Research open source software for building SOC

Description: Provide the document regarding the design choice for open source SOC.. The document will include:

- Review of the open source software
- The system architecture for the proposed SOC

Introduction

Cyber attacks in manufacturing have proliferated across the globe, raising concerns about Thai manufacturing whether to move toward industry 4.0. Promotion of industry 4.0 transformation often recommends digital technologies such as cloud computing, data analytics, and Internet of Things. Cybersecurity, though an essential component, is not addressed as much in the context of industry 4.0. Strengthening cybersecurity in manufacturing means securing not only the IT (information technology) infrastructure, but also the OT (operation technology) components such as machines, controllers, and shop-floor facilities. To prevent and reduce impacts of cyber attacks in a factory setting, a complex real-time monitoring and evaluation of cyber risks is necessary. Because the industrial communication protocols consist of both proprietary and open standards, typical cybersecurity software tools, which are suitable for a normal IT/enterprise environment, may not be sufficient to monitor communication and data transfer among factory machines. While large manufacturers can invest in such complex security prevention services, either internal or outsourced services, small and medium manufacturers cannot afford to invest in neither internal nor outsourced cybersecurity services. Such a gap in cybersecurity affordability creates a kind of digital divide which is not healthy in today's global supply chains where large and small manufacturers must rely on one another.

An alternative solution to defense against both internal and external threats either come from enterprises or industrial factories is to develop and implement Security Operations Center (SOC) [1, 2, 3] for taking in charge of monitoring, analyzing, and mitigating the incoming threats. In general, SOC is the term used to describe activities that span (1) securing a portion of cyberspace, (2) monitoring and analyzing threats and incidents, (3) proactively protecting against the emerging threats, and (3) responsively managing and recovering from the incidents. In the core part of SOC, Security Information and Event Management System (SIEM) has been integrated as a centralized control point where all security notifications from various technologies including firewall, Intrusion Detection and Prevention systems (IDS/IPS), and anti-virus logs are collected, analyzed, and virtualized. In other words, it acts as a log management software specialized in security, which connects the endpoint security tools like firewalls and IDS/IPS, collects their log files, evaluate the security level of the organizational network, and provides alert information to the involved parties in case security threats have been detected. Thus, the effective operation of SOC in an organization is dependent on how well the SIEM can filter log events and generates actual alerts. Here lies the major problem faced by SOC analysts in detecting threats [4]. If proper alert correlation is not accomplished, analysts would have to deal with a large number of alert noise due to a high false positive detection rate. This would ultimately cause analysts to miss critical security incidents, thus resulting in a negative impact to the organization. On the one hand, the performance of a SIEM can be enhanced through adding external functional components such as threat hunting, threat intelligence, malware identification and prevention for reducing against false positive alarms. On the another hand, a machine learning technique can be applied to increase the accuracy and efficiency of the overall security operations process of the SOC system.

Laying down a foundation to develop and implement SOC, it can roughly classified into five components (as shown in Figure. 1).

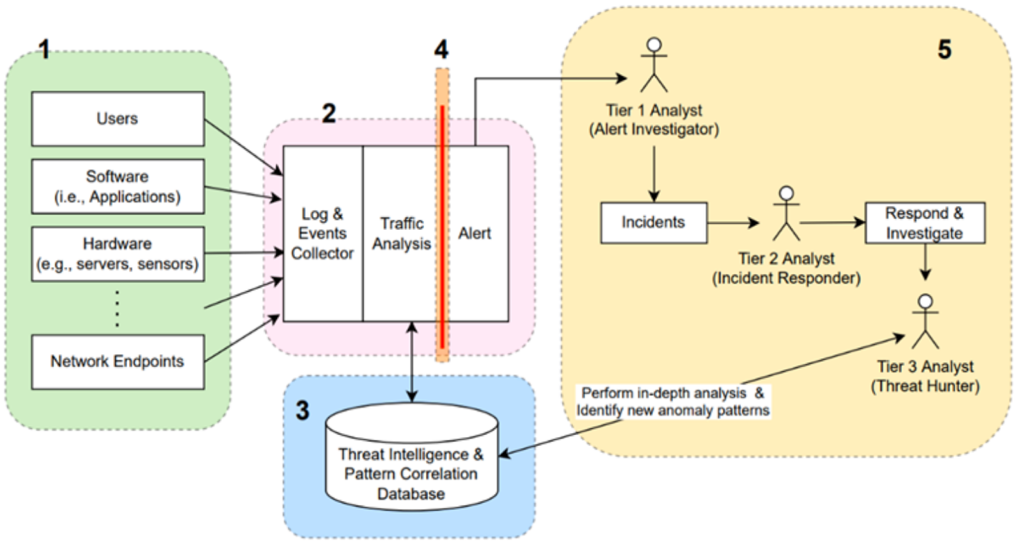

1
Users
Software (i.e., Applications)
Hardware (e.g., servers, sensors)
Network Endpoints
2
Log & Events Collector
Traffic Analysis
Alert
4
3
Threat Intelligence & Pattern Correlation Database
5
Tier 1 Analyst (Alert Investigator)
Incidents
Tier 2 Analyst (Incident Responder)
Respond & Investigate
Tier 3 Analyst (Threat Hunter)
Perform in-depth analysis & Identify new anomaly patterns

Figure 1: SOC components which comprises of the following components: (1) Assets identification and log forwarding, (2) Logs & event management, traffic analysis, and threat detection, (3) Threat intelligence and pattern correlation database, (4) Alert policy, and SOC workflow

1. **Assets identification and log forwarding:** It collects data assets from various security devices, monitors and analyzes these information. Then, it issues the alert results in case the potential attacks, information security incidents, or even compliance issues have been detected. Based on assets identified from the data collection process, data assets will be grouped into categories. The study will explore an appropriate integration approach to connect each type of data source to the centralized logs and management systems as well as present guidelines for managing the addition or removal of the assets from the SOC system.
2. **Logs & events management, traffic analysis, and threat detection:** These functionalities are typically performed by an automated Security Information and Event Management (SIEM) software. Comparative analysis will be conducted on open-source SIEM tools to evaluate capabilities and performance. Finetune adjustment will be performed to ensure the selected SIEM are interoperated with the factory environment and settings.
3. **Threat intelligence and pattern correlation database:** Potential threats will be identified to create initial data of threat intelligence and pattern correlation database. The design will also consider supporting connection with external threat intelligence feeds for catching up with the new type of cyber attacks and capability to plugin to support the addition of other SOC components or plugin modules to support more sophisticated functionalities, such as User and Entity Behavior Analytics (UEBA).
4. **Alert policy:** This can be used as a standard guideline in defining security policies and validating non-compliance of all networking resources deployed within an organization network.
5. **SOC Workflow:** SOC model will be customized to suit operation in factories by simplifying from the full-fledged SOC. Guidelines on the structure of the incident response team will be defined (e.g., roles, responsibilities, number of people required). An incident response plan will be developed as a minimum practice to ensure an adequate protection level is met.

In the following, we will discuss in more detail about the available open-source software examples that have been widely deployed to build a powerful SOC system with the purpose to collect, analyze, detect, and react against cyber-attacks.

1. Central Log Management and processing pipeline

It can be impossible for admins to individually check every logs for all networking devices deployed across their environment. Thus, Central Log Management (CLM) [5, 6] is built as a part of SOC system that can enable admins to easily monitor their networking environment by tracking key metrics and change activity. It is a type of logging solution system that consolidates all log data and forwards it to a central server. The main functionalities of CLM include (1) storing log data from multiple sources in a central location, (2) enforcing retention policies to gain a specific log data for a specific time period, (3) easily searching inside the logs for important information, (4) generate alerts based on the predefined security polices, and (5) setting up security alerts and granting login access to particular uses without granting server root access. Examples of open-source centralized logging solutions are discussed as follows.

1.1 Fluentd

Fluentd [7] is one of the most popular and widely deployed open-source tool that collects and forwards logs to different destinations. It is a cross-platform software project originally developed at Treasure Data. The program was designed to solve the challenge of big data log collection. It’s licensed under Apache License v2.0 and written in Ruby programming language. Fluentd has been designed as a unified logging layer that supports various log sources and formats (such as log files, syslog, network protocol, and TCP sockets), and then converts these logs into structured data before sending them to their intended destinations including databases, object storage, message queues, and other log receivers.

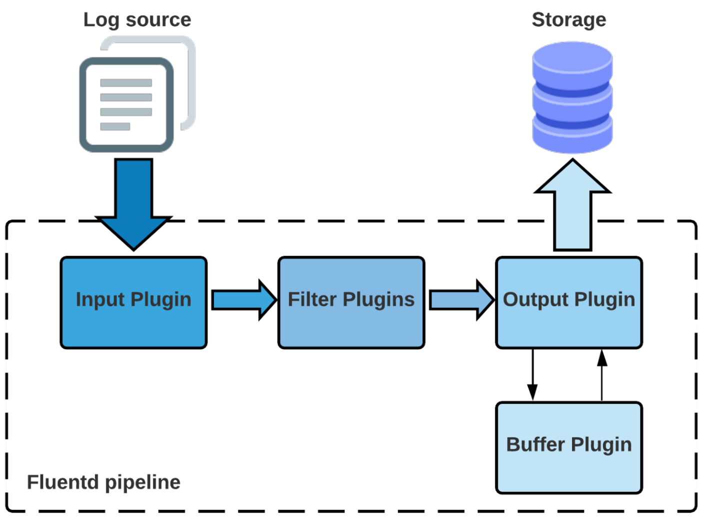


Figure 2: Fluentd architecture

Figure 2 shows the design architecture of Fluentd. It has a flexible plugin system that provides a wide variety of plugins to connect many different data sources and destinations. Typically, the plugins can be classified into six categories:

- Input: which is used to gather log data from multiple sources.
- Parser: allows users to parse source's custom data format which Fluentd can then understand.
- Filter: allows Fluentd to modify and adjust incoming logs before forwarding it to output plugins. It can extract or discard data based on various conditions, such as filtering logs that contain specific keywords/patterns.
- Output: uses to transmit log data to various systems (e.g., databases, search engines, cloud services, and log management tools).
- Buffer: used by output plugins to temporarily store the incoming log data before flushing them to the storage system. The size and duration of the buffer can be configured according to the specific use case.
- Formatter: used by output plugin to modify output formats according to user needs.

1.2 Fluent Bit

Fluent Bit [8] was developed in 2014 by the Fluentd team at Treasure Data to address the need for a lightweight log data processing solution in resource-constrained environments, such as Embedded Linux systems and gateways. It was designed as an integral part of the Fluentd Ecosystem and serves as an open-source program for efficient log data collection and processing in diverse environments that may have limitations, such as resource-constrained setups or complex cloud infrastructures. The core principle of Fluent Bit is to receive data from various sources, perform transformations and processing on the collected data, and then forward it to multiple destinations. It excels at achieving high performance while consuming minimal resources, making it well-suited for systems with resource limitations or small footprints. Moreover, it offers flexibility for customization and configuration to meet the specific requirements of users. It supports multiple configurations and data export formats, including sending data to multiple servers concurrently, writing data to files, or forwarding data to other systems like Apache Kafka, Elasticsearch, Amazon S3, Google Cloud Storage, etc.

In addition, Fluentd and Fluent Bit are two closely related projects. Both projects are licensed under the Apache License v2.0 and are hosted by the Cloud Native Computing Foundation (CNCF). They are vendor neutral and community-driven projects. Fluent Bit is designed and built on top of the architecture and general design of Fluentd, incorporating its best ideas. The choice between the two depends on the specific needs of the end-user. Table 1 provides a comparison of different aspects of the projects. Both Fluentd and Fluent Bit can work as Aggregators or Forwarders, they both can complement each other or use them as standalone solutions. In the recent years, Cloud Providers switched from Fluentd to Fluent Bit for performance and compatibility reasons. Fluent Bit is now considered the next generation solution.

Table 1: Comparison of Fluentd and Fluent Bit.

Aspect	Fluentd	Fluent Bit
Scope	Containers / Servers	Embedded Linux / Containers / Servers
Programming Language	C & Ruby	C
Memory	> 60MB	~1MB
Performance	Moderate	High
Dependencies	Relies on certain Gems for some features	Does not depend on external dependencies except for special plugins
Plugins	Over 1000 external plugins available	Over 100 internal plugins available
License	Apache License v2.0	Apache License v2.0
Resource Usage	Fluentd is larger and resource-intensive due to its extensive features and capabilities	Fluent Bit is smaller and uses fewer resources compared to Fluentd, suitable for resource-constrained systems
Customization	Fluentd offers a flexible plugin system to connect with various data sources and destinations	Fluent Bit also provides customization through plugins but with fewer plugin options compared to Fluentd
Recommended Use	Suitable for collecting and forwarding large-scale log data	Ideal for real-time data ingestion and system monitoring in resource-limited scenarios

1.3 Graylog

Graylog [9] is an open-source log management solution for capturing, managing, storing and analyzing logs. It can filter, sort, analyze, and display data in various formats including structured and unstructured data. It can collect and manage logs from various sources. Also, it has the ability to collect alerts and notifications, such as email, Slack, and PagerDuty notifications. In addition, Graylog provides REST APIs that allows developers to easily integrate it into their platforms and applications by sending requests and receiving responses via HTTP/HTTPS. According to the Graylog architecture (Figure 3), it has been developed based on a master-slave synchronization model comprising five main components.

- Collector: receives log data from various sources such as syslog, GELF (Graylog Extended Log Format), or Beats, and sends this data to the Graylog server.
- Graylog server: it is responsible for managing log and event data, including creating notifications and filtering out irrelevant information, managing users, roles, and permissions to control access to the system's logs and events, analyzing and visualizing log data stored in the system, and providing features such as searching log data based on conditions, filtering log data by event type, and displaying data through graphs and tables.
- Elasticsearch: it is a NoSQL database system used for storing and searching data especially for logs and events, and capable of storing a large amount of log data.
- MongoDB: it is a NoSQL database system used for storing configuration, metadata, and settings, including configuration, user, role, and permission data.
- Graylog web interface: the component used by users to manage log and event data in the system.

