Taming Trillions of Events: Automated Security Telemetry Analysis at Scale

Daniel Tovarňák, Matúš Raček, Martin Gregorík, Martin Hamerník, Michal Čech
CSIRT-MU, Masaryk University, Brno, Czech Republic
{tovarnak,racek,gregorik,hamernik,cech}@ics.muni.cz

***Abstract*—The need of organizations for good network and security visibility remains a critical priority. However, due to the scale and complexity of modern networks, the amount of security telemetry data poses a significant challenge in terms of their ingestion, processing, storage, and analysis. After a decade of experience and several years of development, we will showcase the capabilities of an interoperable, production-proven data platform addressing these challenges. It is capable of sifting through trillions of stored security telemetry events at brilliant speeds, while supporting easy creation of automated analytical scenarios.**

***Index Terms*—Telemetry, Security, Automation**

I. Introduction

In the latest wave of the digital transformation era, the need for good network and security visibility remains a critical priority. However, due to the scale and complexity of modern networks, the amount of security telemetry data is steadily increasing, which presents significant challenges in terms of their ingestion, processing, storage, and analysis. In addition, raw telemetry data are often useless without a proper context, i.e. facts that are valid at the time of telemetry generation.

Moreover, as summarized in a recent survey [1], nearly two-thirds (65%) of cybersecurity professionals claim that their work is more difficult than it was two years ago. The commonly cited reasons are the increase of complexity, larger attack surface, increased workload, and skills shortage. Like many others, we argue that automation may help to at least alleviate the above-mentioned challenges. However, automated systems, much like humans, need to make decisions based on high-value and context-rich data.

II. Security Telemetry and Contextual Data

Security telemetry encompasses all (infinite) streams of data related to the operation of an IT infrastructure, which must be used by security operations in order to fulfill their goals and tasks, such as detection, investigation, or mitigation. It includes network flows, infrastructure logs, endpoint logs, various alerts, and more. We deem security telemetry to significantly overlap with network telemetry data [2].

When considering a sustained rate of 100,000 events per second (*eps*), after less than four months (116 days), the number of stored records reaches the trillion (1×10^{12}) mark. However, in practice, much longer retention periods are usually observed.

Contextual data, on the other hand, are not generated in such vast quantities, nor do they change very quickly in most cases. They include various allowlists and denylists, network topology, IP reputation, or known vulnerabilities. However, when they do change, it must be reflected immediately, since the semantics of the related telemetry may change with them.

The most common types of telemetry and context that we encounter in our environment are *IPFIX network flows* from multiple 100 GbE flow exporters deployed at the network perimeter and other internal exporters; *IPFIX NAT events* from organization-wide NAT boxes; *infrastructure logs* from network devices, network services, servers, and endpoints; *network scan results* from an orchestrated scanning infrastructure running periodic network-wide vulnerability scans; *vulnerability information* from publicly available databases; *IP address space information* from a centralized IPAM solution; and *DNS records* from the organization's DNS server. In terms of streaming data rates, the *IPFIX network flows* reach velocities of up to 60,000 eps, closely followed by the *infrastructure logs*, reaching up to 30,000 eps. The rest of the telemetry data pales in comparison.

III. Security Telemetry Data Platform

Almost a decade ago, our practical experience started to suggest that the traditional methods of processing and analyzing telemetry data were lacking. Since then, we have been steadily pursuing the goal of creating a unified data platform centered around fully structured data streams [3] and corresponding data tables. Thanks to the advent of new technologies, data formats, and the steady evolution of architectural patterns, all promising to store and query data at unprecedented scales, we have been able to create a basic prototype of such a platform, following the Data Lakehouse architecture [4].

We have continued in its development and confirmed that with suitable hardware, it is capable of sifting through trillions of stored events at brilliant speeds. As such, it has been further significantly enhanced to support easy creation of automated analytical scenarios and to meet production parameters.

