Graph-Based User-Entity Behavior Analytics for Enterprise Insider Threat Detection

CAMLIS 2023
Grant Gelven and Shannon Strum
About Me

Grant Gelven - Staff Data Scientist
I am a data scientist working with Stripe Security.
Quantitative Risk Assessments for Security Operations

Agenda

1. Motivation and Problem Statement
2. Quantifying Risk
3. Modeling User Behavior
4. Resource Impact Parameterization
5. Access Risk Score
Motivation & Problem Statement

Industry Trends
**Insider risk** is the potential for anyone with **authorized access** to harm an information system or enterprise through destruction, disclosure, modification of data, and/or denial of service.

This definition includes malicious and non-malicious (unintentional) attacks to assets, including people.
Insider Incidents in 2022

21+ Incidents per Organization
67% of organizations had more than 21 incidents in 2022. This has increase from 60% in 2020 and 53% in 2018.

$17.5 MM Annualized Cost
The average annualized cost of insider incidents was $17.5 MM in North America and $15.4 MM globally. Financial Services had the highest cost at $21.3 MM.

85 days Mean-Time-to-Resolve
The average number of days to contain and insider incident was 85 days. Only 12% of incidents were contained in <30 days. Those contained quickly reduced average cost to $11.2 MM.
Quantifying Risk

Insider Access Risk
How valuable or sensitive is a resource? This can be measured in any units, we choose to keep this unitless but know it is on an ordinal scale, so:

$$I(t) = \{ \text{High} \rightarrow 10, \quad \text{Medium} \rightarrow 2, \quad \text{Low} \rightarrow 1 \}$$

What is the likelihood that a user will interact, or not, with an asset on a given day? This term $L$ is dependent on the user and resource so:

$$L = L(s, t)$$

The most natural form for quantifying risk is to define a metric as the product of the likelihood of an adverse event occurring and the potential negative impact of said event. Given a user $s$ and resource $t$, we define:

$$r(s, t) = L(s, t) I(t)$$
Modeling User Behavior

Likelihood of Resource Use
Heterogeneous Audit Logs

```
{
    "edge_principal": "user1",
    "edge_action": "read",
    "edge_resource": "doc1",
    "edge_event_timestamp": 1483920000,
    "det_log_type": "gdrive",
    "atp_team": "team_a",
    "ata_permissions": null,
    "atr_is_confidential": true,
},
{
    "edge_principal": "userN",
    "edge_action": "ssh",
    "edge_resource": "VM1",
    "edge_event_timestamp": 1484950000,
    "det_log_type": "ssh",
    "atp_team": "team_b",
    "ata_permissions": "pk",
    "atr_is_confidential": null,
}
```

Aggregation

<table>
<thead>
<tr>
<th>principal</th>
<th>action</th>
<th>resource</th>
<th>count</th>
<th>day</th>
</tr>
</thead>
<tbody>
<tr>
<td>user1</td>
<td>read</td>
<td>doc1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>user2</td>
<td>write</td>
<td>app1</td>
<td>97</td>
<td>1</td>
</tr>
<tr>
<td>userN</td>
<td>ssh</td>
<td>VM1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
User-Resource Graph - Link Prediction Problem

Historical

User 1
User 2
User N
Doc 1
VM 1
App 1

Today

User 1
User 2
User N
Doc 1
VM 1
App 1
User-Resource Adjacency Matrix

<table>
<thead>
<tr>
<th>User</th>
<th>Doc 1</th>
<th>VM 1</th>
<th>App 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>4</td>
<td>25</td>
<td>259</td>
</tr>
<tr>
<td>User 2</td>
<td>5</td>
<td>0</td>
<td>97</td>
</tr>
<tr>
<td>User N</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>
We can factorize our historical graph, $X$, via Non-Negative Matrix Factorization (NMF) given it has no negative elements. With our choice of loss, this is equivalent to a Probabilistic Latent Semantic Indexing of $X$.

**Generalized KL-Divergence Loss (distance) where $Y = WH$**

$$d_{KL}(X, Y) = \sum_{i,j} (X_{ij} \log\left(\frac{X_{ij}}{Y_{ij}}\right) - X_{ij} + Y_{ij})$$
Link Prediction on Today’s Graph

New links in today’s graph can be predicted by:

\[ p(s, t) = \sum_z p(s|z) p(t|z) p(z) \]
Likelihood of Resource Use

Given a formula for predicting resource utilization at the individual user level, we can use this to compare actual user behavior to that which our model expects by aggregating over all actions taken in a day.