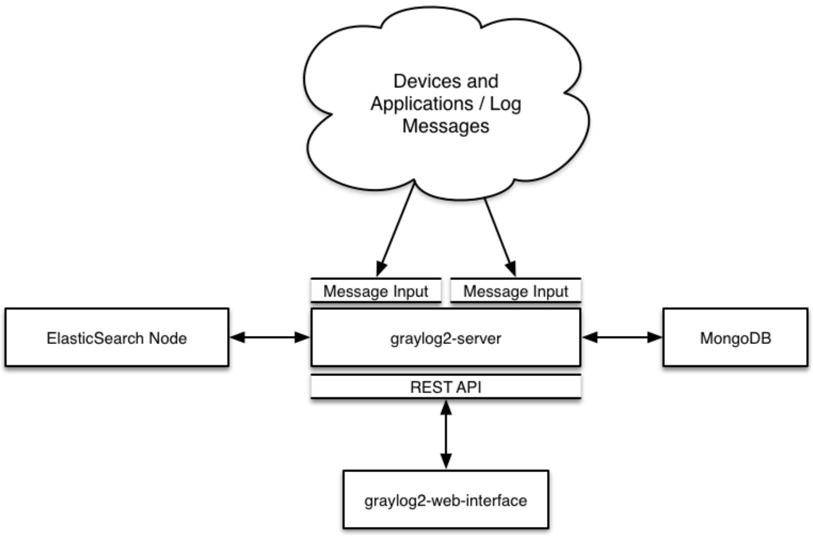


Figure 3: Graylog architecture

1.4 Apache Kafka

Apache Kafka [10-12] is an open-source event streaming platform which provides the ability to durably write and store streams of event and process them in real-time or retrospectively. It is a distributed messaging system of servers and clients that enables the processing of real-time streaming data. This data is continuously generated and transmitted, allowing for asynchronous communication between systems and eliminating the need for processes to wait for simultaneous data transfer. Apache Kafka can function as a message broker or message bus, facilitating direct communication between different applications without requiring an Application Programming Interface (API). It has been built based on three key capabilities: (1) enabling the developers to write and read streams of events including continuous import/export of their data from other systems, (2) storing streams of events durably and reliably for a certain time period as needed, and (3) streams of events can be processed in real-time or retrospectively. As shown in Figure 4, Apache Kafka has been designed based on a published-subscriber messaging model. It consists of servers and clients that facilitates communication between applications and services via a high performance TCP network protocol. Users write programs to send data (producers) and receive data (consumers), and data is stored in a topic format. Each topic is divided into partitions, with the number of partitions predetermined at the beginning and not recommended to be changed during operation. Apache Kafka comprises several components, including Producer, Consumer, Topic, Broker, and ZooKeeper.

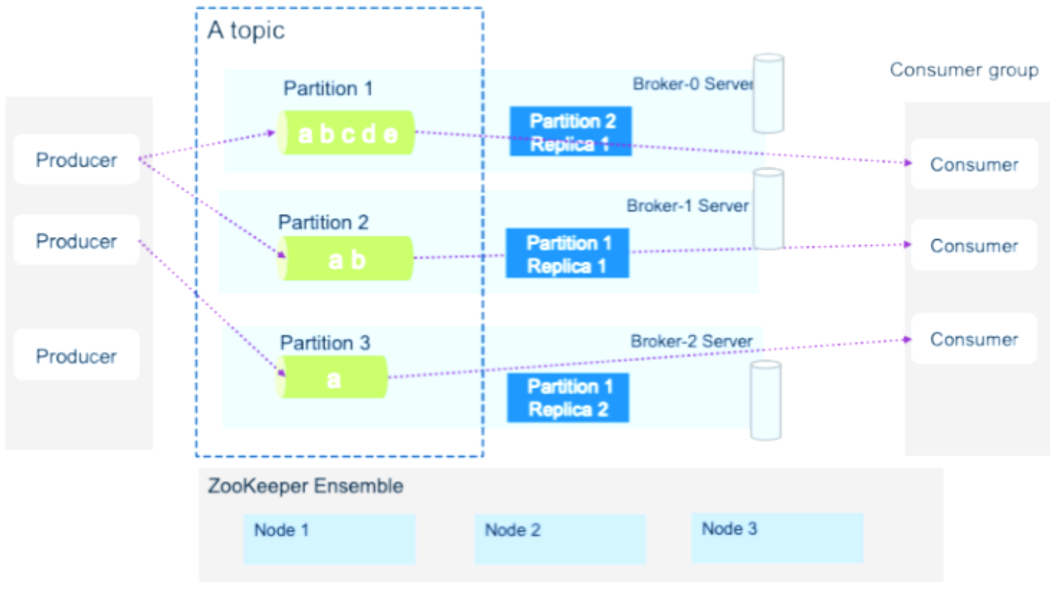


Figure 4: Kafka architecture

- Producer: It is responsible for creating and sending data to a Kafka cluster. It stores data and sends it to Kafka brokers. It has the ability to partition data into different partitions of a Topic based on specific criteria. This allows for better organization and more efficient distribution of data across the Kafka cluster.
- Broker: It is responsible for handling the transmission of data between Producers and Consumers, storing incoming data in Topics, and managing the storage of data within each Topic. It also controls access to data for various Consumers. In a Kafka cluster, there are multiple Brokers operating as a distributed system. Each Broker in a Kafka cluster is responsible for managing one or more partitions of a Topic. Brokers replicate data across the cluster to ensure fault tolerance and high availability. If a Broker fails, other Brokers can take over its responsibilities to ensure that data is still available for Consumers. In addition, data stored in each partition is ordered continuously, meaning that older data is located at the front and newer data is located at the back. Consumers read data from the front to the back (according to the offset value in each partition). It is important to note that there are various ways for consumers to read data. In each partition, data is always ordered, as shown in Figure 5.
- Topics: Topics in Kafka are similar to categories or channels in other messaging systems. Each Topic can have one or more partitions, and each partition can have multiple replicas for fault tolerance. Topics can be created, deleted, and configured using Kafka's command-line tools or a variety of Kafka client libraries.
- Consumer: It is responsible for reading and processing data from brokers. Consumers can select the desired topic to read data from, and Kafka brokers will send messages to consumers upon request. Consumers can track the status of data reading and configure the partition offset. In addition to reading data from a specific topic, consumers can also join a consumer group to distribute the load of processing data across multiple instances, which helps reduce the processing workload and increase the efficiency of using the Kafka cluster. Each partition in a topic can only be consumed by a single consumer group at a time, but multiple consumer groups can consume data from the same topic simultaneously.

- ZooKeeper: It is a distributed coordination service that helps manage and monitor Kafka brokers and Consumers. It maintains information about the status and location of each Broker, as well as metadata about Topics and partitions. Kafka uses ZooKeeper to elect a leader for each partition and to coordinate Consumer Groups.

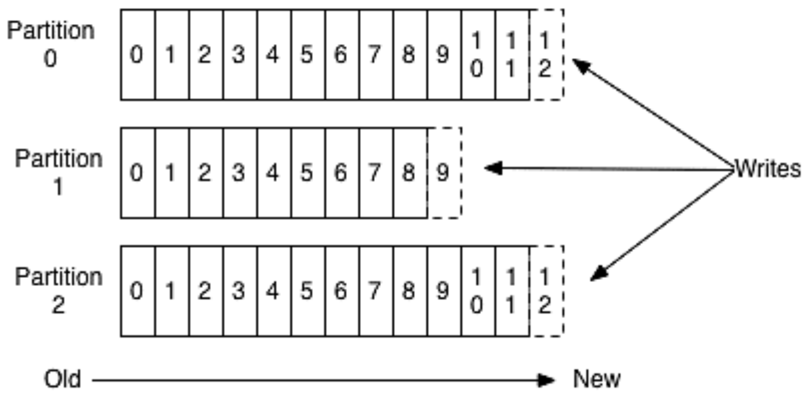


Figure 5: Kafka partitions

1.5 Rsyslog

Rsyslog [13-15] is Rsyslog is an open-source software utility used on Linux and UNIX-based operating systems for forwarding log messages and event data in an IP network. It has the ability to collect and send events to multiple destinations including various files and database systems such as MySQL, PostgreSQL, and MongoDB. It has been implemented using the basic syslog protocol, while extending the capabilities with flexible configuration options, rich filtering capabilities, content-based filtering, queued operations to handle offline outputs,

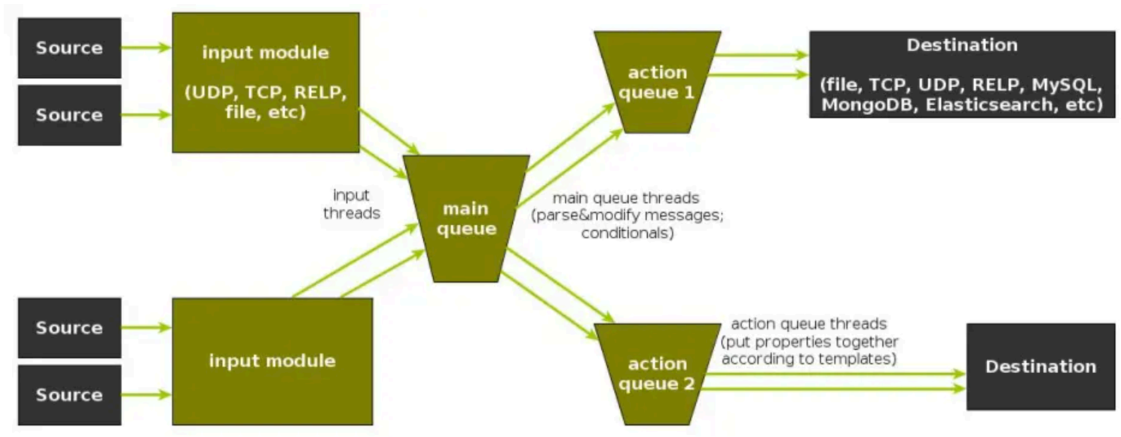

Source
Source
input module
(UDP, TCP, RELP, file, etc)
input threads
main queue
action queue 1
Destination
(file, TCP, UDP, RELP, MySQL, MongoDB, Elasticsearch, etc)
main queue threads (parse&modify messages; conditionals)
Source
Source
input module
action queue 2
action queue threads (put properties together according to templates)
Destination

Figure 6: Rsyslog architecture

and supporting for different module outputs. Figure 6 shows the architectural framework of Rsyslog which consists of the following modules.

- Input modules: There are used for receiving data from various sources such as syslog, TCP, UDP, file, MySQL, etc.
- Message parser: It is responsible for converting data into a format that Rsyslog can understand. In particular, Rsyslog can handle only data in plain text format. If the data has other format files (e.g., JSON, XML, and binary format), a message parser is required to convert this data into a format that can be understood and processed by the Rsyslog.
- Ruleset engine: It is the component that processes data and perform actions on that data according to the predefined rules such as filtering, outputting to files, or sending to other sources.
- Output module: It is the component that sends data to various destinations (e.g., syslog, TCP, UDP, file, MySQL, Elasticsearch). Once the data is processed by the Ruleset engine, it is passed to the output module for forwarding the data to the specified destination.
- Action queue: It is used to store data before sending out to the output destinations. In other words, there can be multiple Action queues operating simultaneously.
- RainerScript: It is a scripting language specially designed for processing network events and configuring event processors. It is the prime configuration language used to define data processing and rules for Rsyslog. It allows users to specify various actions to be taken on log messages, such as filtering, transforming, and forwarding.

In addition, Rsyslog has plugins as components that help to manage log data. Plugins are additional modules called by Rsyslog to detect and manage log data based on the nature of each plugin's work. For example, the *imfile* plugin connects to real-time log files, the *imtcp* plugin receives log data via the TCP protocol, and the *omelasticsearch* plugin sends log data to Elasticsearch, as shown in Figure 7. The plugins shown on the left and right hand sides of the Figure are the input and output plugins, respectively. The input plugin is responsible for receiving log data from various sources such as files, Syslog, UDP/TCP, or other collections. Meanwhile, the output plugin is responsible for sending log data to various destinations such as files, Syslog, MySQL, Elasticsearch, AWS CloudWatch, or other collections. Each plugin can specify the format and storage of log data, and the selection of each plugin depends on the usage and requirements. Also, Users can develop their own plugins according to their needs.

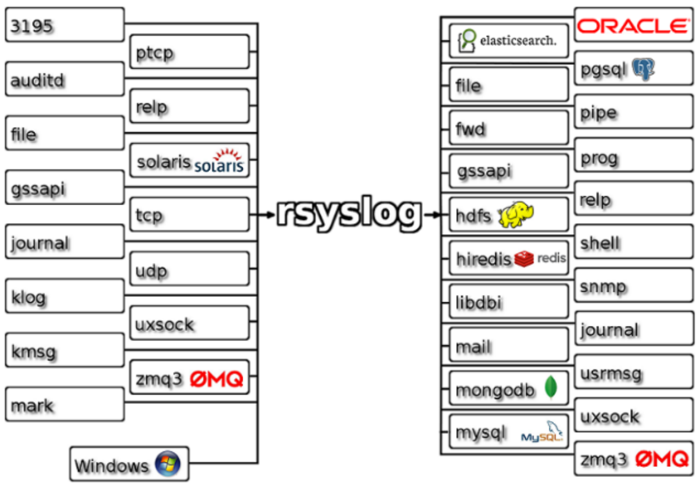


Figure 7: The different types of Rsyslog plugins

1.6 Elastic Stack (ELK Stack)

Elastic Stack [16-19] is a group of open-source tools from Elastic designed to helps users for managing and analyzing data in various formats. It can handle both structured and unstructured data and is designed to support data management and analysis in different situations such as security monitoring, web analytics, and IoT data management and analysis. It can be deployed on premises or made available as Software as a Service (SaaS). In general, Elastic Stack was formerly known as the ELK Stack (Elasticsearch, Logstash, and Kibana), but

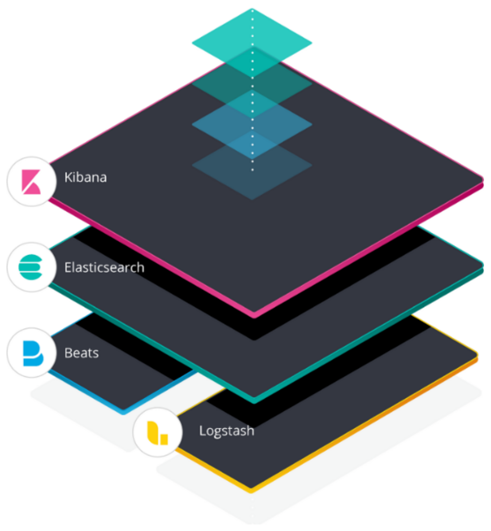


Figure 8: The overview architecture of Elastic Stack

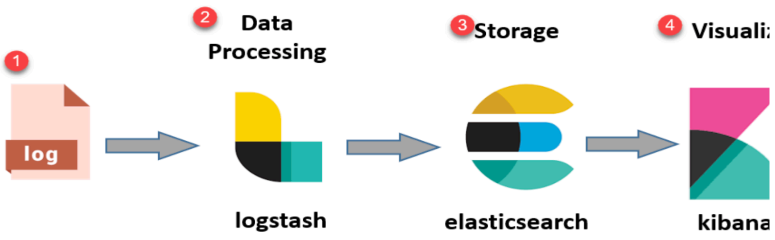


Figure 9: Elastic Stack flow diagram

has been rebranded as the Elastic Stack by subsequently added Beats to the stack. Figure 8 shows the core products of Elastic Stack along with their functionalities.