The production instance of the data platform is currently deployed at 10 nodes in total, with all the nodes connected via two dedicated 100 GbE switches. The nodes are configured to form two independent Kubernetes clusters, each having a different purpose and HW setup. The *storage cluster* (4 nodes) consists of 96 phy. cores, ≈2 TiB RAM, and ≈560 TiB of NVMe flash storage. The *compute cluster* (6 nodes) consists of 232 phy. cores, ≈11 TiB RAM, and SSD system disks.

Figure 1 depicts the open-source technological stack of our solution, as well as important building blocks that needed to be developed, configured, and integrated to date to support many heterogeneous analytical scenarios of our security team.

978-3-903176-75-1 ©2025 IFIP

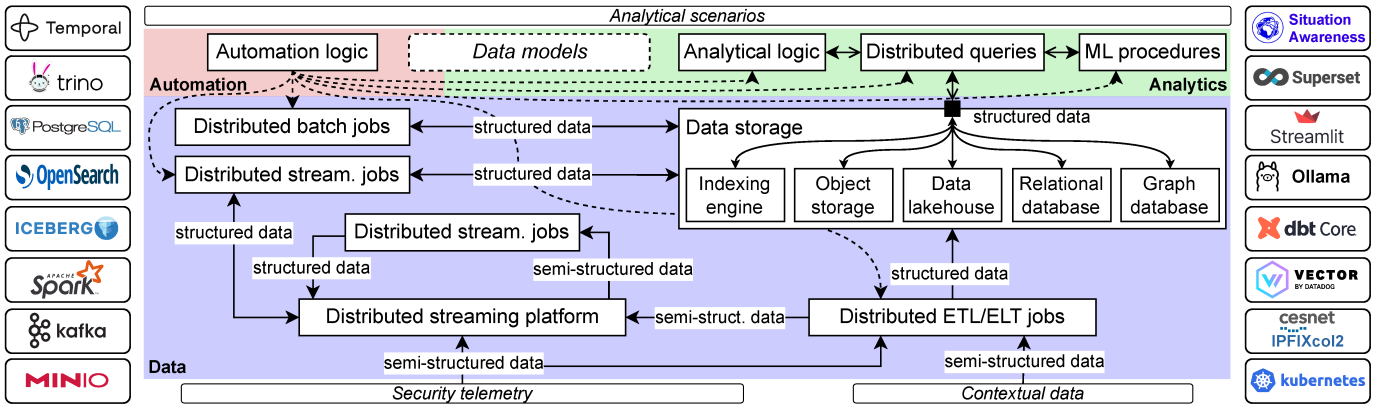

Figure 1: The technological stack of the data platform, its important building blocks, and integration logic

Kubernetes (K8s) is a distributed runtime and container orchestration system. Thanks to its holistic use in our environment, we are able to rapidly deploy new functionality and related software artifacts, as well as integrate new components within the technological stack. This helps us to make the process of developing new use cases and **analytical scenarios** quicker and more predictable. At the same time, it enables us to operate the platform with high-performance, high-availability, fault-tolerance, scalability, and observability in mind.

IPFIXcol2 is a high-throughput IPFIX flow data collector developed by the Czech NREN, CESNET. It is able to store or forward the ingested IPFIX flows using various data formats and protocols. In our case, it is used to convert IPFIX flows into semi-structured JSON objects, which are then sent into the *Apache Kafka* data streaming platform for further processing. We support both commercial and open-source flow exporters.

Vector is a versatile, yet lightweight, tool for ingesting, parsing, filtering, transforming, and routing of observability data. Thanks to its swiss-knife nature and very high performance, we use it to ingest and pre-process the rest of the semi-structured telemetry data, mostly in the form of log records. The primary data sink is again *Apache Kafka*.

Apache Kafka is a well-known **distributed platform for streaming data**. It is used for scalable, high-volume, low-latency data exchange as the security telemetry passes through its respective stages of processing and transformation. It also serves as an important point of data integration since it supports structured data streams with strict data contracts (schemas).

Apache Iceberg is an open format for big-data tables saved in a distributed storage. It allows for structured data to be organized into huge distrib. analytical tables that can be queried via SQL. It supports schema evolution, partitioning, and powerful data filtering to achieve high-performance, low-latency SQL queries over petabytes of data in **data lakehouses**.

