Secure all the things with graphs and predictive analytics

Kevin Mahaffey
CTO
Lookout

Tim Wyatt
Director, Security Engineering
Lookout
Over-simplification of security

Protection
bars on windows

Detection
alarm system

Remediation
locksmith to change locks

Attribution
whodunit

Response
arrest/prosecution
Detection

predict whether this thing in front of me is good or bad given what I can observe and my knowledge
Examples

Anti-malware
Fraud detection
Insider threat
DLP
IPS/IDS
expert-system (e.g. signatures, heuristics)
ML-based (e.g. random forest, bayesian)
stateful models (e.g. anomaly detection)

Science!

Input

binary
transaction data
identity
click data
behavior

Model

Output

score
classification
decision
Signature Expert-system

If something matches a known pattern, match.

- Hash of file content
- Contains string “This v1ru5 cr3@t3d by 1337kr3w.ru”
- At offset 0x12213, contains sequence “feab34fe0cdba30700”
- Length >1024 in SMB protocol field

**Good:** Fast, exactly match threat

**Bad:** Easy to evade (brittle knowledge)

- The burglar always a car with license plate DL324ABX
Heuristic Expert-system

If it fits this pattern, match.
Like a signature, but inexact
Signature with wildcards

Good: Detect multiple variants of a threat
Bad: Easy to evade, may have false-positives
Behavioral Expert-system

If engages in a particular behavior, match.
   Calls 10 sys calls in with particular order with particular args
   Opens /dev/kmem
   Attempts to modify /usr/bin/sudo
   PDF being opened results in new process spawned
   Silvertail example

Good: gets to root of problems

Bad: many things are used for good and evil, you can’t always measure where you want to (on endpoints) and are forced to measure where you don’t (on an email gateway)
Anomaly-detection

If it’s weird, match.

Good: great at finding new threats

Bad: great at waking IT admins up at night
All is not well.
Problem #1: TANSTAAFL
There ain’t no such thing as a free lunch.
Choose one:

- too many false positives
- too many false negatives
Systems that minimize false negatives tend to increase false positives.
Hackers in China Attacked The Times for Last 4 Months


By NICOLE PERLOTH
Published: January 30, 2013   391 Comments
Over the course of three months, attackers installed **45 pieces of custom malware**. The Times – which uses antivirus products made by [redacted] – found **only one instance** in which [redacted] **identified** an attacker’s software as malicious and quarantined it...
Over the course of three months, attackers installed 45 pieces of custom malware. The Times – which uses antivirus products made by [redacted] – found only one instance in which [redacted] identified an attacker’s software as malicious and quarantined it...

"Systems that minimize false positives tend to increase false negatives."
In high volume classification systems, particularly those powered by human analysts, absolute accuracy is hard.
Problem #2:
Hidden Relationships

How do you spot relationships that are important ahead of an event (to predict) or after an event (to attribute).
u@server:~$ cat /log/access.log | grep 23.11.541.66
<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Event Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tue Nov 22 08:53:25</td>
<td>Teardown UDP connection 144744478313156395 for DET-SEC-12.19:10.124.19.12:45091 to OUTSIDE:10.68.15.11:53 duration 0.807s</td>
</tr>
<tr>
<td>Tue Nov 22 08:53:25</td>
<td>Teardown UDP connection 144744478313156396 for DET-SEC-12.19:10.124.19.12:52757 to OUTSIDE:10.68.15.11:53 duration 0.807s</td>
</tr>
<tr>
<td>Tue Nov 22 08:53:26</td>
<td>Teardown UDP connection 144744478313156397 for DET-SEC-12.19:10.124.19.12:47309 to OUTSIDE:10.68.15.11:53 duration 0.807s</td>
</tr>
<tr>
<td>Tue Nov 22 08:53:26</td>
<td>Teardown UDP connection 144744478313156398 for DET-SEC-12.19:10.124.19.12:52485 to OUTSIDE:10.68.15.11:53 duration 0.807s</td>
</tr>
<tr>
<td>Tue Nov 22 08:54:20</td>
<td>Teardown UDP connection 144744478313156408 for DET-SEC-12.19:10.124.19.12:35728 to OUTSIDE:10.68.15.11:53 duration 0.807s</td>
</tr>
<tr>
<td>Tue Nov 22 08:54:20</td>
<td>Teardown UDP connection 144744478313156409 for DET-SEC-12.19:10.124.19.12:43103 to OUTSIDE:10.68.15.11:53 duration 0.807s</td>
</tr>
<tr>
<td>Tue Nov 22 08:54:20</td>
<td>Teardown UDP connection 144744478313156410 for DET-SEC-12.19:10.124.19.12:51752 to OUTSIDE:10.68.15.11:53 duration 0.807s</td>
</tr>
</tbody>
</table>
Those really were the droids you were looking for.
What changed recently?
Scalable software (Hadoop)
Scalable storage
Scalable compute
Machine learning knowledge spreading

Capability to do whole-population analytics, association, and data-driven response.
We can use **machines** to identify more complex signals and relationships in **big datasets** than humans can.
The Goal

Use **machine learning** on top of **big datasets** to **predict the future** and build better detection systems.