- Beats: There are data shippers that are installed on server as agents who used to send different types of operational data to Elasticsearch either directly or through Logstash, where the data might be enhanced or archived.
- Logstash: It is a data collection engine that unifies data from multiple sources, offering database normalization and distribution of data.
- Elasticsearch: It is a RESTful distributed analytics and search engine for all types of data including textual, numerical, geospatial, structured and unstructured.
- Kibana: It is an open-source fronted application that sits on top of the Elastic Stack. It provides search and performs data visualization for data indexed in Elasticsearch. The software makes complex data streams more easily and quickly understandable through graphic representation. Also, it acts as the user interface for monitoring, managing, and securing an Elastic Stack cluster, as well as providing a centralized hub for built-in solution developed on the Elastic Stack.

For concreteness, Figure 9 illustrates the connection between Beats, which are agent-based Log collectors installed directly on servers. It communicates with Logstash using the HTTP protocol in order to send Log data to the Logstash server. In other words, Logstash acts as a middleware for data communication between Elasticsearch and Kibana, allowing data accessed via Elasticsearch to be easily displayed in various formats through Kibana. Meanwhile, Kibana is capable of creating dashboards, visualizations, and alerts.

1.6.1 Beats

The Elastic Stack expands the capabilities of Elasticsearch by adding extremely useful tool to work alongside Logstash or Elasticsearch. One of the most useful tools is the Beats ecosystem [20-22]. Specifically, it is a collection of lightweight, single-purpose data shipping agents used to send data from hundreds or thousands of machines and systems to Logstash or Elasticsearch. Beats are great for gathering data as they can sit on any servers/containers, or deploy as functions then centralize data in Elasticsearch. Nowadays, there are several types of Beats available as shown in Figure 10. For example, Filebeat, Metricbeat, Packetbeat, Winlogbeat, and Auditbeat. Each type of Beat works in a similar way. That says an agent is installed and used on each machine or device to monitor and collect data from various sources such as server logs or IoT devices, and then parses the data to Elasticsearch or Logstash in real-time. The different types of Beats can be explained as follows:

- Filebeat: It is used to collect and harvest logs and data from various sources such as log files, system logs, and audit logs.
- Metricbeat: It is used to collect metrics such as CPU usage, memory usage, and network traffic.
- Packetbeat: This beat used to analyze and collect data from network traffic such as HTTP, DNS, MySQL, and other protocols.
- Heartbeat: It is used to check the status and health of various systems such as web servers, database servers, etc., by sending HTTP, TCP, ICMP, or UDP requests to servers and devices and verifying their response. Heartbeat can be used to determine if a network or application is functioning properly and can notify users in case of any issues.
- Winlogbeat: This beat used to collect Windows event logs.
- Auditbeat: It is used to collect audit logs from various sources such as the Linux audit framework, Kubernetes audit log, etc.

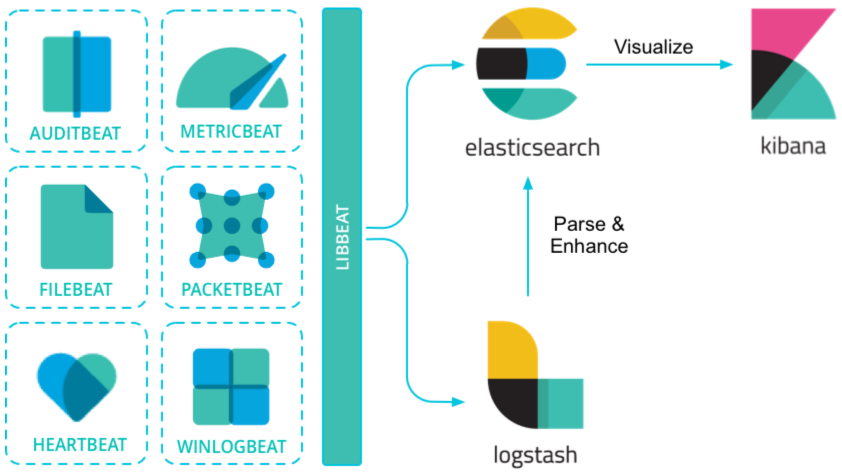


Figure 10: Types of Beats

1.6.2 Logstash

Logstash [21,23] is an open-source server-side data processing pipeline which has the capability of collecting data from a variety of sources, parsing, filtering, and transforming the data before storing it in a database system, Syslog server, or other systems in various formats. It takes care of data collection and processing. That says, it is used to collect, parse, filter, and transform the system log data. It works as a pipeline where the incoming log data from servers is being centrally taken and processed by Logstash as input, and ships the executed data to various supported destinations such as Elasticsearch as output. With pre-built filters and support for over 200 plugins, Logstash supports a variety of inputs by allowing users to easily ingest unstructured data regardless of the data source or format, while converging its into a common format and a continuous streaming fashion for further enhancing powerful analysis and business value. For example, system logs, website logs, and application server logs. In particular, the design principle of Logstash involves three stages (as shown in Figure 11).

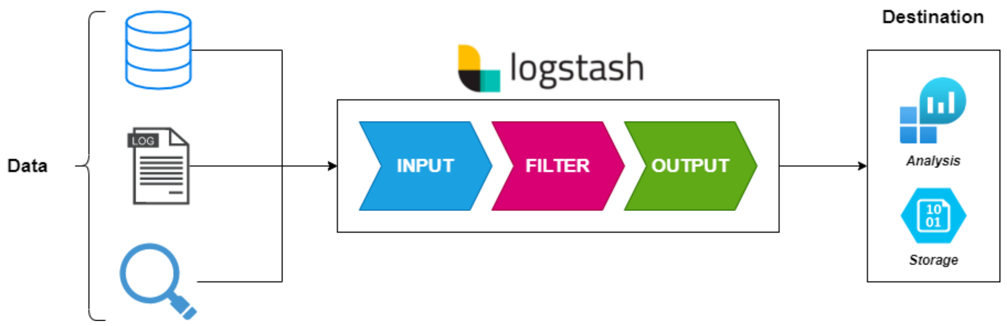


Figure 11: Logstash architecture

- Inputs: This is the first stage in the Logstash pipeline in which data is ingested into Logstash from a source. It can manage various inputs such as plain file, apache logs, events log, system log, nginx log, etc. To achieve the goal, Logstash uses input plugins to receive, collect, and place the data on an internal queue.
- Filters: This is the intermediate stage in the pipeline of Logstash which uses filter plugins to parse and manipulate the input events with regards to the required criteria. Logstash allows for combining filters with certain conditions to perform an action on the input event in order to meet a particular criteria. Some useful filters include '*age*' which is used to calculate the age of an event, '*clone*' for duplicating events, '*drop*' for dropping all events.
- Outputs: This is the final stage of the Logstash pipeline. Once the data has been ingested, the processing threads send the data to appropriate output plugins, which

are responsible for formatting and sending data onwards. Some commonly used outputs include Elasticsearch, HTTP, email, S3 file, Syslog, PagerDuty alert, and so on.

1.6.3 Elasticsearch

Elasticsearch [16-19,22] is a distributed, open-source search, and analytics engine built on Apache Lucene and developed in Java. It becomes the most popular search engine and is commonly used for log analytics, full-text search, security intelligences, business analytics, and operational intelligence use cases. In particular, Elasticsearch allows for users to store, search, and analyze huge volumes of data, and then provide the answers in real-time. It relies on a NoSQL database that utilizes search technology to facilitate data retrieval and searching. It is capable of managing both structured and unstructured data, and can divide data into nodes to improve performance and accessibility. Once the raw data is parsed, normalized, and indexed by the Elasticsearch, users can run complex queries against their data and use aggregations to retrieve complex summaries of their data. As a result, by utilizing Elasticsearch as a database, users are able to efficiently search and manage large quantities of data with ease. The working principle of Elasticsearch introduces several features for usage. For example, index creation, searching, mapping for managing data relationships, time-based data management, and real-time management. When considering to the core concept of Elasticsearch, it has been designed and developed which consists of the following components (as shown in Figure 12).

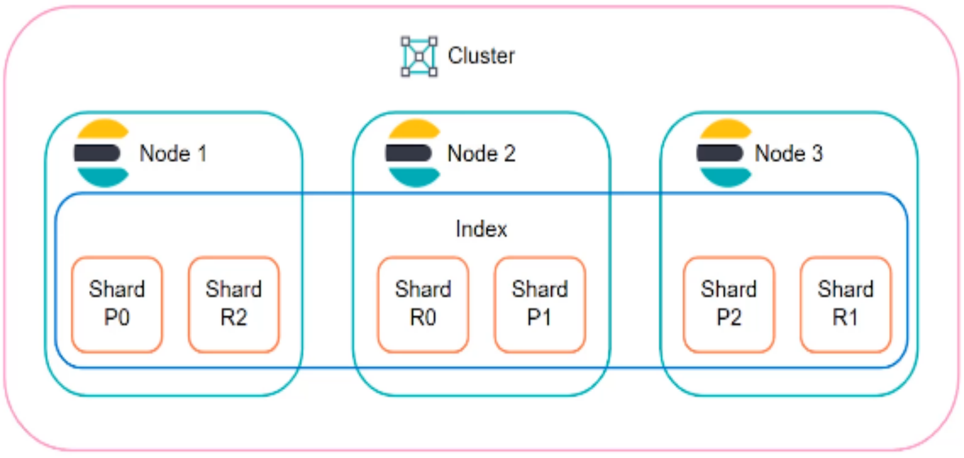


Figure 12: Overview of Elastic architecture

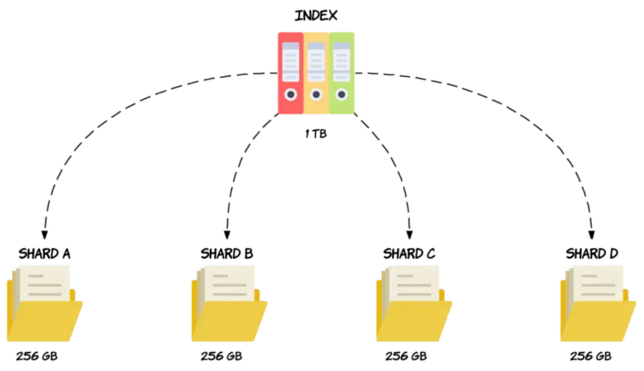


Figure 13: Elasticsearch subdivides the index into multiple pieces, called Shard.

- Node: a tool used for storing data, searching data, and performing other operations in Elasticsearch. It serves as a server or a networked machine, and each node can have multiple machines.
- Cluster: a set of Elasticsearch nodes that are grouped together to increase performance and stability. Each cluster has its own unique name and can have a large number of nodes in the same cluster.
- Index: a set of data collected and stored, which can be searched by full-text search and is the basic unit of data storage in Elasticsearch as shown in Figure 13.
- Document: data that is stored within an index, which can be searched by full-text search and is formatted as JSON, allowing for adding, deleting, and modifying.
- Field: a component of a document that defines the structure of the data in the document, with different types of fields available such as text, keyword, date, boolean, etc.
- Shards: Shards are sub-units of data that are used to improve data storage and search efficiency in Elasticsearch. Each shard is actually a self-contained index. By distributing the documents in an index across multiple shards, and distributing these shards across multiple nodes, thus this can ensure redundancy which both protects against hardware failures and increases query capacity. In particular, shards can be classified into two types: primaries and replicas.
    - Primary shard: It is the main shard that stores data and acts as the central point for searching data in Elasticsearch for each index. Each index has only one

primary shard, and when new data is added to the index, it is written to the primary shard first. Then Elasticsearch divides the data into sub-units and distributes it to replica shards.

- Replica shard: It is a copy of the primary shard that is created to increase the stability of data storage and searching in Elasticsearch. Replica shards are stored in different nodes that are members of the cluster, to distribute data storage and searching system

1.6.4 Kibana

Kibana [24] is a data visualization and exploration tool used for log and time-series analytics, application monitoring, and operational intelligence use cases. It offers a powerful tool and easy-to-use features for allowing users to visualize and analyze data stored in Elasticsearch. Its key features include dashboards and visualizations, which provide various options for presenting data such as histograms, line graphs, bar charts, pie charts, heat maps, and built-in geospatial support. It also acts as the user interface for monitoring, managing, and securing an Elastic Stack cluster. While, it provides tight integration with Elasticsearch for visualizing the indexed data. Additionally, Kibana offers alerting capabilities that allow users to set up customized alerts when certain conditions are met.

