stripe

Quantitative Risk Assessments for Security Operations

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

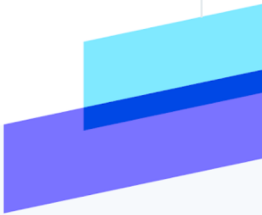
I am a data scientist working with Stripe Security.

Agenda

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

Motivation & Problem Statement

Industry Trends



stripe

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.

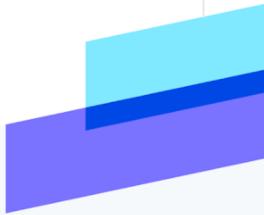
85 days

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



stripe

Quantifying Risk

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:

Likelihood of Access

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:

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

Resource Impact

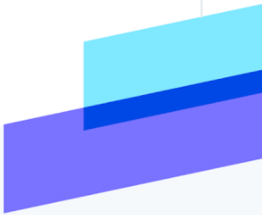
How valuable or sensitive is a resource? This can be measured in any units, we choose to keep this unitless but know it is on an ordinal scale, so:

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

$$r(s, t) = L(s, t)\, I(t)$$

Modeling User Behavior

Likelihood of Resource Use



stripe

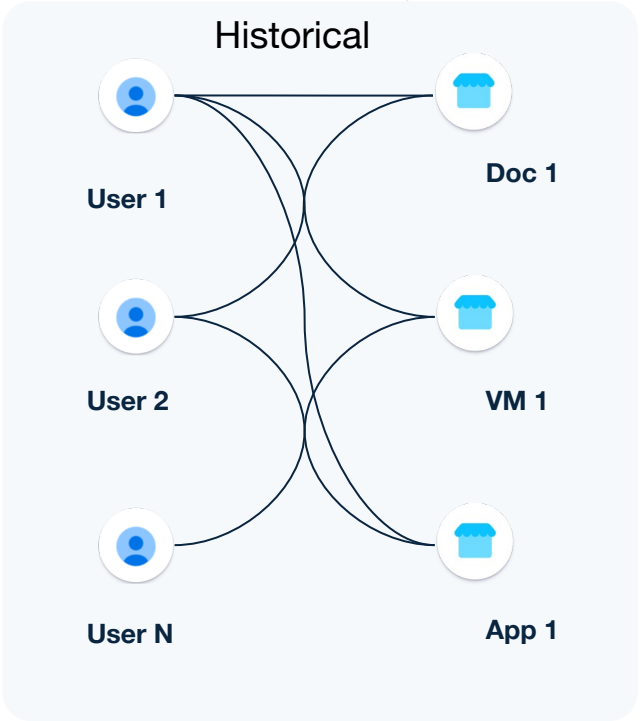
Heterogeneous Audit Logs

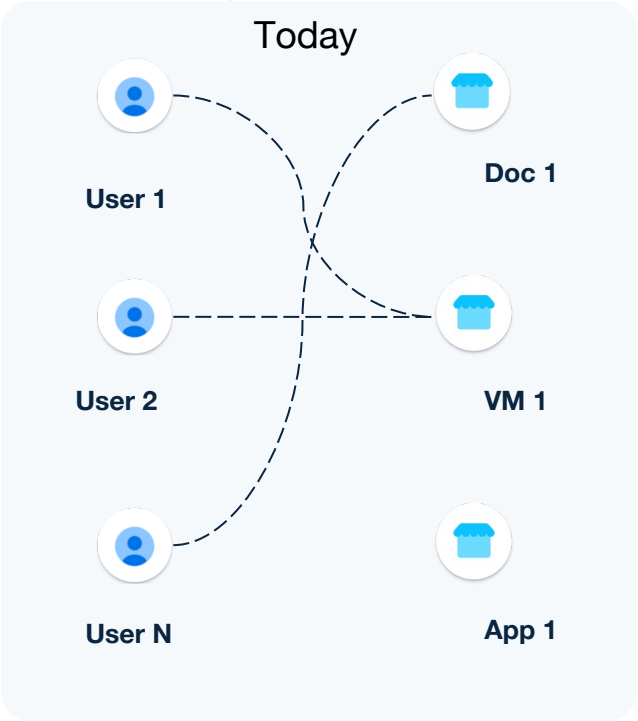
```
{
  "edge_principal" : "user1",
  "edge_action": "read",