Use **graphs** to **add context** to **steaming piles of data** to help in detection and attribution.
Core Techniques

Supervised ML
Unsupervised ML
Precise relationships
Fuzzy relationships
Supervised ML
Supervised ML Model I/O

**Inputs**: “features” of something you’re trying to make a prediction about.

**Outputs**: the prediction you’re trying to make.

**Function of model**: transform some inputs to some outputs based on a model.

**Training**: take known inputs and outputs and make a model that transforms input to output.
Supervised ML Model

Input — Science! — Output
Feature extraction

Input

2 12 29 32 4
16 98 13 6 99
21 42 45 63 81
90 1 77 23 69
96 75 10 83 78
73 92 48 29 71
2 8 12 9 22
33 37 88 31 93
94 27 11 30 80

Model

Output

92.31 Risk Score
Training Data

Feed training data into model

Model #1

Calculate goodness of model

Fitness Function

Model 1 Fitness Score
Model #1

Mutate model

Model #2

Model 1
Fitness Score
Training Data → Model #2
Feed training data into model → Calculate goodness of model → Fitness Function → Model #1 Fitness Score
Model 2 Fitness Score
Choose best model

Model 1 Fitness Score

Mutate model

Model #3
Repeat many many many many many times.
Example Supervised ML Systems

Naive Bayes
Neural Networks
Random Forest
Support Vector Machines (SVM)
Adaptive Standard Additive Model fuzzy system (ASAM)
Building an ML System
Feature engineering is the art behind the science

1. Pre-requisite: Understand the field
2. Identify candidate features
3. Build tools to extract them

Examples:
- permissions
- behaviors (active and passive)
- anomalous characteristics
Data Scrubbing

Dirty data trains bad / corrupts good models

android.

pre

mission.GET_ACCOUNTS !=

android.

per

mission.GET_ACCOUNTS

Cleanse, normalize, debug for best behavior

android.

pre

mission.GET_ACCOUNTS !=

android.

per

mission.GET_ACCOUNTS
Example Training Set

- **Good Training Set**
  - App
  - App
  - App
  - App

- **Bad Training Set**
  - App
  - App
  - App
  - App

- **Extract Permissions**
  - true
  - false
  - false
  - ...
  - bad

- **Vectorized Features**

- **Random Forest Trainer**
Example Process (Live Classification)

1. **Incoming App**
2. **Extract Permissions**
3. **Vectorized Features**
   - true
   - false
   - false
   - ...
   - bad
4. **Random Forest Classifier**
5. **Good**
6. **Bad**
### Performance

#### Evaluation result

- Scheme: RF - Partitioned: RandomForest
- Options: `-I 1000 -K 21 -S 1 -num-slots 1`
- Relation: relation

<table>
<thead>
<tr>
<th>Correctly Classified Instances</th>
<th>16892</th>
<th>84.4558 %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incorrectly Classified Instances</td>
<td>3109</td>
<td>15.5442 %</td>
</tr>
<tr>
<td>Kappa statistic</td>
<td>0.6889</td>
<td></td>
</tr>
<tr>
<td>Mean absolute error</td>
<td>0.2123</td>
<td></td>
</tr>
<tr>
<td>Root mean squared error</td>
<td>0.3447</td>
<td></td>
</tr>
<tr>
<td>Relative absolute error</td>
<td>42.4308 %</td>
<td></td>
</tr>
<tr>
<td>Root relative squared error</td>
<td>68.2154 %</td>
<td></td>
</tr>
<tr>
<td>Coverage of cases (0.95 level)</td>
<td>96.7802 %</td>
<td></td>
</tr>
<tr>
<td>Mean rel. region size (0.95 level)</td>
<td>76.0537 %</td>
<td></td>
</tr>
<tr>
<td>Total Number of Instances</td>
<td>20001</td>
<td></td>
</tr>
</tbody>
</table>