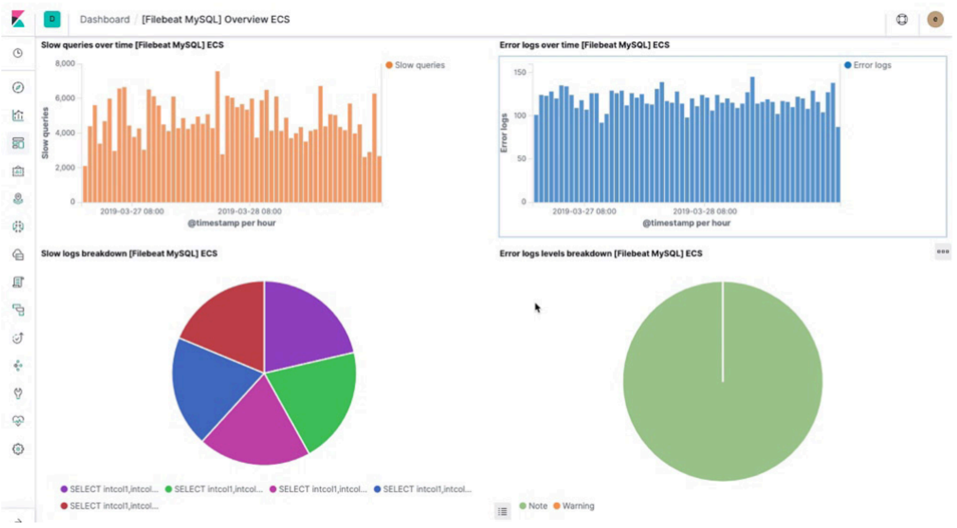

Dashboard
[Filebeat MySQL] Overview ECS
Slow queries over time [Filebeat MySQL] ECS
Slow queries
@timestamp per hour
Error logs over time [Filebeat MySQL] ECS
Error logs
@timestamp per hour
Slow logs breakdown [Filebeat MySQL] ECS
Error logs levels breakdown [Filebeat MySQL] ECS
Note
Warning

Figure 14: Example of Kibana dashboard

1.7 Functionality analysis

Table 2 and Table 3 describe the differences characteristics between the open-source tools discussed in the previous section.

Tables 2: Functionality analysis in terms of scalability, flexibility, language/run-time machine and caching

Tools	Description	Scalability	Flexibility	Language/Run-time Machine	Caching
Fluentd	Data collection and streaming platform for unified data collection	Can scale horizontally to handle large-scale deployments	Flexible and extensible architecture with a wide range of plugins	Ruby, C	Supports
Fluent Bit	Lightweight data collector and processor for cloud-native environments	Supports horizontal scaling and distributed deployments	Supports various data sources and destinations with plugin ecosystem	C	Supports
Graylog	A centralized log management platform that allows the collection, indexing, and analysis of log data. It has a built-in web interface for searching and visualizing log data.	Can handle massive amounts of log data.	Provides advanced log analysis and alerting features.	Java	Does not support
Apache Kafka	A publish-subscribe messaging system that allows for high-throughput and low-latency data streaming. It can be used for real-time data streaming and event-driven architecture.	Highly scalable and can handle high volumes of log data.	Supports real-time processing of log data.	Scala, Java	Does not support
Rsyslog	A powerful and flexible log management system. It supports input from various sources, output to various targets, and advanced filtering capabilities.	Capable of handling large amounts of log data.	Offers a broad range of log processing capabilities.	C	Supports
Elasticsearch	A search and analytics engine. It supports indexing and searching of log data for fast and efficient querying and analysis.	Highly scalable and can handle large amounts of log data.	Offers advanced search and analysis features.	Java	Does not support

Logstash	A log pipeline tool that helps to collect, parse, and store logs. It has a wide range of plugins for input and output sources	Capable of handling large amounts of log data.	Offers a wider range of log processing capabilities.	Ruby,Java	Supports

Tables 3: Functionality analysis in terms of data collection, data parsing, data transformation, data output and user-friendly web interface

Tools	Data Collection	Data Parsing	Data Transformation	Data Output	User-friendly Web Interface
Fluentd	Supports various input sources such as HTTP, Unix sockets, TCP, and more	Supports data parsing using regex, JSON, and more	Supports data transformation using filters such as grep and awk	Supports various output destinations such as file, HTTP, and more	limited web interface for monitoring and management
Fluent Bit	Supports various data sources such as logs, metrics, and events	Supports data parsing using plugins and custom configurations	Supports data transformation using filters and plugins	Supports various data destinations such as files, databases, and messaging systems	Does not have a built-in user-friendly web interface
Graylog	Supports various input sources such as Syslog, GELF, and more	Supports data parsing using built-in extractors or custom extractors	Supports data transformation using pipeline rules	Elasticsearch, Graylog Extended Log Format (GELF), and more	user-friendly web interface for searching, visualizing, and managing log data
Apache Kafka	Supports publish-subscribe messaging	Does not have built-in parsing capabilities, but data can be parsed before being sent to Apache Kafka	Does not have built-in transformation capabilities, but data can be transformed before being sent to Apache Kafka	Supports various output destinations such as Elasticsearch, Graylog Extended Log Format (GELF), and more	Does not have a built-in web interface
Rsyslog	Supports input from various sources including syslog, files, and more	Supports data parsing using regular expressions	Supports data transformation using rainers	Supports various output destinations such as file, syslog, and more	limited web interface for monitoring and management
Elasticsearch	Supports indexing and searching of log data	Does not have built-in parsing capabilities, but	Supports data transformation	Supports indexing and searching of log data	user-friendly web interface for searching and visualizing log data

		data can be parsed before being indexed in Elasticsearch	during indexing in Elasticsearch		
Logstash	Supports various input sources such as file, HTTP, and syslog	Supports data parsing using filters such as grok and date	Supports data transformation using filters such as grok and mutate	Supports various output destinations such as file, HTTP, and more	limited web interface for monitoring and management

2. Threat intelligence platform

Threat intelligence is a critical element of any Security Operations Center (SOC) as it provides vital information about potential threats and vulnerabilities. The core concept of threat intelligence platform aims to automate the collection, aggregation, and reconciliation of threat data from multiple sources in real-time to support defensive actions. The ultimate goal is to help security teams to identify the threats that relevant to their organizations by providing them with useful information on known malwares and threats, while powering efficient and accurate threat identification, investigation and response. Also, it allows security teams to easily share threat intelligence data with other stakeholders and security systems. In other words, the threat intelligence is said to be a key ingredient for cybersecurity defenders that enable decision making for threat management and mitigation. It makes possible for organizations to gain an advantage over the adversary by detecting the presence of threat actors, blocking and tackling their attacks, or degrading their infrastructure [25]. Using the threat intelligence solution, organizations can identify the threat sources that are relevant to their environments, potentially reducing the cost associated with unnecessary commercial threat feeds [26]. In this section, we aim to study in more details about a set of open source platforms for threat intelligence that have been widely used nowadays. Specifically, threat intelligence can be divided into two main parts: (1) Threat Intelligence Management and (2) Network Intrusion Detection and Prevention System (NIDPS).

2.1 Threat Intelligence Management

Threat Intelligence Management enables organizations to better understand the global threat landscape, anticipate attacker's next moves, and take prompt action to stop attacks. It is the collection, normalization, enrichment and actioning of data about potential attackers and their intentions. Also, it provides SOC analysts actionable intelligence with associated normalized risk scores and the necessary context from intelligence sources that are required in order to detect, prioritize and investigate security event. Examples of open-source tools are Malware Information Sharing Platform (MISP), Open Cyber Threat Intelligence Platform (OpenCTI), Collaborative Research Into Threats (CRITs), and Collective Intelligence Framework (CIF).

2.1.1 Malware Information Sharing Platform (MISP)

MISP (Malware Information Sharing Platform) [27-29] is an open-source platform for sharing, storing and analyzing threat intelligence data. It was created by the non-profit organization Computer Incident Response Center Luxembourg (CIRCL) and is widely used by threat intelligence analysts, incident responders, and security researchers. It provides a standardized format for sharing threat intelligence information, which enables organizations to exchange information about indicators of compromise (IOCs), malware samples, and other threat data. One of the key features of MISP is the ability to facilitate collaborative threat intelligence sharing among organizations, while providing a variety of mechanisms for securely sharing threat intelligence data between different organizations, including a distributed architecture, role-based access control, and the ability to selectively share data with specific organizations. In addition, MISP can support a wide range of data types, including network traffic captures, malware samples, and vulnerability information. It also provides a range of tools for analyzing and visualizing threat intelligence data, including heat maps, timelines, and graphs. The platform is highly customizable and can be configured to meet the specific needs of different organizations.

2.1.1.1 Data Model

In particular, MISP is not only a software but also a series of data models created by the MISP community. It includes a simple and practical information sharing format expressed in JSON that can be used with MISP software or by another other software. The MISP data model can be divided into two layers: data layer, and context layer, as shown in Figure 15.

- **Data layer:** which stores deep-level data used to exchange information related to cybersecurity threats and attacks among various organizations and users. According to this layer, there are four components involved.

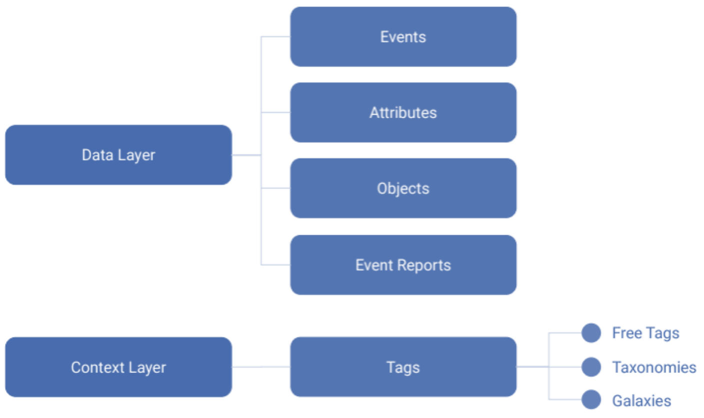


Figure 15: Type of Data model

  - *Event:* This component stores information on events or activities related to various risks or threats. For example, deep-level data that describes these events or activities. It consists of attributes and objects.
  - *Attribute:* This component stores data related to events or activities, with a key-value pair structure where the key is the name of the attribute and the value is the data for that attribute. MISP attributes can have various types, such as IP addresses, domain names, malware samples, and more.
  - *Object:* This component stores data related to events or activities that are more complex and consist of multiple related attributes. MISP objects are used when displaying data that is more complex than just attributes, such as data related to attacked applications.
  - *Event reports*: This component stores additional information related to events and various risks.
- **Context layer:** It includes various mechanisms for organizing and categorizing data, including free tags, galaxies, and taxonomies.
  - *Free tags*: These are the simplest form of contextualization, where a label or text can be set without restriction. While, they are flexible that can make automation and understanding difficult because they lack a standardized vocabulary.

- *Taxonomies*: They are a more structured way of categorizing data, which provide simple labels that are standardized on a common set of vocabularies. Taxonomies provide an efficient classification globally understood by the community and can ease consumption and automation.
- *Galaxies:* They are normalized classifications boosted by metadata, enable the description of complex high-level information. They are used internally in MISP to represent the MITRE ATT&CK (MITRE Adversarial Tactics, Techniques, and Common Knowledge) Framework, which is a globally recognized knowledge base of adversary tactics and techniques.

2.1.1.2 Sharing models

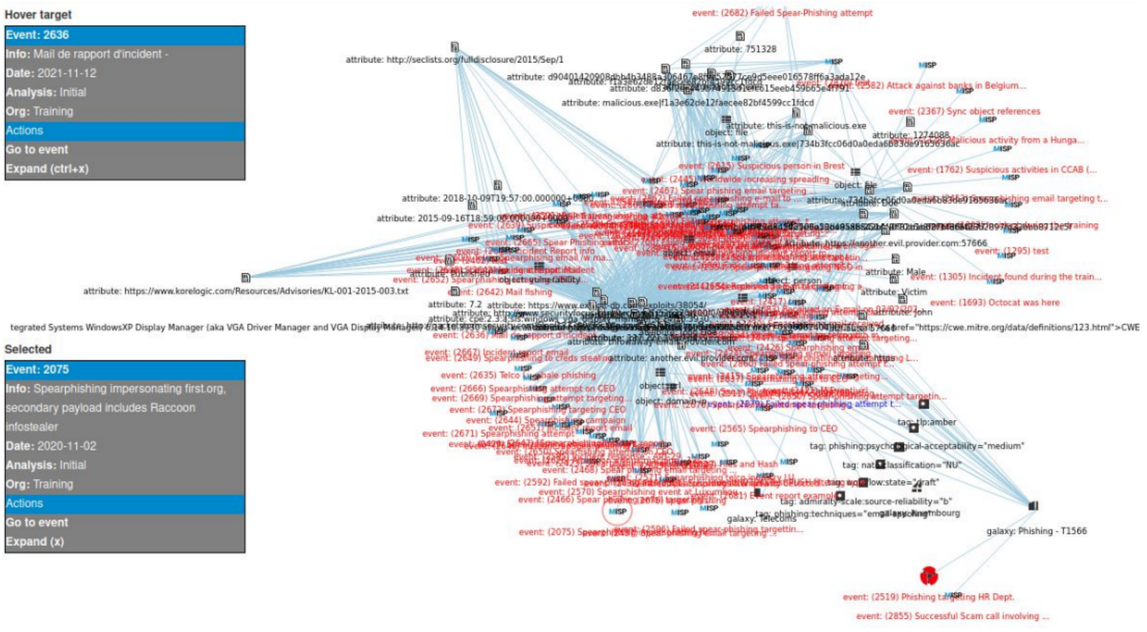
MISP has a "Sharing Models" framework for data sharing, which includes the different levels of data sharing, as follows:

- Organization Only: which limited to the organization that can access the data.
- Community Only: which limited to the community within MISP that can access the data.
- Connected Communities: It allows members of connected communities to share the data.
- All: It allows everyone in MISP to access the data.

In each level of data sharing, access privileges can be specified, such as allowing access only to authorized users in certain groups or sharing data publicly, which can be accessed without restrictions. In addition, automatic data sharing can also be set up using rule-based sharing, which automatically checks and shares data based on conditions specified in the MISP system.