  "edge_resource": “doc1”,
  "edge_event_timestamp": 1483920000,
  "det_log_type": “gdrive”,
  "atp_team": "team_a",
  "ata_permissions": null,
  "atr_is_confidential": true,
 },
{
  "edge_principal" : "userN",
  "edge_action": "ssh",
  "edge_resource":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

Aggregation

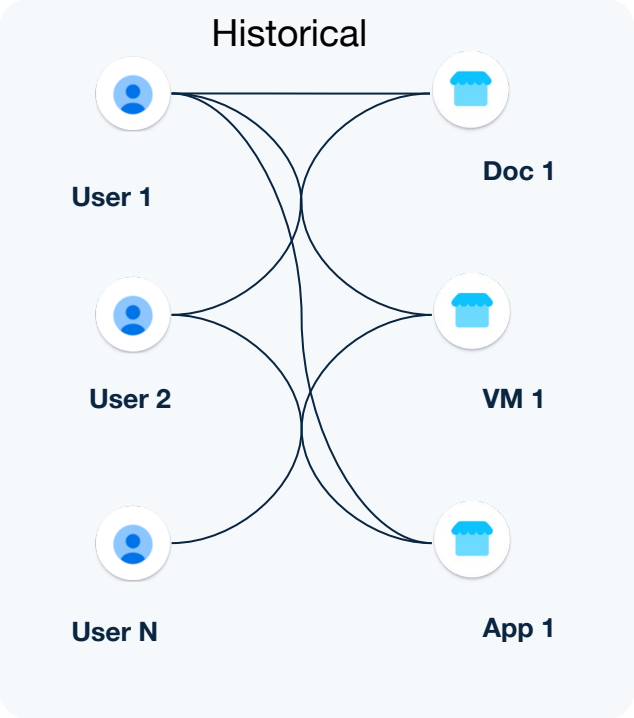
principal	action	resource	count	day
user1	read	doc1	4	1
user2	write	app1	97	1
userN	ssh	VM1	2	1

User-Resource Graph - Link Prediction Problem





User-Resource Adjacency Matrix



	Doc 1	VM 1	App 1
User 1	4	25	259
User 2	5	0	97
User N	0	2	0

Learning via Matrix Factorization