Apache Spark is a distributed platform for building general-purpose **batch and streaming jobs** in a scalable manner. This is where the most complex data processing and transformation tasks are executed as massively distributed jobs. The ultimate goal of these (streaming) jobs is to transform the semi-structured telemetry data into fully structured records with respect to the desired **data models**. Such records can then be persisted as the distributed Iceberg tables. Additionally, periodic, distributed batch jobs are used to further enrich the persisted data tables, e.g. with IP geolocation. Last but not least, several batch and streaming job combinations are used to implement various **machine-learning procedures**.

MinIO S3 is a high-performance distributed **object storage** system with support for the S3 API. It serves as the central storage for all the persistent (in-situ) primary telemetry data within the platform. Most importantly, it holds all the distributed *Iceberg* tables and also the unprocessed raw data for backup and re-ingestion purposes.

OpenSearch is a distributed **indexing database and search engine**. It is best suited for medium-velocity semi-structured data with unstructured attributes, such as logs. We use it as a secondary storage with very short retention times, i.e. weeks, for full-text search over log data. It is helpful for initial exploratory analysis of semi-structured data in order to guide the process of their transformation into structured data models.

PostgreSQL is a well-known **relational database**, which can be used to store (comparatively) small amounts of data for direct, low-latency access via SQL queries. It is best suited for the storage of data models with contextual data and highly aggregated data models derived from other data sources, e.g. from the huge *Iceberg* tables.

Trino is a distrib. SQL query engine for scalable analysis of massive data sets. What *Apache Spark* represents for processing and storage of data, *Trino* represents for their search and analysis. Thanks to its compatibility with the *Iceberg* format, it is capable of executing powerful, **distributed SQL queries** over trillions of records persisted in the S3-backed tables. In addition to supporting materialized and traditional SQL views, *Trino* supports so-called cross-catalog queries and joins. As a result, it is possible to issue SQL queries over multiple other data sources, e.g. *PostgreSQL*, MariaDB, or *OpenSearch*. This allows for a high-complexity **analytical logic** when needed. Lastly, the corresponding Python client allows for excellent interoperability and integration capabilities, e.g. with *Temporal*.

dbt Core is a data manipulation platform for the orchestration of analytical SQL-based workflows. Thanks to the support of the *Trino* SQL engine, it serves as the main mechanism for an automated and scalable transformation of the primary data

models (*Iceberg* tables) into new derived data models, i.e. new tables. The **analytical logic** is typically defined as a collection of multi-step SQL queries, rendering the creation of the derived **data models** straightforward and possibly recursive.

Temporal is a distributed platform for durable execution of orchestration and automation workflows. It offers multiple SDKs for general-purpose programming languages (e.g. Java, Go, Python), which can then be used to define and implement the **automation logic**. The basic use-case in our context is the implementation of **distributed ETL/ELT jobs** for contextual data synchronization, e.g. for the IPAM data. It is also used for ETL/ELT jobs for telemetry data in the pull interaction model, e.g. for the network scanning results. Most importantly, it is used for explicit automation and orchestration of workflows that cannot be implicitly automated in other components.

Ollama is a platform for local execution of open-source large language models. When combined with appropriate libraries, e.g. LangChain, it enables developers to create fairly capable chatbots and agents. This is an experimental component used for research purposes in the area of LLM-based data analysis.

Apache Superset is an application for SQL-based exploratory data analysis and business intelligence. It is used as a day-to-day operations tool by security analysts and data engineers alike. It supports the creation of customizable dashboards, visualizations, and SQL query templates.

Streamlit is a framework for building interactive analytical applications in Python. It is best suited for the rapid development of new (automated) **analytical scenarios** with the need for a presentation layer. The combination of the *Trino* Python client and the support for Pandas library allows for the creation of powerful applications in a matter of hours.