2.1.1.3 Correlation

Correlation in MISP refers to the process of linking together events based on their attributes. Whenever an attribute is created or modified in MISP, links are automatically created to allow interconnection between events based on their attributes. The correlation engine in MISP is the system used to create these correlations between attribute’s values, (as show in Figure 16). In addition, it is important to correctly cluster data in MISP to ensure that correlations are accurate and meaningful. This can be done by using extended events if applicable, and splitting data per incident or based on time. By doing so, the correlations between attributes are more likely to be accurate and useful. Also, when configuring non-MISP feeds, it is important to be careful as they may not be designed to work with MISP's correlation engine. It is recommended to thoroughly test and evaluate any non-MISP feeds


Hover target
Event: 2636
Info: Mail de rapport d'incident -
Date: 2021-11-12
Analysis: Initial
Org: Training
Actions
Go to event
Expand (ctrl+x)

Selected
Event: 2075
Info: Spearphishing impersonating first.org, secondary payload includes Raccoon infostealer
Date: 2020-11-02
Analysis: Initial
Org: Training
Actions
Go to event
Expand (x)

event: (2682) Failed Spear-Phishing attempt
attribute: 751328
attribute: http://seclists.org/fulldisclosure/2015/Sep/1
attribute: malicious.exe|f1a3e62de12faecee82bf4599cc1fdcd
event: (2367) Sync object references
event: (1762) Suspicious activities in CCAB (...
attribute: https://www.korelogic.com/Resources/Advisories/KL-001-2015-003.txt
event: (2642) Mail fishing
event: (1305) Incident found during the train...
event: (1693) Octocat was here
event: (1295) test
attribute: Victim
attribute: Male
event: (2565) Spearphishing to CEO
tag: tlp:amber
tag: phishing:psychological-acceptability="medium"
tag: nato:classification="NU"
tag: workflow:state="draft"
galaxy: Phishing - T1566
event: (2519) Phishing targeting HR Dept.
event: (2855) Successful Scam call involving ...
event: (2075) Spearphishing...
event: (2635) Telco whale phishing
event: (2666) Spearphishing attempt on CEO
event: (2592) Failed spear...

Figure 16: Using MISP to create correlations

before integrating them into your MISP instance to ensure that they work properly and do not cause any issues with the correlation engine.

2.1.1.4 Data visualization

The capabilities of data visualization in MISP, are one of its key features that provide users with a variety of powerful tools to analyze and understand the data they shared. It includes several tools for data visualizations (as shown in Figure 17).

- Graph visualization: which shows the relationships between various entities in the system, such as IP addresses, domains, and malware.
- Tree visualization: it represents the hierarchical relationship between the entities.
- Heatmap visualization: it represents the distribution of a particular attribute across a dataset using colors.

- Calendar visualization: it represents events over a calendar year.
- Table visualization: which displays data in a tabular format, making it easy to compare and analyze.
- Geomap visualization: which displays geospatial data in the form of maps.
- Event Graph: it represents events and their relationships in a graph format.
- Galaxy View: it provides an overview of different entities and their relationships in a galaxy format.
- Timeline: which shows the events in chronological order over time.

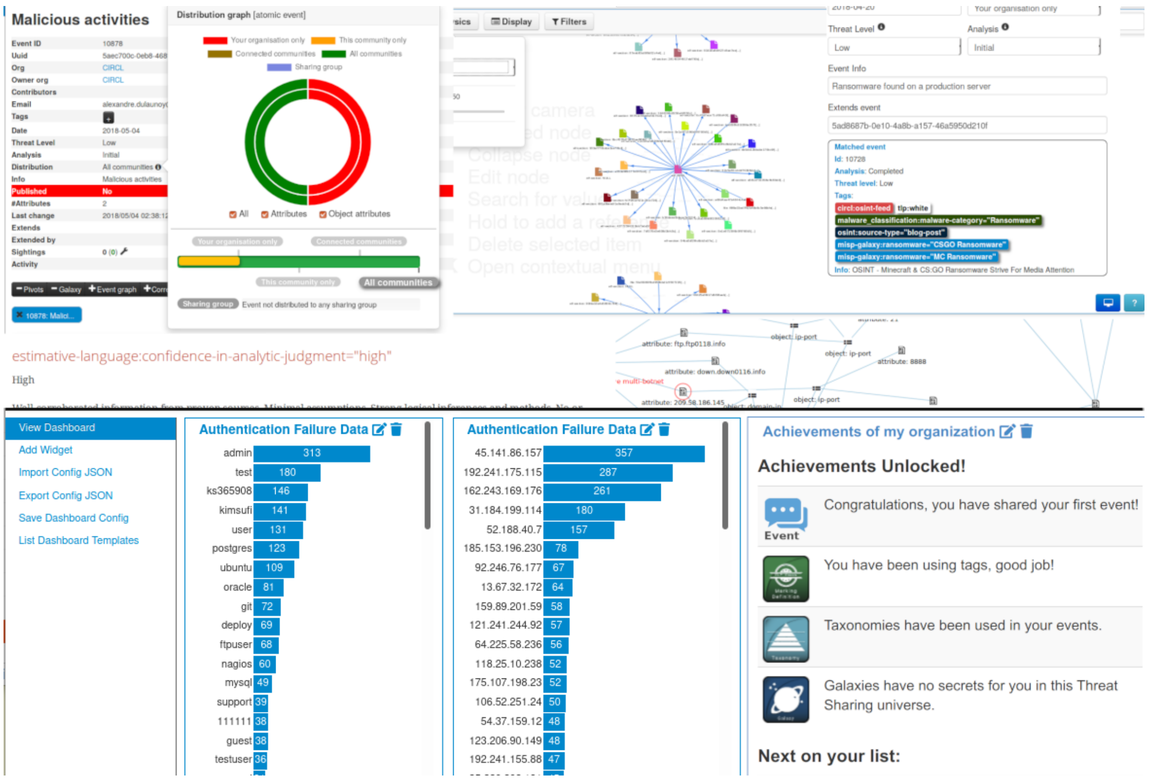


Figure 17: Data virtualization capabilities provided by MISP

2.1.2 Open Cyber Threat Intelligence Platform (OpenCTI)

OpenCTI [30-31] is a platform for processing and sharing threat intelligence knowledge. It has been developed by the French National Cybersecurity Agency (ANSSI) along with the Computer Emergency Response Team of the European Union (CERT-EU) [32]. The main purpose of the OpenCTI platform is to provide a powerful knowledge management database with an enforced schema especially tailored for cyber threat intelligence and cyber operations.

Nowadays, the platform has been fully released in open source and made available to the entire cyber threat intelligence community, by allowing the actors to structure, store, organize, visualize and share their knowledge. In addition, OpenCTI is considered as a comprehensive tool allowing users to use a variety of knowledge schemas in structuring data. All data is structured using the MITRE ATT&CK framework [33] with reference to the STIX2 standards [34]. The data can later be exported in the form of STIX2 bundles, CSV files, and other supported formats. It performs its basic operations by finding the links that exist between the available information. Therefore, it can effectively trace the root source of the given information. Also, it has been designed as a modern web application including a GraphQL API [35] and an UX oriented frontend, and can be integrated with other tools and applications such as TheHive [36], MISP [37], MITRE ATT&CK [33], etc. In other words, OpenCTI has the ability to form new relations on specific data types by inferring to the already existing information. It relies on a number of databases to perform its functions and all these are connected by GraphQL API which allows the brokers to interact with the available databases. Through this platform cyber specialists have been able to improve the tactics deployed when dealing with cyber security threats. With the available information in its archives, cyber security experts can now effectively know how to handle cyber-related security threats. This has made the tool very useful in boosting the cyber security of various organizations. To summarize, the key features of OpenCTI are discussed as follows.

- Data connection and control: OpenCTI enables users to connect and control various data formats from different sources (e.g., STIX 2, MAEC, MISP, and other CTIs) in order to merge data and build a database of the OpenCTI platform.
- Analysis and alerting: It allows users to automatically analyze data movements and create alerts for users when significant changes occur.
- Report generation and visualization: It allows users to create reports and visualize data relevant to assist in decision-making and future planning.
- Integrating with other tools: OpenCTI provides users with the ability to work with other tools. For example, contacting other tools to extract additional data or passing data to other tools for collaborative use, such as MISP and STIX-Shifter.

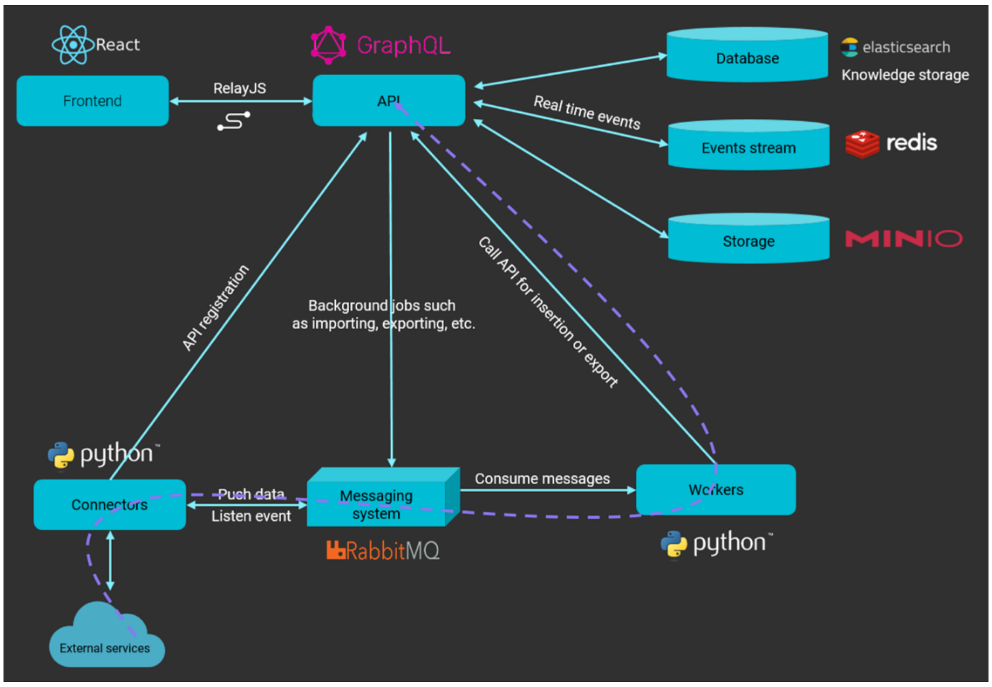


Figure 18: The overview of OpenCTI architecture

Figure 18 presents the architecture of OpenCTI [38] that consists of several main components. Initially, the React frontend uses RelayJS to communicate with an API that uses the GraphQL query language and connects to the Elasticsearch database, which serves as the primary database for storing all system data. There is also an events stream management system for recording various events that occur in the system, as well as a real-time events system that uses Redis to send data in real-time. Data storage in the system is handled by MINIO, an open source storage management program that can connect to multiple applications. In addition to these components, there are other components such as the API registration, which is a Python Connector written by users to connect their own systems to OpenCTI efficiently. Workers are standalone Python processes used for asynchronous write queries that have high speed and can call APIs for efficient data insertion or export. Third-party Connector pieces of software can also connect to external systems, such as push data or listen events through the messaging system RabbitMQ, which is used to manage background jobs such as importing, exporting, and others.

2.1.2.1 Data Model

OpenCTI utilizes a graph database as its underlying data model, which is a database model that uses the structure of a graph to store and represent data. The graph is composed of two following components

- **Nodes:** which represent the various entities or components in the OpenCTI system. Each node consists of properties and values that describe the characteristics of that entity. For example, a network node may have properties related to its IP address and connected networks, while a malware node may have the properties related to its type and behavior.
- **Edges:** which represent the relationships between nodes. Each edge connects two nodes and has types and properties that describe the nature of the relationship between these nodes. For example, an edge connects a malware node and a network node may represent the connection between the malware and the network that it targeted.

To enable a unified approach for using a graph database for the description of different kind of threats, the Structured Threat Information eXpression (STIX) database schema was developed [34]. It is a language and serialization format used to exchange cyber threat intelligence (CTI). STIX enables organizations to share CTI with others in a consistent and machine readable manner, allowing security communities to better understand what computer-based attacks they are most likely to see and to anticipate and/or respond to those attacks faster and more effectively. It is designed to improve many different capabilities, such as collaborative threat analysis, automated threat exchange, automated detection and response, etc. With STIX, all aspects of suspicious, compromise and attribution can be

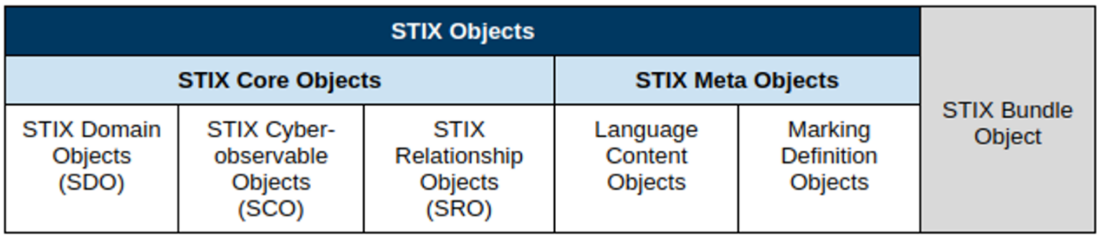

STIX Objects
STIX Core Objects
STIX Meta Objects
STIX Domain Objects (SDO)
STIX Cyber-observable Objects (SCO)
STIX Relationship Objects (SRO)
Language Content Objects
Marking Definition Objects
STIX Bundle Object

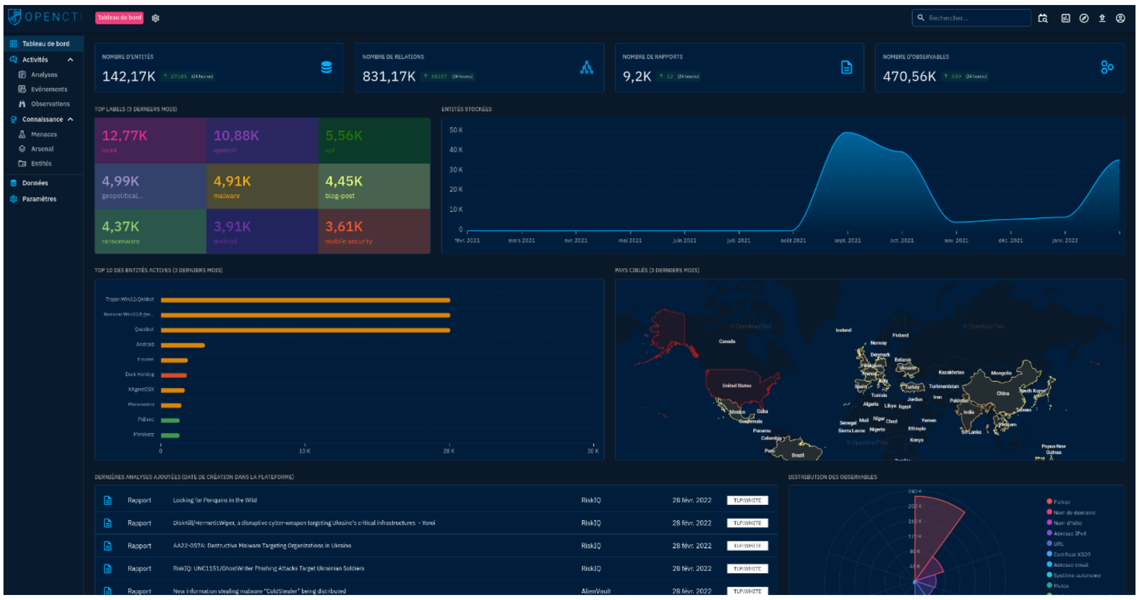
Figure 19: A list of STIX objects

represented clearly with objects and descriptive relationships. Specifically, the STIX objects are designed to be interoperable across multiple platforms and capable of connecting to various systems, allowing users to efficiently exchange risk information between systems and organizations. Also, STIX Objects support customization and updating to adapt to evolving risk situations. Figure 19 illustrated a list of STIX objects.