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**.

$$\mathbf{X} \cong \mathbf{WH}$$

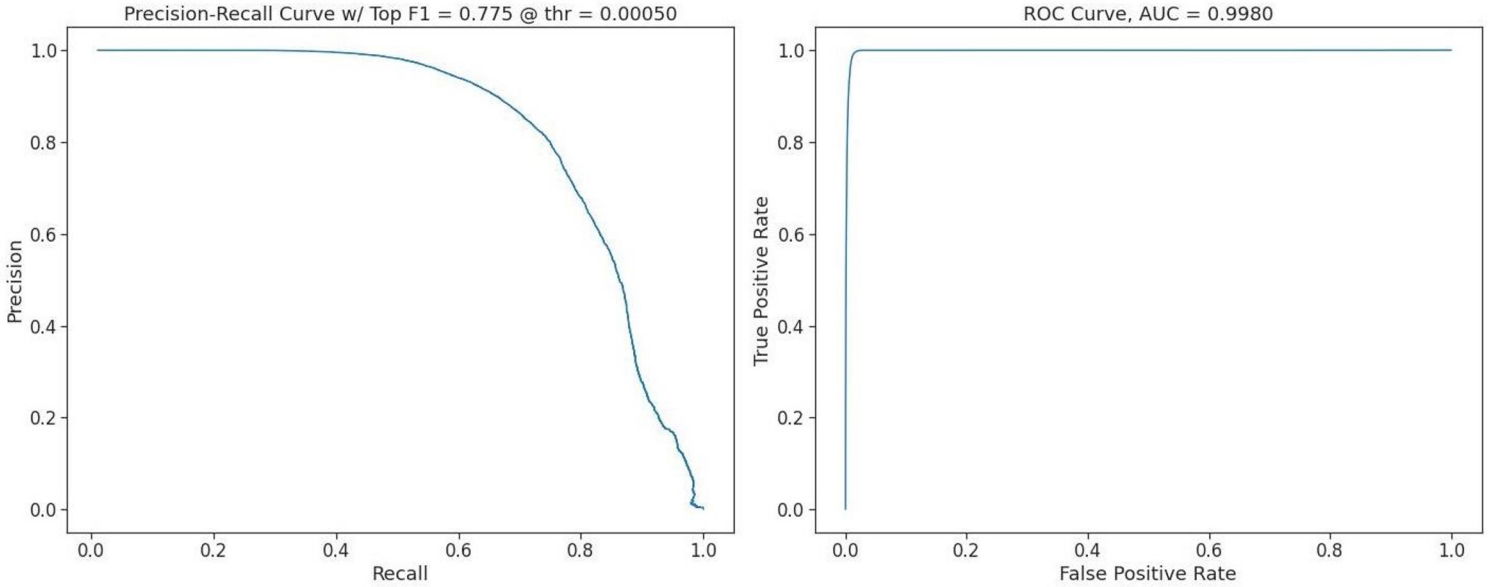
Generalized KL-Divergence Loss (distance) where Y = WH

$$d_{KL}(X, Y) = \sum_{i,j} (X_{ij} \log(\frac{X_{ij}}{Y_{ij}}) - X_{ij} + Y_{ij})$$

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.

$$Pr(s) = 1$$

Likelihood of Anomalous User Behavior

$$L(s) = 1 - Pr(expected\ use|\ s) = 1 - \sum_{t \in T} p(s,t)$$

Resource Impact Parameterization

Impact Framework

stripe

Impact Framework

	Ordinal Impact Value	Examples	Actions
Critical	100	Non-Public Financial Information, Bank Accounts, Keys	Exfiltration, Money Movement
