Fighting Identity Theft

Big Data Analytics to the Rescue

Seshika Fernando
WSO2
Me - Seshika

- Computer Science & Finance
- Streaming Analytics

WSO2

- 100% Open Source Middleware Company
- Apache Way
- http://wso2.com/
$2.5m per Enterprise
#1 Consumer Complaint
Every 2 seconds
51% Enterprises use Big Data Analytics

Sources: Javelin Strategy & Research, PwC 2016 GSISS, FTC 2015 Report
Service Provider

Identity Providers

User
Authentication Analytics

- Blacklisted IP address
- Single IP, multiple users
- Single user, multiple IPs
- Login from new IP address
- Abnormal frequency of logins
- Abnormal login times
- Multiple login failures
- Multifactor authentication failures

“Should I arrest Clark Kent for identity theft or should Clark Kent have me arrested for identity theft? This is all so very confusing!”
Authorization Analytics

- User/Role accessing a new resource
- Abnormal resource access frequency
- Access denied for multiple resources, for the same user
- Abnormal usage frequency of high privilege accounts
- High risk privilege escalation
Complex Event Processing

- Notify if there is a 10% increase in overall trading activity AND the average price of commodities has fallen 2% in the last 4 hours
Blacklists

```java
define table BlacklistedIPTable (ipAddress string);

from loginStream[ (ip == BlacklistedIPTable.ip) in BlacklistedIPTable ]
select *
insert into alertStream;
```

Whitelists

```java
define table IPTable (ipAddress string);

from loginStream[ not(ip == IPTable.ip) in IPTable ]
select *
insert into alertStream;
```
Counting

```sql
from loginFailureStream#window.time(1 hour)
select username, count(timestamp) as loginFailCount
group by username
having loginFailCount > 30
insert into alertStream;
```

1 to many relationships

```sql
from e1 = loginStream ->
e2 = loginStream[(e1.ip == e2.ip) and (e1.username != e2.username)] <2:>
within 1 day
select e1.ip, e1.username, e2[0].username, e2[1].username
insert into alertStream;
```
Adaptive Analytics

User Profiling (UEBA)

- Time
- IP/Geo-location
- Frequency
- Typing Patterns
- Service Provider(s)
- Identity Provider(s)

Wonka usually logs in between **8am - 10am**, from an IP address in **Chicago**, and logs into **Redmine** and **Concur**, using his **Google** Credentials.
Behavourial Rules

- Based on
  - Time
  - Login Frequency
  - Geo Location
  - List of Service Providers
  - List of IDPs

```sql
from loginStream#window.time(1 hour) as str
join loginCountTable as tbl
on str.username == tbl.username
select str.username, count(str.timestamp) as curLoginCount, tbl.maxLoginCount
group by str.username
having curLoginCount > maxLoginCount
insert into alertStream;
```
Scoring

- Use combination of rules
- Give weights to each rule
- Single number to represent suspicion through multiple indicators
- Use a threshold to identify anomalies

Score = $w_1 \times \text{time} + w_2 \times \text{frequency} + w_3 \times \text{location} + w_4 \times \text{SPs} + w_5 \times \text{IDPs}$
Clustering

Features
- Time
- Geo Location
- IdP
- SP Type
Markov Models

1. Classify Events
2. Update Probability Matrix
3. Compare Incoming Sequences

Events → Probability Matrix → Alerts
Audit Trail Analytics

<table>
<thead>
<tr>
<th>Service Provider</th>
<th>User</th>
<th>Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>PeopleHR</td>
<td>Paul</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event</th>
<th>User</th>
<th>Date</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Config Change</td>
<td>Paul</td>
<td>10/10/2016</td>
<td>10:00 am</td>
</tr>
<tr>
<td>SP Permission Added</td>
<td>Admin</td>
<td>10/10/2016</td>
<td>9:50 am</td>
</tr>
<tr>
<td>SP Permission Revoked</td>
<td>Admin</td>
<td>10/10/2016</td>
<td>10:15 am</td>
</tr>
</tbody>
</table>
Investigate

Access historical data using

- Expressive Querying
- Easy Filtering
- Useful Visualizations

to isolate incidents and unearth relationships
Deployment

IAM

Service Providers

Persisted Storage

WSO2 Complex Event Processor

Events

Alerts

Dashboard
Challenges
Unusual behaviour?
Big Data Challenge

- Millions of Events
- Highly Dimensional

<table>
<thead>
<tr>
<th>EventID</th>
<th>Timestamp</th>
<th>Auth Success</th>
<th>Username</th>
<th>Roles</th>
<th>Service Provider</th>
<th>IDP</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1420092114000</td>
<td>True</td>
<td>Norman</td>
<td>Dev; Admin</td>
<td>Expedia</td>
<td>Google</td>
<td>100.3.2.88</td>
</tr>
<tr>
<td>2</td>
<td>1420092114200</td>
<td>True</td>
<td>John</td>
<td>Dev</td>
<td>Concur</td>
<td>Facebook</td>
<td>10.13.2.15</td>
</tr>
<tr>
<td>3</td>
<td>1420092115500</td>
<td>False</td>
<td>Mary</td>
<td>QA</td>
<td>Ebay</td>
<td>Facebook</td>
<td>20.3.2.132</td>
</tr>
</tbody>
</table>

- Real-time Dashboards

Last Hour | Last 24 Hours | Last 30 Days | Last Year | 📅 Custom
Fight against Time

CEP

Spark
from AuthEventStream#window.TimeBatch(1 sec)
select sum(AuthCount), year, month, date, hour, min, sec
insert into PerSecAuthCountStream

from PerSecAuthCountStream#window.TimeBatch(1 min)
select sum(AuthCount), year, month, date, hour, min
insert into PerMinAuthCountTable

insert into PerHourAuthCountTable
select sum(AuthCount), year, month, date, hour
from PerMinAuthCountTable
group by year, month, date, hour

insert into PerDayAuthCountTable
select sum(AuthCount), year, month, date
from PerHourAuthCountTable
group by year, month, date
Battling Dimensionality

By Identity Provider

By Service Provider

By User

1h
1h
1h

1h
1h
1h

1h
1h
1h

1d
1d
1d