- **STIX Core Objects**: This is the design principle of STIX Objects. There are three subcomponents involved.
  - STIX Domain Objects (SDO): These are representatives of each organization or sub-organization that are important for sharing risk information. Examples include indicators, attack patterns, mampaigns, courses of action, intrusion sets, malware, tools, and identities.
  - STIX Cyber-observable Objects (SCO): These are data that can be used to further detect and verify attacks. For example, files, registry keys, processes, network traffic, email addresses, URLs, and user accounts.
  - STIX Relationship Objects (SRO): These are connections between data from the same organization or group, which provide meaning to the data and help users to track and analyze risks effectively. Examples include related-to, targets, indicates, uses, located-at, and part-of.
- **STIX Meta Objects**: These are STIX Objects used to provide additional information for using STIX Core Objects.
  - Language Content Objects: These used to specify the language of the data. They can be used with each STIX Core Object to provide clarity and understanding of the data.
  - Marking Definition Objects: These are STIX Objects used to define markings in STIX data, which help users to easily distinguish and process risk-related information. Markings can specify confidentiality, integrity, and trustworthiness.
- **STIX Bundle Object**: It used to bundle multiple STIX objects together into a bundle. In the latest version of STIX, STIX 2.x, the bundle object is used as a key data sharing model that can be used to exchange risk information between organizations.

2.1.2.2 Data visualization

Data visualization in OpenCTI [39] is a tool for displaying data that affects risk analysis and security verification. It provides the ability to display data in graph format, which is a crucial tool for creating different perspectives on data analysis and verification of entities related to risk and security. Figure 20 exemplified data visualization in OpenCTI platform. The key features of data visualization in OpenCTI are discussed as follows.



NOMBRE D'ENTITÉS	NOMBRE DE RELATIONS	NOMBRE DE RAPPORTS	NOMBRE D'OBSERVABLES
142,17K ↑ 27185 (24 heures)	831,17K ↑ 38237 (24 heures)	9,2K ↑ 12 (24 heures)	470,56K ↑ 339 (24 heures)

DERNIÈRES ANALYSES AJOUTÉES (DATE DE CRÉATION DANS LA PLATEFORME)

Rapport	Looking for Penquins in the Wild	RiskIQ	28 févr. 2022	TLP:WHITE
Rapport	DiskKill/HermeticWiper, a disruptive cyber-weapon targeting Ukraine's critical infrastructures - Yoroi	RiskIQ	28 févr. 2022	TLP:WHITE
Rapport	AA22-057A: Destructive Malware Targeting Organizations in Ukraine	RiskIQ	28 févr. 2022	TLP:WHITE
Rapport	RiskIQ: UNC1151/GhostWriter Phishing Attacks Target Ukrainian Soldiers	RiskIQ	28 févr. 2022	TLP:WHITE
Rapport	New information stealing malware "ColdStealer" being distributed	AlienVault	28 févr. 2022	TLP:WHITE

Figure 20: Data visualization in OpenCTI

- Entity and relationship data display: This feature allows analysts to display entity data and the relationships between variables that they want to verify easily. Users can select the entities and relationships of interest in different views.
- Victimology data comparison and analysis: This feature allows users to compare Victimology data from different attack sets. Users can easily analyze these data and find relationships between them.
- Indicator linking with threat: This feature makes it easy for analysts to link indicators with threats. Users can view related information such as source, maliciousness score, and others.
- Analysis system using dynamic widgets: OpenCTI has an analysis system that uses dynamic widgets to help users analyze data automatically and quickly.
- Display of related data in graph and bar chart format: This feature helps users understand the patterns of linkages and relationships between things they want to

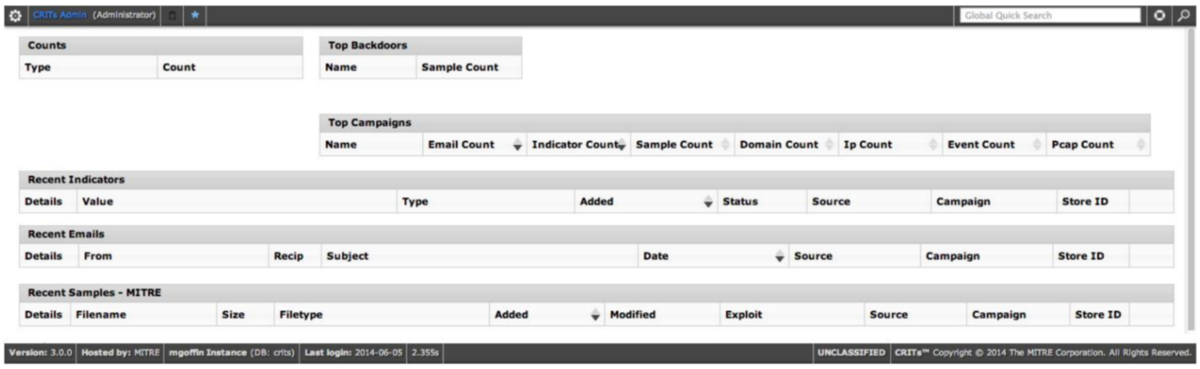
verify. For example, users can display a tree map that shows the size of data and proportions.

2.1.3 Collaborative Research Into Threats (CRITs)

CRITs [40-42], is an open-source malware and threat repository that leverages other open-source software to create a unified tool for collecting, analyzing, and sharing information about the cyber threats. It has been developed in 2010 with the purpose to give the security community a flexible and open platform for analyzing and collaborating on threat data, security events, and risks [43]. CRITs employs a hierarchy to structure cyber threat information, thus enabling the analysts to perform complex queries and discover previously unknown related content. The platform includes a flexible data model, event management system, and various tools for data analysis and sharing. It uses a non-relational MongoDB database and can be integrated with other services such as OpenDNS, ThreatExchange, and Yara. The platform covers all stages of threat intelligence operations, including preparation, processing, analysis, and dissemination. In addition, CRITs provides a web-based GUI and a command-line interface via API, thus enabling users to access the platform via a web browser. Also, CRITs enables analysts to receive, analyze, and share data from various sources, such as malware analysis tools, network sensors, and risk-related information feeds. While, it provides a flexible data model that can accommodate custom data types and features such as search, filtering, tagging, and notifications. The platform has an event management system that covers all stages of operations, allowing organizations to effectively track the lifecycle of events and follow appropriate procedures. According to [43], CRITs has been designed based on the following components.

- Web Application: The front-end interface of CRITs, which enables users and administrators to access and manage data in the system using web technologies such as HTML, CSS, JavaScript, as well as frameworks and libraries like Django, jQuery, and Bootstrap.
- Database: CRITs uses a NoSQL-based database structure and utilizes MongoDB to store information related to threats and attacks.
- Analysis engine: This component analyzes and manages threat-related data using a variety of techniques, such as data extraction, risk assessment, technical analysis, and personal analysis.

- Collection server: It is responsible for collecting and storing the threat data from different sources for analysis and management by various modules in CRITs.
- Authentication and authorization: It uses to verify and authorize user access to the system for preventing against unauthorized access and actions.
- User interface: It displays system data and allows users to interact with the data in a user-friendly and intuitive manner.
- Application Programming Interfaces (APIs): These components provide a way for CRITs to connect with other systems (e.g., importing data into CRITs or exporting data from CRITs to other systems), with the purpose to enhance the efficiency of threat and attack management.
- Reports: It allows users to generate customized reports based on their specific needs and requirements related to threat and attack management.



Counts

Type	Count

Top Backdoors

Name	Sample Count

Top Campaigns

Name	Email Count	Indicator Count	Sample Count	Domain Count	Ip Count	Event Count	Pcap Count

Recent Indicators

Details	Value	Type	Added	Status	Source	Campaign	Store ID

Recent Emails

Details	From	Recip	Subject	Date	Source	Campaign	Store ID

Recent Samples - MITRE

Details	Filename	Size	Filetype	Added	Modified	Exploit	Source	Campaign	Store ID

Version: 3.0.0 | Hosted by: MITRE | mgoffin Instance (DB: crits) | Last login: 2014-06-05 | 2.355s | UNCLASSIFIED | CRITs™ Copyright © 2014 The MITRE Corporation. All Rights Reserved.

Figure 21: CRITs dashboard

Overall, CRITs uses a collection server to store original or aggregated data from various sources for later analysis and management, and leverages analysis engine to analyze and manage risk and attack data, which can include Data Extraction, Risk Assessment, Technical Analysis, and Personal Analysis. CRITs uses MongoDB to store data related to risk and attack, with a NoSQL structure that is highly flexible for data storage. The platform also features a data visualization component to allow users to quickly and efficiently view risk and attack data within the system. Figure 21 exemplified an unpopulated CRITs dashboard and STIXv2 as the main framework.

2.1.4 Collective Intelligence Framework (CIF)

CIF [44] is a cyber threat intelligence management system that manages information related to threat intelligence. That says, it allows user to combine known malicious threat information from many sources, and use this information for identification (incident response), detection, and mitigation. Thus, the main objective of CIF is to consolidate information for supporting threat identification, detection and mitigation. It collects data and provide a simple lookup service spanning all the configured sources, as well as a way of constructing bigger lists out of all available data (output feeds). The platform has been designed to support four stages of Threat Intelligence Sharing (TIS) process, which includes preparation, collection, analysis, and dissemination. Specifically, the main idea behind the CIF is to create a tool enabling easy aggregation of data from many small data sources (input feeds) dispersed in the public Internet. This includes text files published on Pastebin-like services, website rankings (Alexa etc.), public botnet controller lists and others. CIF should not be treated as a central intelligence database, but it acts as an intermediary block in a data collection system, functionally similar to a funnel.

In [45], the authors pointed out that CIF is client/server system for sharing threat intelligence data. It includes a server component (as shown in Figure 22) which is responsible for collecting and storing the CTI data. Data can be IP addresses, ASN numbers, email addresses, domain names and Uniform Resource Locators (URLs) and other attributes. These data can be accessed via various client programs. The standard client is a Perl command line utility. A browser plugin is also available. CIF data also includes information on the type of

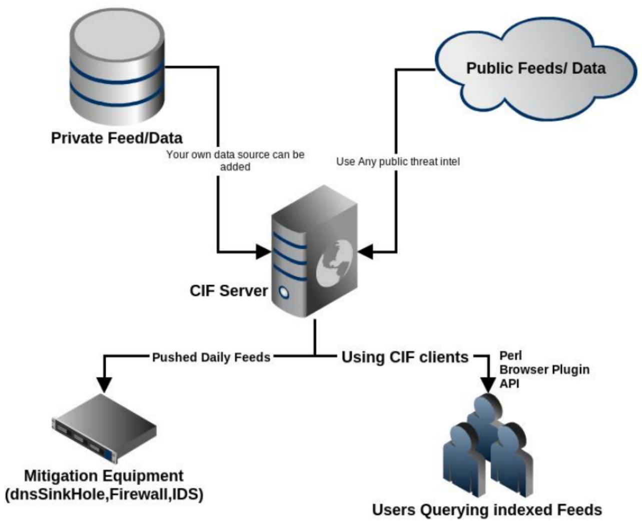


Figure 22: CIF overview

threat, severity of an attack and the confidence of the data. CIF provides the ability to control access through the use of an API-Key and the ability to place restriction levels on the data. Internally, CIF stores data using the IODEF format. It is also capable of exporting CTI for specific security tools, and can output data as Snort rules or iptables rules as well as other formats.

Moreover, CIF can import and store data from various sources, ranging from private information to public sources such as domain names, IP addresses, and URLs. This is done by importing basic information into the system via YAML configuration that supports XML [46], JSON [47], and CSV [48]. The platform uses an application called cif-smrt to download, separate, and import data into the platform. Data collection is ongoing 24 hours a day and stored in JSON format using Elasticsearch [17]. The platform also has basic functions for automatic data collection, and has threat intelligence data aggregation functions that can separate, standardize, and complement collected data. Nevertheless, CIF is a command-line tool and does not have visualization capabilities. In other words, it can leverage Kibana for graph generation [49-50]. In addition, the analysis process can work together in detail. The distribution of threat intelligence can be activated by pushing and pulling data. It can be pushed to other instances or integrated into existing security structures. Users can also retrieve data by setting filters to share the threat intelligence by selecting groups. However, there is no information on the type of filters available.