High	10	PII, Merchant Data, IP, Production Write Privs	Exfiltration, Sabotage, Espionage
Medium	2	Confidential Strategy Information	Exfiltration
Low	1	Operational Data	Exfiltration
Informational	0	Non-sensitive read-only	Login

Access Risk Score

Putting it all together

stripe

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.

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.

$$R(s) = I_m L(s)$$

***k* is a normalization constant.**

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

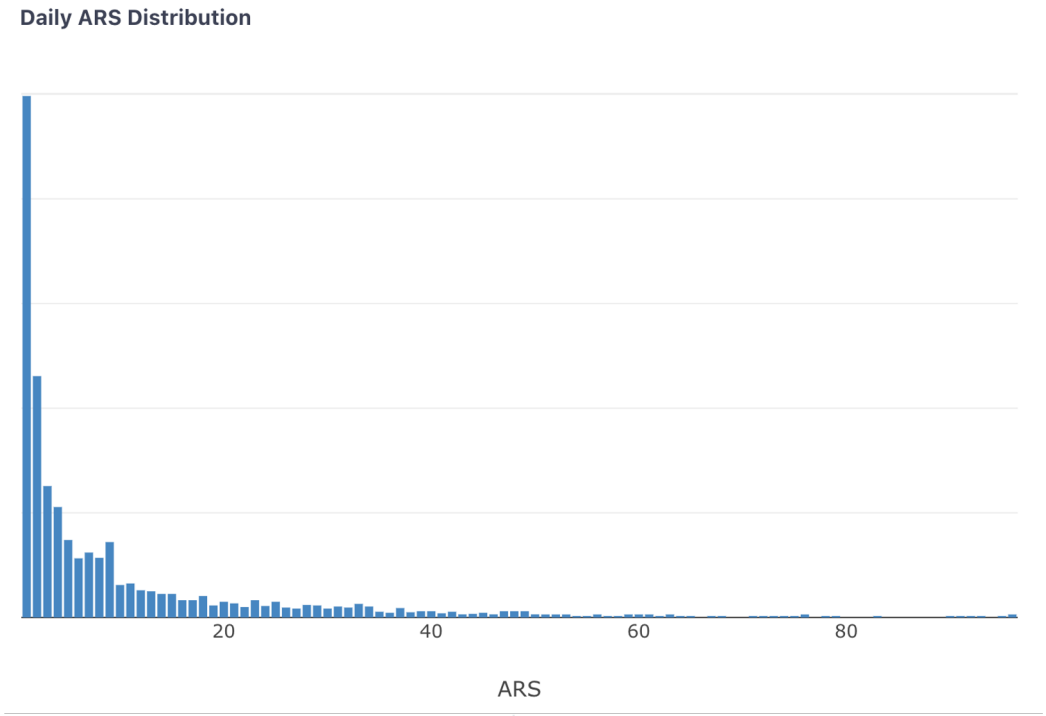
$$\text{where } I_m = max_{t \in T} I(t)$$