#### Detailed Accuracy By Class

<table>
<thead>
<tr>
<th>TP Rate</th>
<th>FP Rate</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
<th>MCC</th>
<th>ROC Area</th>
<th>PRC Area</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.749</td>
<td>0.061</td>
<td>0.925</td>
<td>0.749</td>
<td>0.828</td>
<td>0.702</td>
<td>0.951</td>
<td>0.928</td>
<td>clean</td>
</tr>
<tr>
<td>0.939</td>
<td>0.251</td>
<td>0.790</td>
<td>0.939</td>
<td>0.858</td>
<td>0.702</td>
<td>0.951</td>
<td>0.962</td>
<td>dirty</td>
</tr>
</tbody>
</table>

| Weighted Avg. | 0.845 | 0.156 | 0.857 | 0.845 | 0.843 | 0.702 | 0.951 | 0.945 |

#### Confusion Matrix

```
<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>--- classified as</th>
</tr>
</thead>
<tbody>
<tr>
<td>7466 2502</td>
<td>a = clean</td>
<td></td>
</tr>
<tr>
<td>607 9426</td>
<td>b = dirty</td>
<td></td>
</tr>
</tbody>
</table>
```
### Evaluation Result ###

Scheme: RF - Partitioned: RandomForest
Options: -I 1000 -K 21 -S 1 -num-slots 1
Relation: relation

<table>
<thead>
<tr>
<th>Correctly Classified Instances</th>
<th>16892</th>
<th>84.4558 %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incorrectly Classified Instances</td>
<td>3109</td>
<td>15.5442 %</td>
</tr>
</tbody>
</table>

- Kappa statistic: 0.6889
- Mean absolute error: 0.2123
- Root mean squared error: 0.3447
- Relative absolute error: 42.4308 %
- Root relative squared error: 68.2154 %
- Coverage of cases (0.95 level): 96.7802 %
- Mean rel. region size (0.95 level): 76.0537 %
- Total Number of Instances: 20001

### Detailed Accuracy By Class ###

<table>
<thead>
<tr>
<th>TP Rate</th>
<th>FP Rate</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
<th>MCC</th>
<th>ROC Area</th>
<th>PRC Area</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.749</td>
<td>0.061</td>
<td>0.925</td>
<td>0.749</td>
<td>0.828</td>
<td>0.702</td>
<td>0.951</td>
<td>0.928</td>
<td>clean</td>
</tr>
<tr>
<td>0.939</td>
<td>0.251</td>
<td>0.790</td>
<td>0.939</td>
<td>0.858</td>
<td>0.702</td>
<td>0.951</td>
<td>0.962</td>
<td>dirty</td>
</tr>
<tr>
<td>Weighted Avg.</td>
<td>0.845</td>
<td>0.156</td>
<td>0.857</td>
<td>0.845</td>
<td>0.843</td>
<td>0.702</td>
<td>0.951</td>
<td>0.945</td>
</tr>
</tbody>
</table>

### Confusion Matrix ###

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>classified as</th>
</tr>
</thead>
<tbody>
<tr>
<td>7466</td>
<td>2502</td>
<td>a = clean</td>
</tr>
<tr>
<td>607</td>
<td>9426</td>
<td>b = dirty</td>
</tr>
</tbody>
</table>
### Evaluation result ###

Scheme: RF - Partitioned: RandomForest
Options: -I 1000 -K 21 -S 1 -num-slots 1
Relation: relation

<table>
<thead>
<tr>
<th>Correctly Classified Instances</th>
<th>16892</th>
<th>84.4558 %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incorrectly Classified Instances</td>
<td>3109</td>
<td>15.5442 %</td>
</tr>
<tr>
<td>Kappa statistic</td>
<td>0.6889</td>
<td></td>
</tr>
<tr>
<td>Mean absolute error</td>
<td>0.2123</td>
<td></td>
</tr>
<tr>
<td>Root mean squared error</td>
<td>0.3447</td>
<td></td>
</tr>
<tr>
<td>Relative absolute error</td>
<td>42.4308 %</td>
<td></td>
</tr>
<tr>
<td>Root relative squared error</td>
<td>68.2154 %</td>
<td></td>
</tr>
<tr>
<td>Coverage of cases (0.95 level)</td>
<td>96.7802 %</td>
<td></td>
</tr>
<tr>
<td>Mean rel. region size (0.95 level)</td>
<td>76.0537 %</td>
<td></td>
</tr>
<tr>
<td>Total Number of Instances</td>
<td>20001</td>
<td></td>
</tr>
</tbody>
</table>