2.1.4.1 Architecture

According to [51], CIF architecture consists of the following main components.

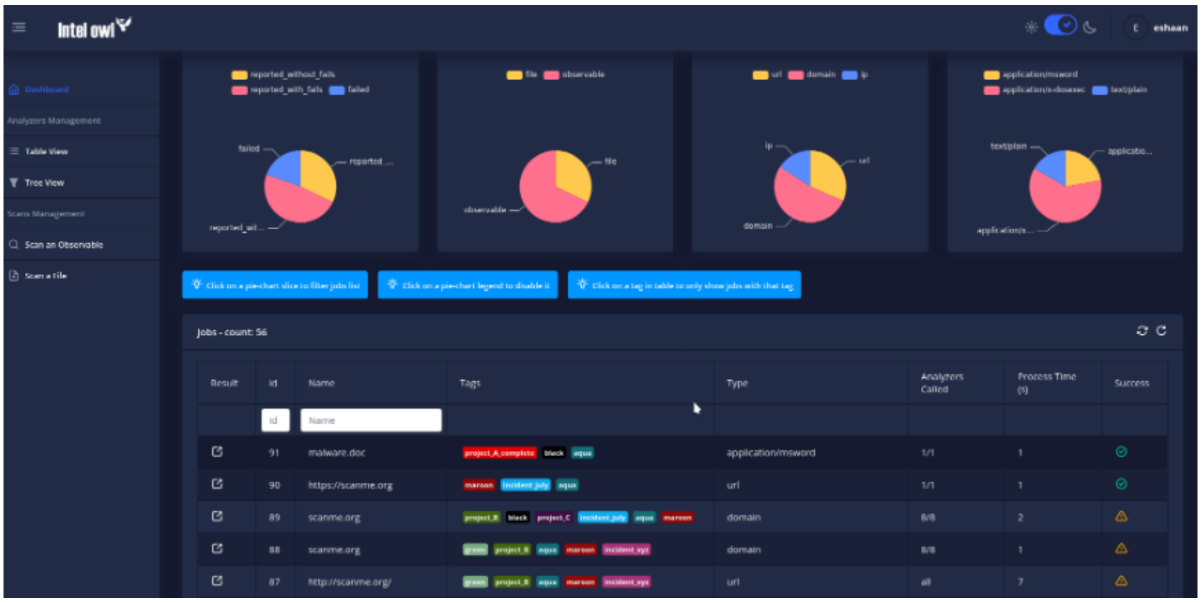
- PostgreSQL database: CIF uses a PostgreSQL database to store and manage threat intelligence data. The database schema is designed to be flexible, allowing CIF to accommodate different types of data, such as IP addresses, domains, URLs, email addresses, and file hashes.
- Apache2 server: CIF uses the Apache2 web server to host the CIF web interface and API. Apache2 is a widely used open-source web server that provides a robust and scalable platform for hosting web applications.
- PERL Code for handling HTTPS requests: CIF uses PERL code to handle HTTPS requests between the CIF web interface/API and the database. In particular, HTTPS is used to encrypt any of the underlying HTTP traffic on an HTTPs connection and secure

communications between the CIF components, ensuring that sensitive threat intelligence data is protected.

- PERL scripts for downloading data from configured sources: CIF uses PERL scripts to collect and download threat intelligence data from configured sources, such as DNS logs, firewalls, and other security tools. Examples of PERL scripts are as follows.
    - *cif-observable-retrieval.pl*: This script is used to download observable threat data from various sources. It connects to the PostgreSQL database and sends the threat data to the database for storage.
    - *cif-observable-download.pl:* Similar to the above description, this script is used to download observable threat data from various sources. It uses HTTP or HTTPS protocol to download data from the sources.
    - *cif-observable-cleanup.pl:* This script is used to remove expired observable threat data from the PostgreSQL database to prevent unnecessary data inflation.
    - *cif-asn-retrieval.pl:* It is used to download Autonomous System Number (ASN) data of IP addresses from various sources such as ARIN, RIPE NCC, and APNIC to aid in the analysis of threats associated with specific ASNs.
- CRON jobs for running scripts in a iimely manner: CIF uses CRON jobs to schedule the execution of the data collection scripts at specific times. This ensures that the CIF database is regularly updated with the latest threat intelligence data.

2.1.5 IntelOwl

IntelOwl [52-53] was designed with the intent to help the community (i.e., researchers) that can not afford commercial solution, in the generation of threat intelligence data, in a simple, scalable, and reliable way. It is an Open Source INTelligence (OSINT) solution that provides threat intelligence data about a specific file, IP addresses, or a domain from a single API. It is designed to be scalable and fast, by integrating a number of analyzers available online and a lot of cutting-edge malware analysis tools. The features of IntelOwl include threat intelligence enrichment for malware and observables such as IP, domain, URL, and hash. The application is built for scalability and speed in retrieving threat information. It includes analyzers that can retrieve data from external sources like VirusTotal or AbuseIPDB, and then generate intel from internal analyzers such as Yara or Oletools. It also includes connectors that can export data to external platforms. In particular, the main objective of Client Intel Owl is to provide a single API interface for retrieving threat intelligence at scale. There are multiple ways to interact with the Intel Owl APIs, including a built-in web interface with a dashboard (as shown in Figure 23), visualizations of analysis data, easy-to-use forms for requesting new analysis, tags management, and other features. This web interface is built using Create React App and based on certego-ui. Additionally, pyIntelOwl (CLI/SDK) can be used as a library for your own Python projects or accessed directly via the command line interface.



Jobs - count: 56

Result	Id	Name	Tags	Type	Analyzers Called	Process Time (s)	Success
	91	malware.doc	project_A_complete black aqua	application/msword	1/1	1	✓
	90	https://scanme.org	maroon incident july aqua	url	1/1	1	✓
	89	scanme.org	project_B black project_C incident_july aqua maroon	domain	8/8	2	⚠
	88	scanme.org	green project_B aqua maroon incident_xyz	domain	8/8	1	⚠
	87	http://scanme.org/	green project_B aqua maroon incident_xyz	url	all	7	⚠

Figure 23: IntelOwn dashboard

Additionally, IntelOwl has a feature for "Organizations and User" management, which includes a new "Organization" section available on the GUI. This section replaces the previous permission management via Django Admin and aims to provide an easier way to manage users and visibility. It also supports the Traffic Light Protocol (TLP) to facilitate sharing of job/analysis results in a standardized manner. The way Intel Owl facilitates threat information sharing using TLP is that each connector has a maximum TLP value (customizable from the configuration file) associated with it that gives you the control of what information is shared to the external platforms.

2.1.5.1 Main features

The main features of IntelOwl are discussed as follows.

- Modern Django-Python application: IntelOwl provides an easy way to understand and write code upon it. Django is a popular Python web framework used for developing web applications. Thus, working with IntelOwl is made easier due to its use of Django, especially when it comes to writing code for database connection, data management, and creating web pages.
- It can get data from multiple sources with a single API request. This helps users avoid writing code to call data from various sources separately, saving time and complexity in coding. IntelOwl has functions that can connect and retrieve data from multiple sources (e.g., VirusTotal, AbuseIPDB, PassiveTotal), by consolidating data from various sources, it provides a unified source of information.
- There are more than 150 available analyzers available: This allows users to generate or retrieve data about a suspicious file or observable (e.g., IP, domain).
- Built-in web interface, and written in React: IntelOwl provides features such as dashboard, visualizations of analysis data, thus resulting in easy to use forms for requesting new analysis and more.
- Official library and CLI client available on GitHub
- Built-in support for integration with other SIEM/SOAR projects using connectors, specifically aimed at Threat Sharing Platforms.
- Easily integrable with other tools: Thanks to the REST API framework and to the PyIntelOwl library, enabling IntelOwl to easily integrate with other tools.
- Easily and completely customizable, both the APIs and the analyzers.
- Compatibility with some of the AWS services.
- Fast and reliable deployment: It is possible to clone the project, set up the configuration and run the IntelOwl via docker-compose.

2.1.5.2 Plugins

Plugins are the core modular components of IntelOwl that can be easily added, changed and customized. Thus, it can been roughly be classified into three types.

- **Analyzers:** Analyzers are the most important plugins in IntelOwl. They allow to perform data extraction on the observables and/or files that users would like to analyze.
- **Connectors:** Connectors are designed to run after every successful analysis which makes them suitable for automated threat-sharing. They support integration with other SIEM/SOAR projects, specifically aimed at threat sharing platforms. Currently, there are three types of Connectors plugins available:
    - MISP: which automatically creates an event on your MISP instance, while linking the successful analysis on IntelOwl.
    - OpenCTI: which automatically creates an observable and a linked report on your OpenCTI instance, linking the successful analysis on IntelOwl.
    - YETI: which refers to "Your Everyday Threat Intelligence". It allows users to find or create an observable on YETI, and link the successful analysis on IntelOwl.
- **Playbooks:** Playbooks are designed to be easy to share sequence of running Analyzers/Connectors on a particular kind of observable. For example, If the users want to avoid re-select/re-configure a particular combination of Analyzers and Connectors together every time, then users should to create a playbook and use it instead for further time saving.

2.1.6 Functionality analysis

Table 4 is a comparison of popular open-source threat intelligence platforms, which allows users to easily compare the differences between programs. The table provides an overview of characteristics in various areas such as import and export format support, data validation and analysis, data visualization, data management and sharing, and integration with other programs. This comparison helps users choose the most appropriate program for their needs.

Tables 4: Comparison of Open-Source threat intelligence management platforms.

Feature \ Program	MISP	OpenCTI	CIF	CRITs	intelOwl
Holistic Architecture					
Use case applicability	Sharing threat intelligence, tracking threat actors, incident response	Threat intelligence management, security operations, threat hunting	Sharing threat intelligence, blocking known malicious activity	Threat intelligence management, incident response, forensics	Threat intelligence management, incident response, forensics
Adherence to 5W3H method	High	High	Low	High	High
Intelligence Process					
Import Format	STIX, OpenIOC, CSV, RPZ, MAEC	STIX, MAEC, CybOX, OpenIOC, CSV, MISP	STIX, IP, DNS, URL, YARA	STIX, CybOX, Email, CSV, PCAP	MISP, VirusTotal, JSON
Export Format	STIX, OpenIOC, CSV, RPZ, MAEC	STIX, MAEC, CybOX, OpenIOC, CSV	STIX, IP, DNS, URL, YARA	STIX, CybOX, Email, CSV, PCAP	JSON, CSV, HTML, PDF
Automatic Gathering	Using MISP feeds	Using connectors with sources or other platforms	Automatic synchronization with different sources	Connecting with various Threat Intelligence feeds	Automatic synchronization with various sources
Graphic Visualization	General and intuitive dashboard and relationship graphics	Diverse dashboards and STIXv2 based graphics	Command line interface with possible integration with visualization tool	Web-based UI and visualizations	Web-based UI and visualizations
Correlation	Automatic for every data in platform	Automatic for every data in platform	Not addressed	Not addressed	IOC correlation and enrichment
Classification	Taxonomies and galaxy clusters	Threat actors, malware families and campaigns	No	Customizable tags	Customizable tags
Integration	Cuckoo Sandbox, VirusTotal, IBM X-Force, etc.	TheHive, MISP, Maltego, etc.	Splunk, ELK, etc.	Maltego, TheHive, MISP, etc.	TheHive, MISP, Splunk, ElasticSearch, etc.
Sharing Method	API, MISP galaxy, email, web interface	Web interface, API, email	Web interface, API, email	Web interface, API, email	Web interface, API, email
Support of Collaboration	MISP galaxy, Sync and Merge functionalities	User management, sharing and collaboration capabilities	Sharing of threat intel data through the web interface	Built-in messaging system for collaboration and sharing	User management, sharing and collaboration capabilities
Analysis Capabilities	IOC detection and analysis, threat intelligence correlation and enrichment	Threat intelligence analysis, relation and impact mapping, kill chain analysis	IOC detection and enrichment, threat intelligence correlation and enrichment	Malware analysis, IOC detection and analysis, threat intelligence correlation and enrichment	IOC detection and analysis, threat intelligence correlation and enrichment
Graph Generation	Yes	Yes	Yes	Yes	Yes
License Model	AGPLv3	Apache License 2.0	GPLv3	Apache License 2.0	AGPLv3

2.2 Network Intrusion Detection and Prevention System (NIDPS)

NIDPS [54-55] is a network security technology originally built for detecting vulnerability exploits against a target application or computer. It monitors network traffic for suspicious activity and alerts when malicious activity is discovered. Snort, Suricata, and Zeek are examples of NIDPS that have the capability to detect and alert when network attacks occur, such as port scanning, network hacking, virus attacks, etc. Each program has distinct characteristics and capabilities based on their objectives and usage. They can support both IDS and IPS modes, while Zeek supporting only IDS mode [56]. The individual capabilities of each solution will be discussed in detail below.

2.2.1 Snort

Snort [57-60] is a widely recognized and extensively used Network Intrusion Detection and Prevention System (NIDPS) developed by Martin Roesch in 1998. It continues to be actively developed and supported by Cisco Talos. Snort is renowned for its lightweight design, cross-platform compatibility, and open-source nature. It utilizes a rule-based language that combines signatures, protocols, and heuristic methods to detect various forms of malicious activities on computer networks. The main objective of Snort is to analyze network traffic in real-time and identify potential security threats. Through deep packet inspection and logging capabilities on IP networks, Snort is capable of monitoring and analyzing network packets as they traverse the network [57].

One of the key features of Snort is the ability to detect and prevent a wide range of malicious activities, including Denial-of-Service (DoS) attacks, Operating System (OS) fingerprinting, brute force attacks, stealth scans, and probing attacks. By comparing network packets to a set of predefined rules, Snort can differentiate between normal network traffic and suspicious or malicious activities [58]. In addition, Snort can be configured in two main modes: (1) passive sensor and (2) inline sensor. In the first mode, Snort analyzes network traffic without being directly in the path of the main data stream. The passive mode allows for non-intrusive monitoring and analysis of network traffic. In the later mode, all streaming packets must pass through the Snort, enabling it to actively block and prevent attacks as they are detected. In this mode, Snort functions as both an Intrusion Detection System (IDS) and an Intrusion Prevention System (IPS). Figure 24 presents the architecture of Snort which consists of five main components [59-60].