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?

Case Study

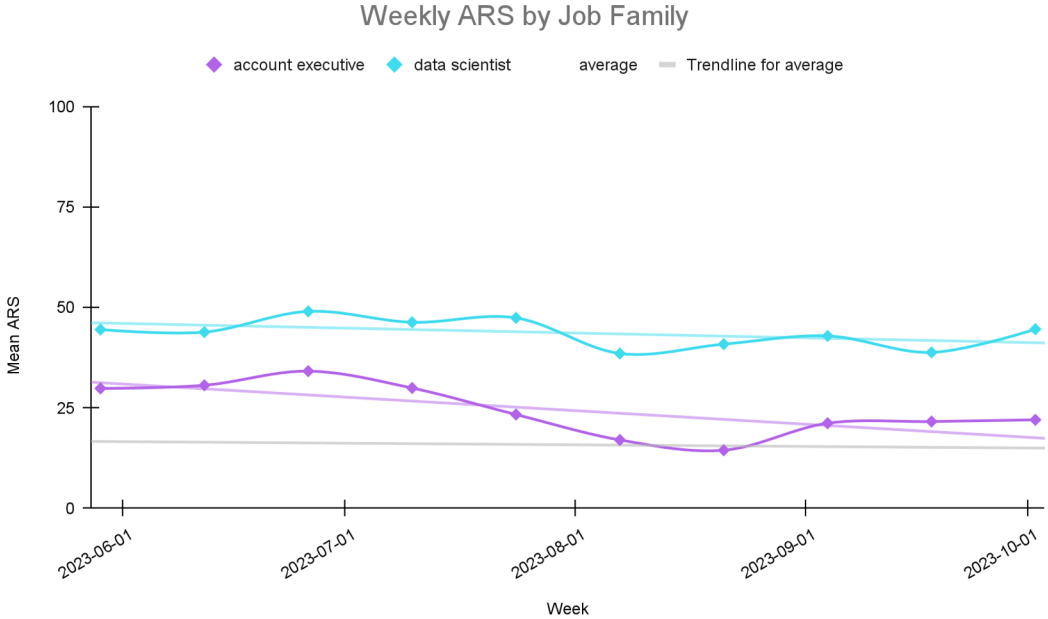
Examining Job Families

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

stripe

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.

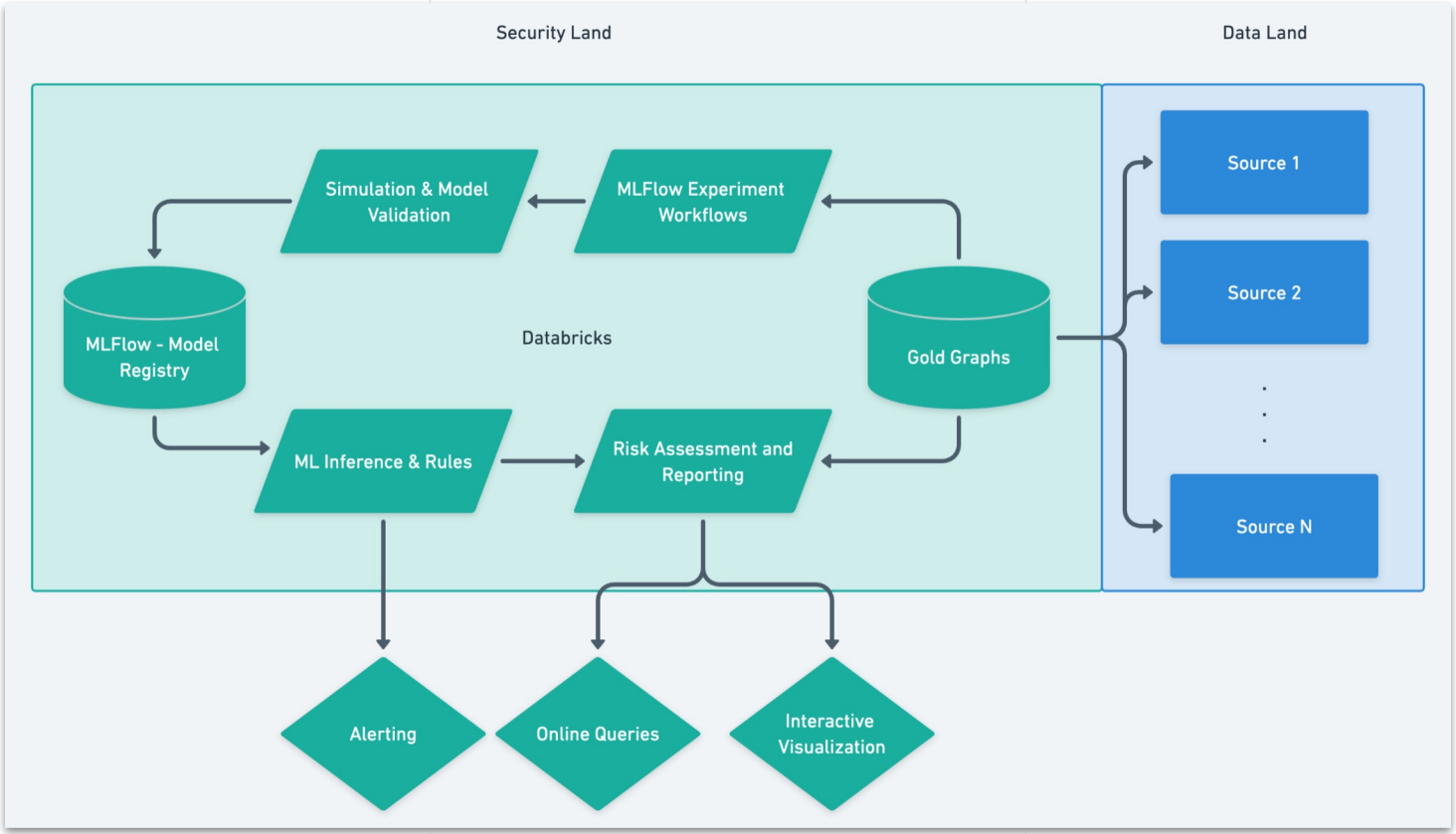
Response and Investigations

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

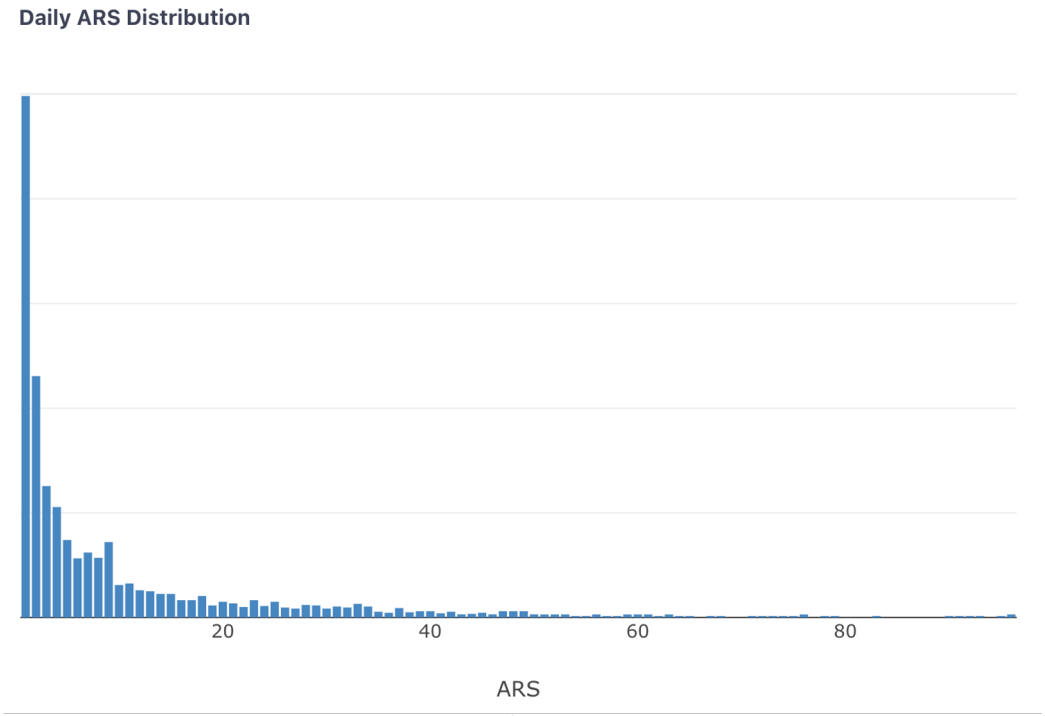


Outlier Detection

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%.



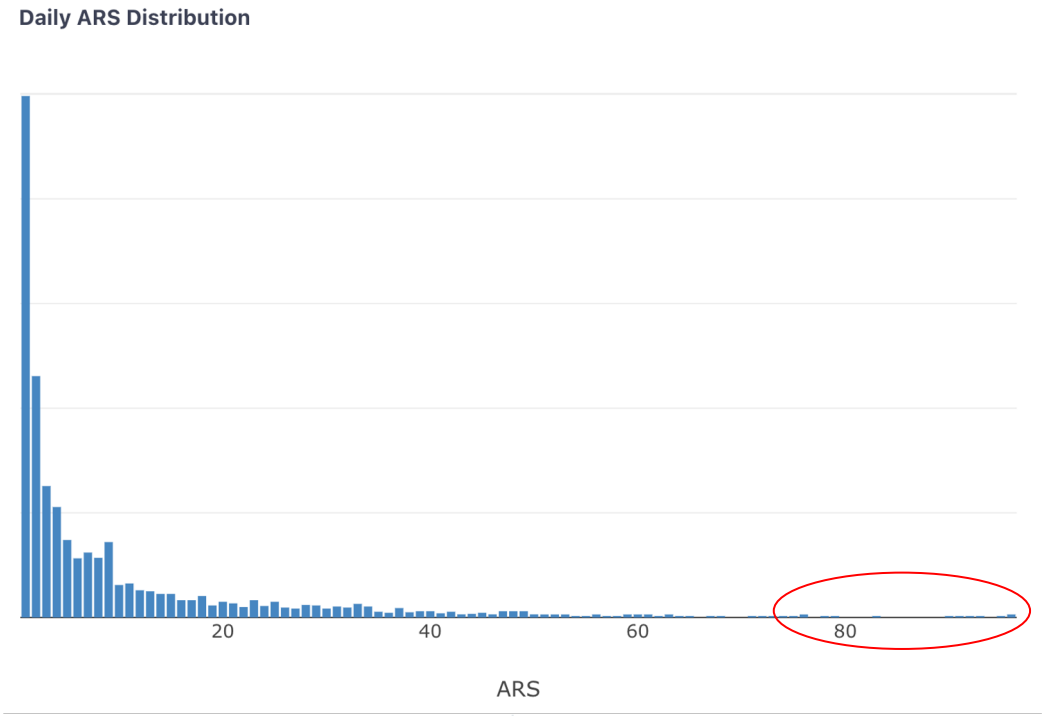
Outlier Detection

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

