SnT 2025 8 SEPTEMBER ONLINE DAY 9 TO 12 SEPTEMBER AT HOFBURG PALACE, VIENNA & ONLINE
CTBT: SCIENCE AND TECHNOLOGY CONFERENCE

T4.3-397

Implementing Observability for the Secure Web Portal

Mr PYNDA, Yaroslav[1]; Mr MACGREGOR, Robert[1]; Mr OLYVA, Serhiy[2]; Mr SUDAKOV, Alexander[1]

[1] - CTBTO Preparatory Commission; [2] - USoft HTI

CTBTO PREPARATORY COMMISSION | PUTTING AN END TO NUCLEAR EXPLOSIONS

INTRODUCTION AND MAIN RESULTS

In complex systems like the Secure Web Portal (SWP), understanding internal operations is essential for performance, reliability, and security. Our project focuses on improving observability using the Elastic Stack. By centralizing logs from all SWP components into Elasticsearch via Syslog, we built a robust and real-time monitoring solution that supports quick issue resolution and system insights.

We successfully integrated the ELK stack into SWP, enabling centralized log analysis and real-time insights into system health, data flows, and dependencies. This improved system resilience, captured key observability metrics, and laid the groundwork for future AI-driven monitoring.

SnT 2025 8 SEPTEMBER ONLINE DAY 9 TO 12 SEPTEMBER AT HOFBURG PALACE, VIENNA & ONLINE
CTBT: SCIENCE AND TECHNOLOGY CONFERENCE

Implementing Observability for the Secure Web Portal

Mr PYNDA, Yaroslav[1]; Mr MACGREGOR, Robert[1]; Mr OLYVA, Serhiy[2]; Mr SUDAKOV, Alexander[1]

T4.3-397

Building Infrastructure for Centralized Logging

To enhance observability in the Secure Web Portal (SWP), we built a logging infrastructure using the Elastic Stack see Fig. 1 (Elasticsearch, Logstash, Kibana) with Filebeat and Syslog. Filebeat collected logs from various sources and forwarded them to a central Syslog server. Logstash then parsed and enriched the data before sending it to Elasticsearch for indexing and analysis in Kibana.

One major task was harmonizing log formats, especially standardizing timestamps and field structures. This allowed consistent parsing and effective querying across all logs. We applied filters to clean and normalize log entries before ingestion.

Where necessary, we collaborated with developers to improve application - level logging. This included adding structured logs, key metadata, and capturing critical events to make logs more useful for monitoring and diagnostics.

The process required close cooperation between infrastructure and development teams, resulting in a scalable and reliable foundation for centralized observability in SWP.

Use of the Elastic Common Schema (ECS)

To standardize log data across multiple sources, we adopted the Elastic Common Schema (ECS). ECS provides a consistent structure and naming convention for fields in Elasticsearch, enabling seamless integration of diverse logs into a single, searchable format.

By aligning our logs with ECS, we significantly improved data consistency. Logs from different systems - whether application, infrastructure, or network - could be analyzed together without additional parsing or transformation. This simplified dashboard creation and enhanced the ability to correlate events across services.

ECS also enabled better filtering, aggregation, and visualization in Kibana. For example, common fields like *event.dataset*, *host.name*, or *log.level* allowed quick comparisons across components, supporting faster root cause analysis and more effective alerting.

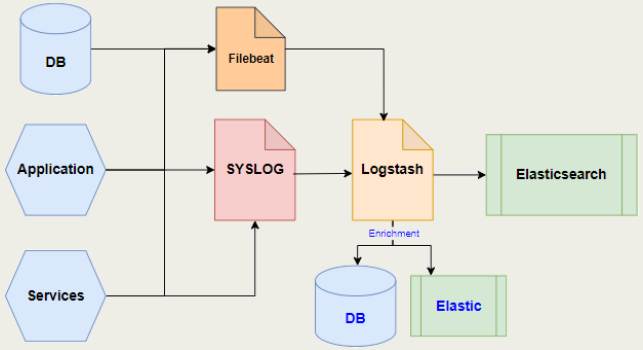

Fig 1. logging infrastructure using the Elastic Stack

Kibana Dashboarding and Alerting

Kibana was our main tool for visualizing and exploring log data see Fig. 2. We built dashboards to monitor system performance, data flows, and error trends in real time. These visualizations helped both developers and admins quickly spot and investigate issues.

Fig 2. Kibana dashboard to monitor SWP activities

Dashboards focused on key use cases like service uptime, log volumes, and error rates by host or application. With interactive filters and time-based graphs, users could easily drill down to analyze specific problems or events.

We also configured alerts in Kibana to detect anomalies such as error spikes or missing logs. Notifications were sent via email or messaging tools, allowing the team to respond quickly and maintain system reliability.