Google | Privacy, Safety & Security

Detecting Threats at Hyperscale

Domagoj Klasic

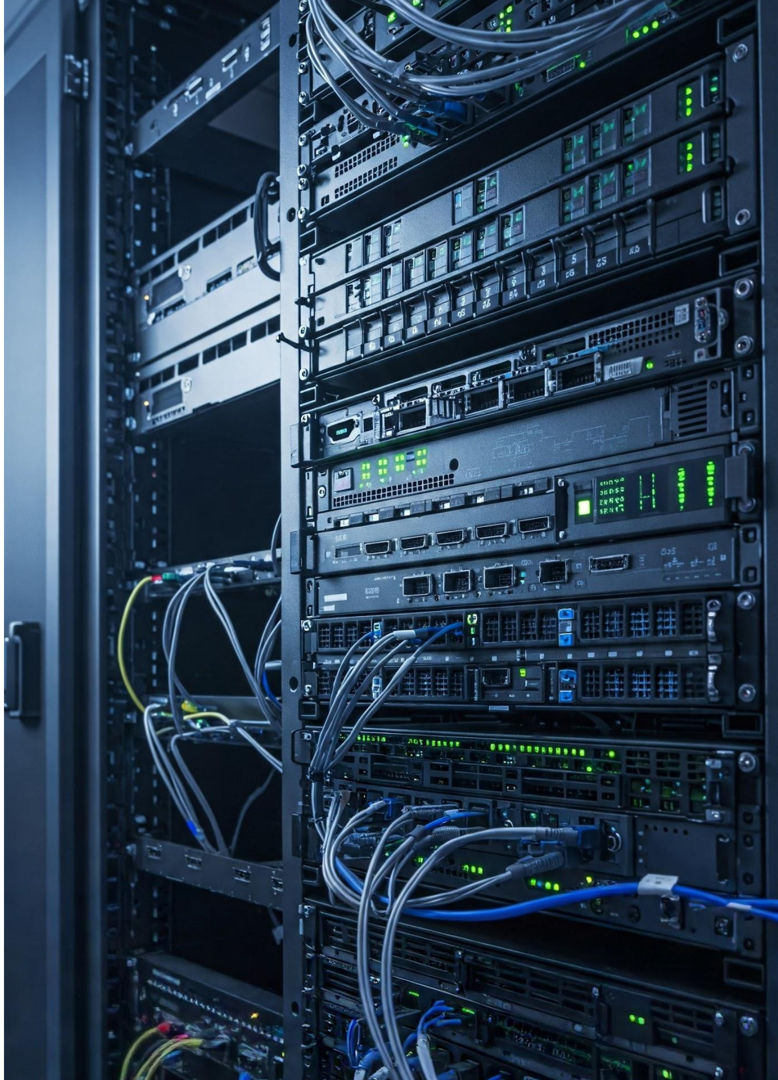
May 2025

About me !

Domagoj Klasic

Security Engineering Manager

- Lifelong interest in security
- Started in Google 12 years ago.
- Work in Detection & Response organization.
- Speciality: Intrusion detection and security operations
    - Newest skills: people management



Detection and Response in short

01

Mission

Our mission is to protect, respect and defend our users, googlers and the Internet. Many different teams contribute to the mission - intrusion detection teams, digital forensics, incident management etc.

02

Engineering and Operations

We combine security operational and engineering into one role. Excellence in both operations and engineering is crucial to success.

03

Strong partnerships

We're not alone. Strong partnership with other teams and organizations is also critical for success. Particularly important is partnership with our Software Engineering organization.

Detection at Scale

“Operating at Google scale is easy. Everything is uniform, predictable and most of the time - nothing new happens”

Nobody ever

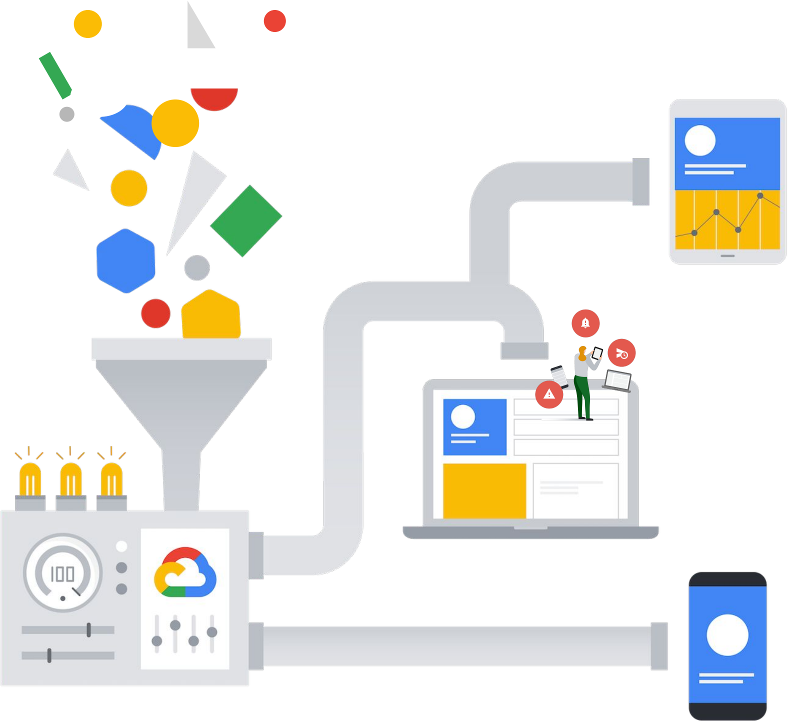
Quick Stats

Data Sources

450+ data sources
~~120 1.9 3~~ 7 trillion loglines / day
~~1 Tib/s network traffic / day~~ 40 TiB / second
1200 internal apps monitored
50+ Acquisitions