Likelihood of Anomalous User Behavior

\[
L(s) = 1 - Pr(\text{expected use} \mid s) = 1 - \sum_{t \in T} p(s, t)
\]
Resource Impact Parameterization

Impact Framework
## Impact Framework

<table>
<thead>
<tr>
<th>Ordinal Impact Value</th>
<th>Examples</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Critical</td>
<td>Non-Public Financial Information, Bank Accounts, Keys</td>
<td>Exfiltration, Money Movement</td>
</tr>
<tr>
<td>High</td>
<td>PII, Merchant Data, IP, Production Write Privs</td>
<td>Exfiltration, Sabotage, Espionage</td>
</tr>
<tr>
<td>Medium</td>
<td>Confidential Strategy Information</td>
<td>Exfiltration</td>
</tr>
<tr>
<td>Low</td>
<td>Operational Data</td>
<td>Exfiltration</td>
</tr>
<tr>
<td>Informational</td>
<td>Non-sensitive read-only</td>
<td>Login</td>
</tr>
</tbody>
</table>
Access Risk Score

Putting it all together
What is the Access Risk Score?

The Access Risk Score (ARS) is a quantitative measure of Stripe’s residual risk to an insider breach based on how internal users interact with our systems and data.

It is an index where higher values indicate that a user’s action may be anomalous.
ARS Calculation

Access Risk Score

Combining our estimates for likelihood and impact gives the final score. We default to daily aggregation here, but this is arbitrary. We can compute the score for any group of resources and/or users over any timeframe.

\[ R(s) = I_m L(s) \]

\[ R_{daily}(s) = k I_m [1 - \sum_{t \in T} p(s, t)], \]

where \( I_m = \max_{t \in T} I(t) \)

\( k \) is a normalization constant.
**ARS Statistics**

**Daily Score Distribution**

Standard ARS’ are Daily Aggregates

The score represents the average behavior of a user over the course of the day.

The vast majority of users (90%) act as we predict to within ~18%.
Quiz: Who is more predictable? A Data Scientist or an Account Executive?
Examining Job Families

Which people are considered “risky”?

The ARS is composable so we can aggregate along any dimension.

To the right, we can compare account executives vs. data scientists.

Data scientists have much more access and spend a lot of time performing discovery.
Security Operations
Monitoring, Detection, Response, and Guidance
Security Operations

Detection, Response, and Posture

Monitoring and Detection
Audit log collection, model scoring, alerting, and enrichment jobs run continuously. We monitor events on first and third-party systems, then alert on anomalies per the ARS and statistics if they occur.

Response and Investigations
Alerts may generate cases automatically which are then triaged by security operations teams. We may also use historical ARS data during incidents for post-hoc analysis.

Forward Guidance for Posture
Aggregate risk statistics can be used to illuminate gaps in security policies and/or high-risk system configurations. We can use our quantified risk metrics to prioritize work and resource allocation.
Security Operations - ARS Logical System Architecture
Outlier Detection

Standard ARS’ are Daily Aggregates

The score represents the average behavior of a user over the course of the day.

The vast majority of users (90%) act as we predict to within ~18%.
Outlier Detection

**Standard ARS’ are Daily Aggregates**

The score represents the average behavior of a user over the course of the day.

The vast majority of users (90%) act as we predict to within ~18%.

We can examine **extreme outliers** if policies warrant.
Event-Level Scoring

Aggregation

<table>
<thead>
<tr>
<th>principal</th>
<th>action</th>
<th>resource</th>
<th>count</th>
<th>day</th>
</tr>
</thead>
<tbody>
<tr>
<td>user1</td>
<td>read</td>
<td>doc1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>user2</td>
<td>write</td>
<td>app1</td>
<td>97</td>
<td>1</td>
</tr>
<tr>
<td>userN</td>
<td>ssh</td>
<td>VM1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
Investigation and Response

Event-Level Scoring

```
{
    "edge_principal": "user1",
    "edge_action": "read",
    "edge_resource": "doc1",
    "edge_event_timestamp": 1483920000,
    "det_log_type": "gdrive",
    "atp_team": "team_a",
    "ata_permissions": null,
    "atr_is_confidential": true,
},
{
    "edge_principal": "userN",
    "edge_action": "ssh",
    "edge_resource": "VM1",
    "edge_event_timestamp": 1484950000,
    "det_log_type": "ssh",
    "atp_team": "team_b",
    "ata_permissions": "pk",
    "atr_is_confidential": null,
}
```

### Aggregation

<table>
<thead>
<tr>
<th>principal</th>
<th>action</th>
<th>resource</th>
<th>count</th>
<th>day</th>
<th>Pr</th>
</tr>
</thead>
<tbody>
<tr>
<td>user1</td>
<td>read</td>
<td>doc1</td>
<td>4</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>user2</td>
<td>write</td>
<td>app1</td>
<td>97</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>userN</td>
<td>ssh</td>
<td>VM1</td>
<td>2</td>
<td>1</td>
<td>0.9</td>
</tr>
</tbody>
</table>
In a least privileged environment, we have a finite exposure of an incident occurring with probability $P_i$.

NMF provides a Chi-Squared Statistic that we can test, namely, $L(s)$.

Let’s test our expected behavior for a group with significance level $\alpha > p(s)$ (or $1 - \alpha < 1 - p(s)$).

We can choose our exposure budget to be anything, why not $P_i = \alpha$?

We’re done! Any $L(s) > P_i$ is an unnecessary risk for Stripe. This can be directly used to inform policy.
Thank you! Questions?
References