### Detailed Accuracy By Class ###

<table>
<thead>
<tr>
<th>Class</th>
<th>TP Rate</th>
<th>FP Rate</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
<th>MCC</th>
<th>ROC Area</th>
<th>PRC Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>clean</td>
<td>0.749</td>
<td>0.061</td>
<td>0.925</td>
<td>0.749</td>
<td>0.828</td>
<td>0.702</td>
<td>0.951</td>
<td>0.928</td>
</tr>
<tr>
<td>dirty</td>
<td>0.939</td>
<td>0.251</td>
<td>0.790</td>
<td>0.939</td>
<td>0.858</td>
<td>0.702</td>
<td>0.951</td>
<td>0.962</td>
</tr>
<tr>
<td>Weighted Avg.</td>
<td>0.845</td>
<td>0.156</td>
<td>0.857</td>
<td>0.845</td>
<td>0.843</td>
<td>0.702</td>
<td>0.951</td>
<td>0.945</td>
</tr>
</tbody>
</table>

### Confusion Matrix ###

\[
\begin{array}{cc}
7466 & 2502 \\
607 & 9426 \\
\end{array}
\]

--- classified as

\[
\begin{array}{c}
a = \text{clean} \\
b = \text{dirty} \\
\end{array}
\]
Evaluation result

Scheme: RF - Partitioned: RandomForest
Options: -I 1000 -K 21 -S 1 -num-slots 1
Relation: relation

Correctly Classified Instances 16892 84.4558%
Incorrectly Classified Instances 3109 15.5442%
Kappa statistic 0.6889
Mean absolute error 0.2123
Root mean squared error 0.3447
Relative absolute error 42.4308%
Root relative squared error 68.2154%
Coverage of cases (0.95 level) 96.7802%
Mean rel. region size (0.95 level) 76.0537%
Total Number of Instances 20001

Detailed Accuracy By Class

<table>
<thead>
<tr>
<th>Class</th>
<th>TP Rate</th>
<th>FP Rate</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
<th>MCC</th>
<th>ROC Area</th>
<th>PRC Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>clean</td>
<td>0.749</td>
<td>0.061</td>
<td>0.925</td>
<td>0.749</td>
<td>0.828</td>
<td>0.702</td>
<td>0.951</td>
<td>0.928</td>
</tr>
<tr>
<td>dirty</td>
<td>0.939</td>
<td>0.251</td>
<td>0.790</td>
<td>0.939</td>
<td>0.858</td>
<td>0.702</td>
<td>0.951</td>
<td>0.962</td>
</tr>
<tr>
<td>Weighted Avg.</td>
<td>0.845</td>
<td>0.156</td>
<td>0.857</td>
<td>0.845</td>
<td>0.843</td>
<td>0.702</td>
<td>0.951</td>
<td>0.945</td>
</tr>
</tbody>
</table>

Confusion Matrix

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>7466</td>
<td>2502</td>
</tr>
<tr>
<td>b</td>
<td>607</td>
<td>9426</td>
</tr>
</tbody>
</table>

a = classified as clean
b = classified as dirty
True Positive Rate vs. False Positive Rate graph.

- Better performance is indicated by points above the diagonal line.
- Worse performance is indicated by points below the diagonal line.

The graph compares the True Positive Rate on the y-axis against the False Positive Rate on the x-axis.
There’s no free lunch ...
For every 10,000 clean apps, you have 2,500 false positives :(

True Positive Rate

False Positive Rate

False Positive Rate
Permissions are a terrible way to classify malware.
ML is not magic.

It makes sense of data experts have already identified as significant.*

*deep learning may change this, but that’s another talk.
Graphs
Explicit graphs:
How do you record where clear relationships exist?
Implicit graphs: How do you identify relationships where no clear ones exist?
Implicit graphs:
Unsupervised ML
Unsupervised ML

Goal: Find hidden structure in unlabeled data.

Input: Features of subjects

Output: Relationship between subjects
Implicit graphs: Fuzzy Relationships
Similarity Engine

43%
<table>
<thead>
<tr>
<th></th>
<th>Obj 1</th>
<th>Obj 2</th>
<th>Obj 3</th>
<th>Obj 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obj 1</td>
<td>0%</td>
<td>13%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Obj 2</td>
<td>41%</td>
<td>0%</td>
<td>0%</td>
<td>31%</td>
</tr>
<tr>
<td>Obj 3</td>
<td>0%</td>
<td>0%</td>
<td>10%</td>
<td></td>
</tr>
<tr>
<td>Obj 4</td>
<td>0%</td>
<td>19%</td>
<td>71%</td>
<td></td>
</tr>
</tbody>
</table>
Explicit graphs:
Precise Relationships
Building Graphs
Why graphs?