Data Processing

Pattern Matching (1200+)
Next-Gen Endpoint Agents
Machine Learning
Statistical Analysis
Manual Hunting
Automated Data Enrichment

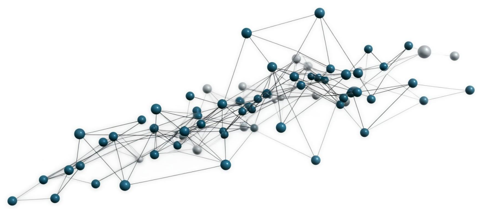


~~500,000~~
1+ Million Events per Year

99%+ are resolved using automation

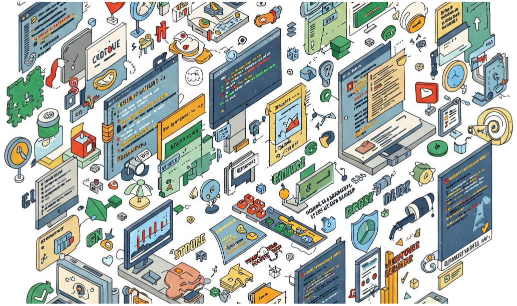
Pillars of detection at scale

Modeling



Simple log parsing and matching won’t do. We need to model the domain. Understand main entities, their relationships and model it in a strongly typed abstractions.

Intent based rules



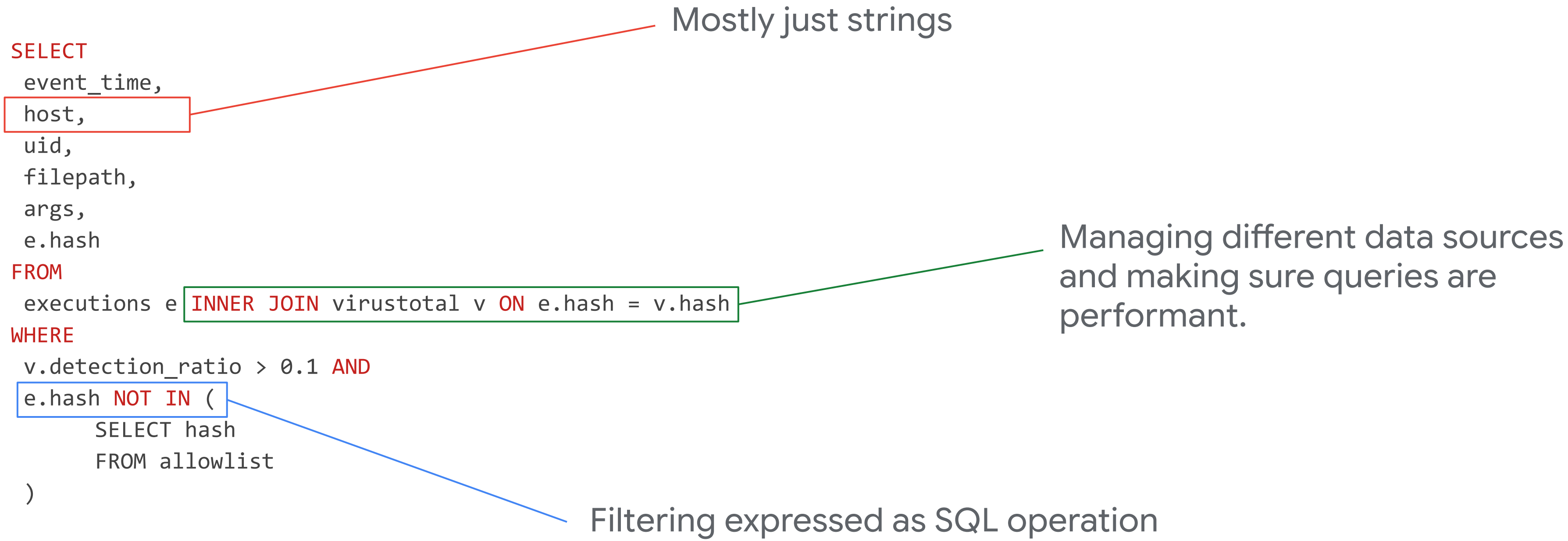
Wrangling data issues, query performance and picking the right data processing pipeline is best left to software systems. Security engineers should express rule logic and intent via unified API.

Context

Right context at **the right time** is crucial for confident analysis and detection. Integrate with various data sources to combat the noise.

Mock code example

Back in my day...



```
SELECT
 event_time,
 host,
 uid,
 filepath,
 args,
 e.hash
FROM
 executions e INNER JOIN virustotal v ON e.hash = v.hash
WHERE
 v.detection_ratio > 0.1 AND
 e.hash NOT IN (
      SELECT hash
      FROM allowlist
 )
```

Mostly just strings

Managing different data sources and making sure queries are performant.

Filtering expressed as SQL operation

Today...

Strongly typed, modeled domain.

```
message ExecutionEvent {
  google.protobuf.Timestamp timestamp = 1;

  HostIdentifier host

  Filepath file_path = 3;
  Filepath parent_file_path = 17;
  Filepath cwd = 4;
  FileHash file_hash = 5;
  Command command = 6;
  ProcessIdentifier process_id = 7;
  ProcessIdentifier parent_process_id = 8;

 (...)
}
```

```
func Rule() {
  filter := p.And(p.HasField(hashField),
p.Ne(p.Field(hashField), ""))

  logs.ExecutionEvent.Filter(filter).
  EnrichWith(enrichments.HashLookup(), p.Field(hashField)).
  FilterEnrichment(ioc.IsBadHash).
  OutputAsFact()
}
```

Bring in context when needed

No need to worry about logs storage

We’re hiring Security Engineers - Google Careers

Thank you