```
{
  "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

principal	action	resource	count	day
user1	read	doc1	4	1
user2	write	app1	97	1
userN	ssh	VM1	2	1

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

principal	action	resource	count	day	Pr
user1	read	doc1	4	1	0.1
user2	write	app1	97	1	0.5
userN	ssh	VM1	2	1	0.9

Informing Better Least Privilege

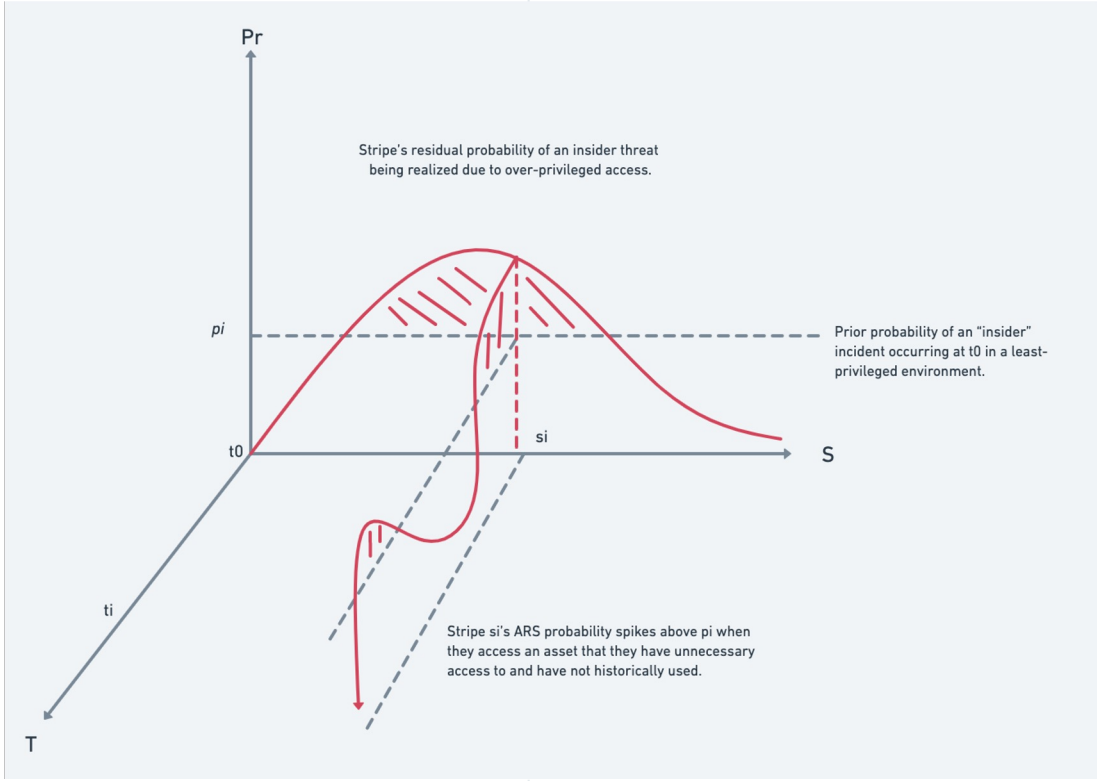
In a least privileged environment, we have a finite exposure of an incident occurring with probability Pi.

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

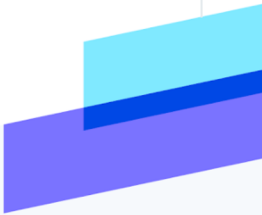
Let's test our expected behavior for a group with significance level $\alpha > p(s)$ (or $1- \alpha < 1 - p(s)$).

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

We're done! Any L(s) > Pi is an unnecessary risk for Stripe. This can be directly used to inform policy.



Thank you! Questions?



stripe

References

[1] 2022 Cost of Insider Threat Global Report. Ponemon Institute. 2022.

[2] Hwan Kim, Byung Suk Lee, Won-Yong Shin, and Sungsu Lim. Graph Anomaly Detection with Graph Neural Networks: Current Status and Challenges. In: *arXiv preprint arXiv:2209.14930v2 [cs.LG]* 4 Oct 2022

[3] Anagi Gamachchi, Li Sun and Serdar Boztas. A Graph Based Framework for Malicious Insider Threat Detection. In: *arXiv preprint arXiv:1809.00141v1 [cs.CR]* 1 Sep 2018

[4] Fucheng, L. et al. 2019. Log2vec: A Heterogeneous Graph Embedding Based Approach for Detecting Cyber Threats within Enterprise. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19). Association for Computing Machinery, New York, NY, USA, 1777–1794. https://doi.org/10.1145/3319535.3363224

[5] Pan, Y. et al. 2023. AttackMiner: A Graph Neural Network Based Approach for Attack Detection from Audit Logs. In Security and Privacy in Communication Networks. SecureComm 2022. https://doi.org/10.1007/978-3-031-25538-0_27

[6] Tadesse Zemichael and Rachel Allen. Heterogeneous Graph Embedding for Malicious Azure Sign-in Detection. In CAMLIS Abstracts 2022. https://www.camlis.org/tadesse-zemichael