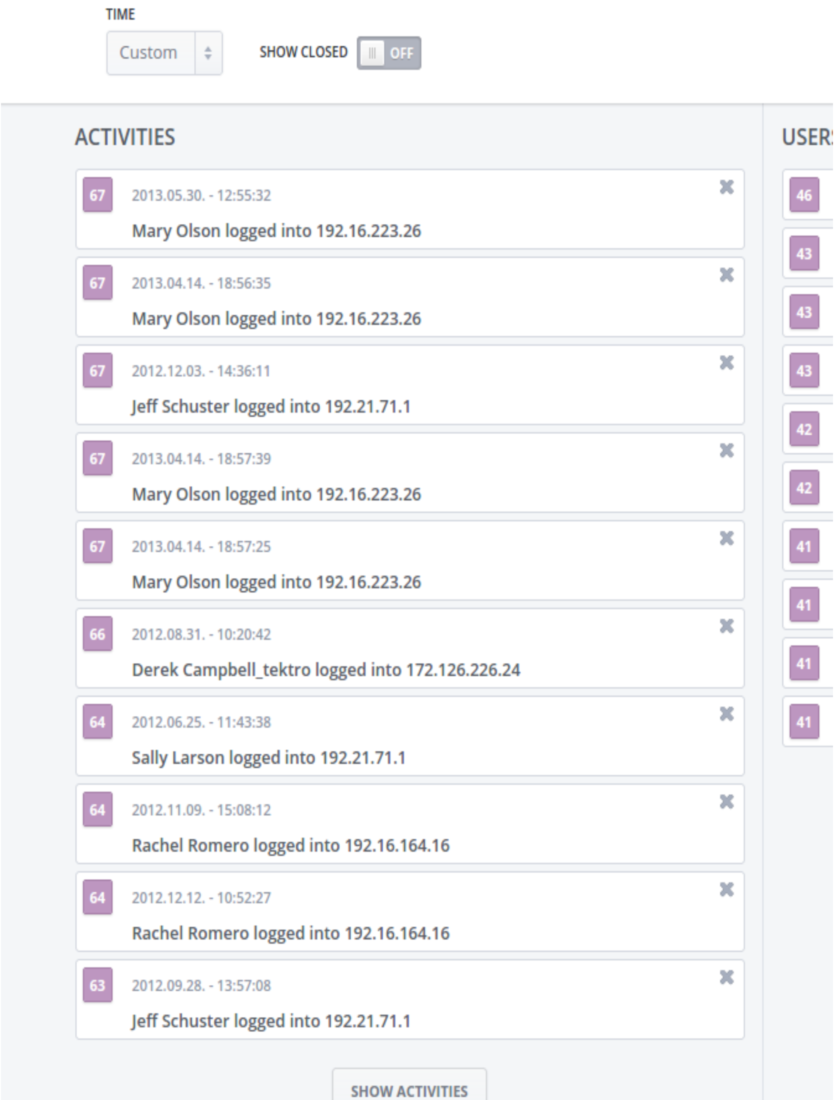
Build a baseline for 1 user, and mix in different user-activites before scoring!





Every threshold corresponds with a different False Positive Rate and False Negative Rate.

Priority ordering enhances the effectiveness of investigation



By tagging every activity with the score it gets form the combination of the algorithms, one does not have to decide on thresholds.

The algorithms can be fine-tuned by the response of the security professional.

Thank you for your attention!