Graph DBs faster than relational DBs for relationship queries

No long wait times for complex queries

Analysts: Quickly identify relationships that are not otherwise obvious

Detection systems: incorporate relationships into models
Key Concepts

Node: type: User
username: bob_a_plinkerton
phone_number: +1-415-555-1231

Relationship:
logged_in_from
time: Dec 29, 2013, 16:42:01Z
ip: 23.15.112.211

Properties:
type: Client
id: acd234908fade09
os: OSX
version: 10.9
browser: Chrome 33.0.1750.117
Security Modeling Example

Entities

- Application Binary
- Public Key
- Hostname
- Permission
- IP
- Autonomous System
Relational Model

- hostnames
- ips
- autonomous_system
- tcp_connections
- application_binaries
- public_keys
Graph Domain Modeling

What should be a node? relationship? property?

Entities (people, places, things, unique content) tend to be nodes.

- A user
- A client
- An IP
- A URL

Events (things that happen at a point in time) tend to create relationships.

- A user click on a link
- A user makes a transaction from a client at an IP
Graph Data Modeling

AS is a property

AppBinary 1 connects_to AS 1

count: to
ip: 24.12.44.251

IP is a node

AppBinary 1 connects_to IP 1

IP 1 member_of AS 1

Connection is a node

AppBinary 1 makes TCP Connection 1

tx: 1.2 MiB
rx: 1.5 MiB

dest IP 1

IP 1 member_of AS 1
To node or not to node?

Nodes that always only have 2 neighbors of the same types usually better as relationships with properties

Will you ever run analytics on particular relationships?
- all apps that connect to IP?
- all apps that connect to AS?

What is important for your analysts to see?
- As a consumer cloud service, you might care more about IP
- As an ISP, you might care more about AS
Graph DBs

Neo4j
Titan
Rexster
OrientDB
...many more
Neo4j Server
Programmatic Queries

Cypher

neo4j query language

Detection example: tell me when what apps connect to an IP that known malware also connects to?

```
1 match (app:`AppBinary`) -[]-(ip:`Ip`) -[]-(badapp:`AppBinary`) 
2 where app.malware = false and badapp.malware = true 
3 return app
```
match (app:`AppBinary`)-[]-(ip:`Ip`)-[]-(badapp:`AppBinary`) where app.malware = false and badapp.malware = true return app
match (app:`AppBinary`) -[ ]-(ip:`Ip`) -[ ]-(badapp:`AppBinary`) 
where app.malware = false and badapp.malware = true 
return app
1 match (app:`AppBinary`)-[[:no_edge:]]-(ip:`Ip`)-[[:no_edge:]]-(badapp:`AppBinary`) 
2 where app.malware = false and badapp.malware = true 
3 return app
1 match (app: `AppBinary`) - [] - (ip: `Ip`) - [] - (badapp: `AppBinary`) 
2 where app.malware = false and badapp.malware = true 
3 return app
1 match (app:`AppBinary`)-[[]]-(ip:`Ip`)-[[]]-(badapp:`AppBinary`) 
2 where app.malware = false and badapp.malware = true 
3 return app
Places you might want to use graph analytics

Account/transaction fraud detection
  Link accounts, IPs, clients, inter-account interactions

Forensic/realtime netflow analysis
  Who talks to whom on your network?

Cybercrime investigation
  Mapping adversary infrastructure
Network Effect:
The bigger graphs hold more predictive power.
Malware

AppBinary 2

Public Key 1

Public Key 2

Unknown App

AS 1

IP 1

AS 2

IP 4

Domain 1

Domain 2

Domain 3

IP 2

IP 3

IP 2
You can’t make predictions using data you don’t have.
Graphs + ML
Graphs + ML

Extract features from your graph to power better ML models.

Reputation: what are your relatives like?
Similarity: what other things are like you?
Big security problems need an arsenal of data tools.
A new ad network?

Ad Network SDK collects:
- Phone number
- IMEI
- Android ID

Risky server-initiated actions:
- Download + ask to install app
- Install shortcut
- View URL
- Change server address
Is it bad?
We decided to block these apps for our customers…

…even though they hadn’t “done” anything malicious.
April 15, 2013 7:03 PM “Ad Server” pushes:


Download URL is a known malware family, AlphaSMS
“Ad Server” then pushes:

```json
```

New malware that can exfiltrate contact list, send SMS, rotate C&C servers, etc.

4 classes have 43-60% similarity to “ad network” code
BadNews summary

32 apps, 4 developer accounts on Google Play

Google’s statistics: downloaded between 2-9 million times
Apply

Big datasets can help solve tough security problems
...but there’s no free lunch. They’re not magic.

Graphs + ML are a powerful way to interact with data to solve detection and attribution problems

Free + open source tools available to start experimenting