CSIRT-MU SA is an in-house portal application for situation awareness. It combines all the data available in the platform into a single huge knowledge graph in order to provide a 360° view of both inventoried and non-inventoried network assets and their security posture.

IV. Demonstration

First, the security telemetry data models available as analytical tables will be introduced. The most important models and their semantics will be discussed in detail with slight emphasis on IPFIX flows. Contextual data models will be equally described, especially in terms of IPAM data. To put things in contrast, we will also sufficiently discuss the ingestion path of the raw data into the data platform, the building blocks interactions, and the processing steps that are needed to normalize the data and impose a strict structure on them. Also, interoperability, scalability, and performance characteristics of the solution will be presented.

We will then move on to typical analytical queries used by our security team, showcasing the data platform capabilities, performance, and ease of use. For example, the support for CIDR membership and the use of IP geolocation will be focused on, as they are indispensable in everyday use. Another example will include the search for a MAC address corresponding to a post-NAT IP address involved in an external security incident.

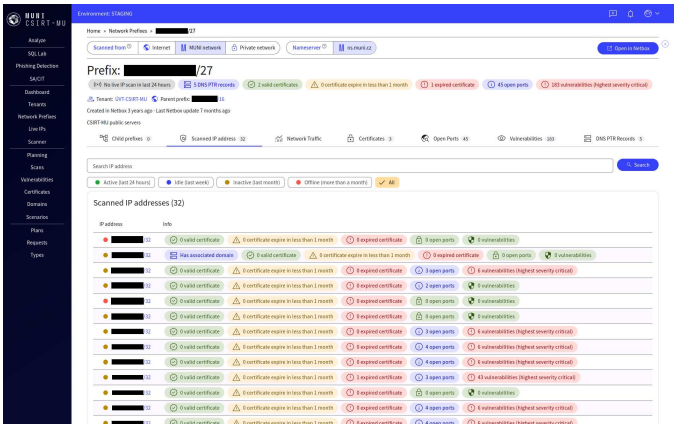

Figure 2: Situation awareness portal application by CSIRT-MU

Although perfectly achievable with months of historical DHCP and IPFIX data at the ready, it will also be shown that the data joining is not as trivial as it might seem.

To discuss the automation capabilities, we will go through an illustrative process of rapid prototyping of a new analytical scenario over DNS flows. It will be shown how straightforward it is to continuously build a passive DNS lookup table from primary DNS flows, thanks to the use of automation and orchestration of SQL code. The presentation layer, based on the Streamlit framework, will be included as well.

Next, the power of contextual data will be discussed in detail. It will be shown that by adding a simple context data model, new analytical scenarios quickly arise, for example, in the case of web server logs. In addition, the in-house CSIRT-MU situation awareness portal that combines all the available security telemetry and related context into a huge knowledge graph will be showcased in depth (see Figure 2).

Towards the end of the demonstration, we will show that thanks to the use of strict, well-described data schemas and the use of structured languages for querying, it is fairly easy to create LLM chatbots over the presented data models and the stored data, using only open-source tools and open models.

Last but not least, we will delve into the positive economic implications of an open-source security telemetry data platform, especially in the case of a fully open-source flow metering, export, and collection pipeline.

Acknowledgements

This research was supported by the Programme Open Calls for Security Research 2023-2029 (OPSEC) granted by the Ministry of the Interior of the Czech Republic under No. VK01030070 Automated Analysis of Security Telemetry.

References

[1] J. Oltsik and B. Lundell, *The Life and Times of Cybersecurity Professionals, VOLUME VII*. Enterprise Strategy Group by TechTarget, Sep. 2024.

[2] D. Zhou *et al.*, "A Survey on Network Data Collection," *Journal of Network and Computer Applications*, vol. 116, pp. 9–23, 2018.

[3] T. Jirsik *et al.*, "Toward Stream-Based IP Flow Analysis," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 70–76, 2017.

[4] D. Tovarňák, M. Raček, and P. Velan, "Cloud Native Data Platform for Network Telemetry and Analytics," in *2021 17th International Conference on Network and Service Management (CNSM)*, 2021.