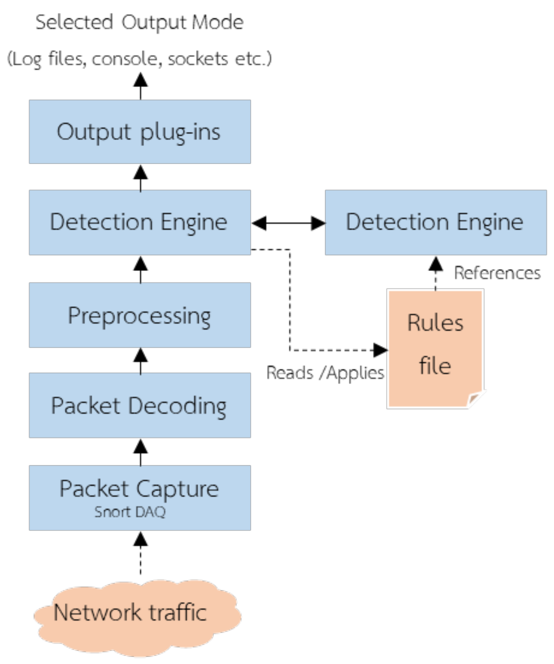


Figure 24: Snort architecture

- **Packet Capture:** Snort uses this module (a.k.a. Data Acquisition: DAQ) to capture packets that enter the network. The Snort DAQ utilizes the Libpcap library to capture packets from the Network Interface Card (NIC). The captured packets are then forwarded to the next processing module.
- **Packet Decoding:** The packets that are captured and forwarded to this module are decoded according to the structure of various network protocols to understand the information within the packets. The decoded protocols include Ethernet, IP, ICMP, TCP, UDP, and others. The decoded information is prepared for further processing.
- **Preprocessing:** It processes the decoded packets further to prepare them for detection and analysis. It performs additional checks and processing, such as TCP reassembly (reconstructing fragmented packets), tracking and managing TCP and UDP sessions, and others.
- **Detection Engine:** This module is a crucial component that detects and analyzes packets to identify unauthorized access. In IDS Snort, the Detection Engine uses rules to specify the characteristics of the observed access and the actions to be taken when packets matching the defined violation patterns are encountered. Detection and matching of violation patterns utilize feature matching techniques, including Boyer-

Moore pattern matching and PCRE Regular Expression Engine for text-based pattern matching. When the Detection Engine identifies packets matching the violation patterns defined in the rules, it triggers alert events. These events can be logged or forwarded to other notification systems such as a Security Operations Center (SOC) for further investigation and action.

- **Output Modules:** These modules are used to present the results of packet detection and analysis to system administrators or users, allowing them to be informed about unauthorized access or detected violations. The Output Modules can generate different formats of output, including:
  - Log Files: The log files can be in plain text, CSV, XML, or database formats, enabling users to review and analyze the data afterward.
  - Statistical Reports: Statistical reports provide summarized information about detected incidents and identified violation characteristics, such as the number of attacks detected within a specified time period or the most frequently observed attack patterns. Statistical reports help users adjust prevention measures and strengthen system security.
  - Alerts: Alerts can notify system administrators or users when packets matching detection rules are encountered. Alerts can be sent as messages or notifications through various channels, such as alert logs for future review and search (including information about the detected packets and related violation characteristics), network alerts using suitable network protocols like SNMP (Simple Network Management Protocol), or notification systems like email or online notification platforms.

2.2.2 Suricata

Suricata [61-66] is an Intrusion Detection and Prevention System (IDPS) that was launched in 2010 by the Open Information Security Foundation (OISF), a community network dedicated to developing open-source solutions related to network security [61]. It has been designed to analyze and inspect real-time network communication by detecting and preventing network attacks which support both signature-based detection and anomaly-based detection. This versatility allows Suricata to detect both known and unknown attacks by analyzing network packets at various layers, ranging from the data link layer to the application layer. On the one hand, it provides the capability detecting and preventing malicious activities on the network. In IDS (Intrusion Detection System) mode, it can notify and log events when an attack is detected. On the other hand, in IPS (Intrusion Prevention System) mode, Suricata can directly take action to prevent and eliminate harmful activities. It utilizes rules to detect attacks in the form of the Suricata rules language, which is compatible with Snort rules. Additionally, it offers flexibility to customize rules according to specific requirements. The detection results and reporting can be presented in various formats, such as text-based alerts or JSON [47], enabling system administrators to quickly become aware of events and respond to attacks promptly [63].

Suricata also offers a very extensive list of features. One of the important features is Suricata threading which provides the capability of running multiple threats on multi-core architecture for enhancing the processing efficiency. According to Figure 25 present the threading architecture of Suricata which consists of four threat modules.

- **Packet acquisition:** It is responsible for reading packets from the network.
- **Decode and stream application layer:** This module decodes the packets and inspects the application.
- **Detection:** After decoding, the packets are forwarded to the detection module to analyze the attacks with regards to signature-based detection. The detection process can be run in multiple threats for improving the processing efficiency.
- **Output:** Once the *detection* process is complete, it generates output which can be in the form of log files or notifications through the console. The objective is to inform system administrators about the events and enable timely responses to attacks.

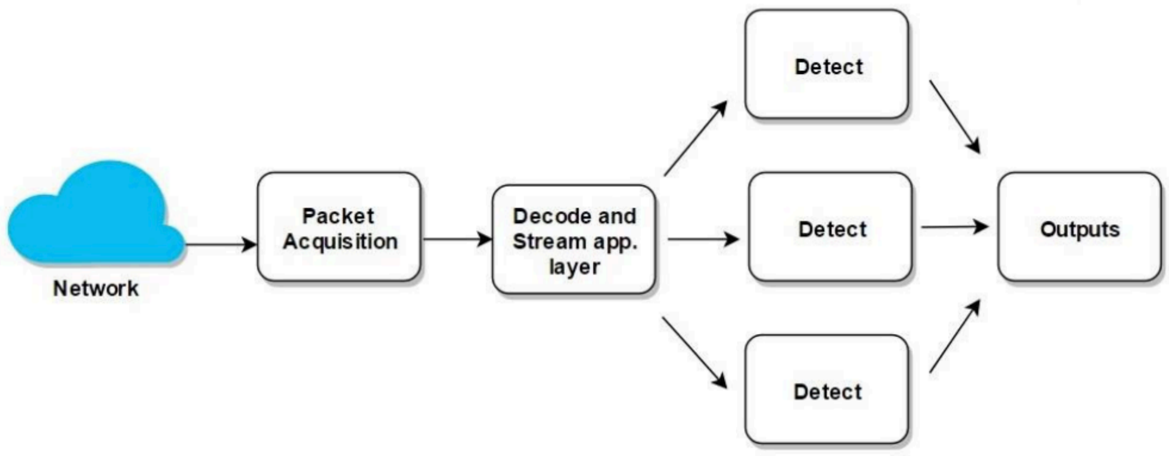


Figure 25: Threading architecture of Suricata

Furthermore, Suricata shares a similar architecture with Snort by dividing the components of preprocessors in Snort's architecture into two parts: (1) the decoding module enriches the internal representation of packets in Suricata with additional information; and (2) the detection module relies on the rules defined to detect the attacks and employs rules written in the Suricata rules language, which is compatible with Snort rules. Suricata supports rules at layer 3, layer 4, and layer 7. The rules contain signatures that match the content within packets. In addition, the detection module can be divided into multiple modules to handle packets and rule matching. If a packet matches a dangerous pattern according to the rules, it is forwarded to the output phase [64-66].

2.2.3 Zeek

Zeek [67-69] is an open-source network traffic analyzer program that operates passively. It is primarily used as a Network Security Monitor (NSM) to detect and support the investigation of suspicious or dangerous activity on the network. Zeek can also be used to analyze and troubleshoot network performance issues [66]. The program generates logs that provide a detailed description of network activity, including application-layer transcripts such as HTTP sessions, DNS requests, SSL certificates, SMTP sessions, and more. These logs can be written in tab-separated or JSON file formats, suitable for post-processing with external software. Nowadays, Zeek has been developed with a wide range of analysis and detection functions. For example, file extraction from HTTP sessions, malware detection by connecting to external registries, reporting of software vulnerabilities on the network, SSH brute-forcing detection, SSL certificate chain verification. The platform is fully customizable and extensible for traffic analysis using a Turing-complete scripting language that can be used freely. Zeek supports running on affordable hardware, making it a cost-effective solution for traffic analysis [56].

In addition to the Zeek architecture (shown in Figure 26), it can be divided into two major components: Event Engine and Policy Script Interpreter.

- **Event Engine (or Core):** It receives packets from the network and converts them into higher-level events that describe network activity in a policy-neutral manner. These events do not provide the meaning of network activity or indicate their importance. For example, every HTTP request turns into a corresponding '*http_request*' event that specifies the IP address, port, URI, and HTTP version involved. However, this event does not provide any further interpretation, e.g., whether the URI is a malware site or not [69]. In the Event Engine, it includes several subcomponents, such as (1) input sources that receive traffic from network interfaces, (2) packet analysis that examines different protocols starting from the link layer, (3) session analysis that handles application-layer protocols like HTTP and FTP, and (4) file analysis that analyzes the content of files transferred over sessions. Moreover, Event Engine also provides a plugin architecture that allows users to expand Zeek's capabilities as needed.

- **Script Interpreter:** It executes a set of event handlers written in Zeek’s custom scripting language. These scripts can express a site’s security policy (i.e., what actions to take when the monitor detects different types of activity). More generally, they can derive any desired properties and statistics from the input traffic, while maintaining the state of the script over time, enabling correlation across connections and host boundaries. Zeek's language comes with extensive domain-specific types and support functionality; and allows scripts to maintain state over time, enabling them to track and correlate the evolution of what they observe across connection and host boundaries. Zeek scripts can generate real-time alerts and also execute arbitrary external programs on demand, allowing users to trigger active responses to attacks.

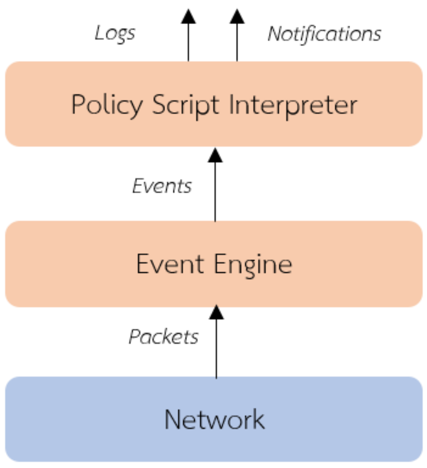


Figure 26: Zeek architecture

2.2.4 Functionality analysis

Table 5 provides a comparison of the key features of popular open-source intrusion detection and prevention software: Snort3, Suricata, and Zeek.

Table 5: Comparison of Open-Source NIDPS

Feature	Snort3	Suricata	Zeek
Software Type	IDS/IPS	IDS/IPS	IDS
Language	C	C	C++/Bro Script
Input Sources	Network tap or switch span/mirror port or pcap files	Network interface or pcap files or syslog, NetFlow, IPFIX, sFlow, FTP, DNS, HTTP, and others	Network tap or switch span/mirror port
Detection Mode	Passive mode, inline mode	Passive mode, inline mode	Passive mode
Prevention Capability	Available (requires Snort Inline or third-party plugins)	Available	Not available
Rule Usage	Can use Snort rules	Can use Suricata rules	Can use Bro Script or Suricata YAML rules
Rule Creation	Can create rules using Snort rules	Can create rules using Suricata rules	Can create rules using Bro Script
Rule Management	Snort rules language, popular and standardized	Suricata rules language, flexible and supports Snort rules	Bro Script language, which may be complex for non-programmers
Log Viewing	GUI for log viewing	GUI for log viewing	Command-line interface for log viewing
Log Recording	Capable of recording network activity logs	Capable of recording network activity logs	Capable of recording network activity logs
Performance	High-performance but resource-intensive	High-performance with lower resource usage than Snort	High-performance but resource-intensive
Detection Capability	Can detect various network attacks	Can detect various network attacks	Can detect various network attacks
Alerting	Can send email notifications or display on GUI	Can send email notifications or display on GUI	Can send email notifications or display on GUI
Encoding	Protocol-specific	Protocol-specific	Protocol-independent
Vulnerability Detection	Signature-based	Signature-based, anomaly-based	Signature-based, anomaly-based
Data Format Support	Unified2, syslog	EVE JSON, unified2	Bro logs, JSON
Advanced Detection	Not available	Custom events and packet analysis	Custom events and packet analysis
DDoS Mitigation	Not available	Supported (in inline mode)	Not available
Event Handling	Log files, alert messages, email notifications	Log files, alert messages, email notifications	Log files, alert messages, email notifications

3. Data visualization dashboard

Understanding data is the key to making the best decisions for any business. Therefore, data visualization dashboard is considered as an important part of today's analytics-driven enterprise. It provides a way to organize and display data clearly and concisely, allowing users to identify the key trends, patterns, and insights quickly. It also provides decision-makers with a high-level view of the most important metrics, by combining numbers, charts, graphs, and other graphics to represent the complex data into an easy-to-understand format [70]. This will help users to identify when problems arise in their business systems or operations, by monitoring, analyzing, displaying the key metrics, and providing data-driven decision-making to improve their organization performance. In other words, dashboards enable technical and non-technical users to understand and apply business intelligence to make better decisions [71]. Users can actively participate in the analysis process by compiling data and visualizing trends to obtain the results. It is also used to convey messages and understand patterns easily, allowing users to gain a better understanding about the relationships between the data and the key metrics of interest. One of the major benefits of using data visualization dashboard is to provide users with faster decisions and better organization performance. Thus, choosing the right program for data visualization dashboard is crucial because it can help data analysts to analyze data quickly and efficiently. With charts and visualizations in graphic and table format, it is easier to present and understand data. In this review, we will discuss in more details about the widely used tools for dashboard visualization: Apache Superset, Grafana, Metabase, and Kibana.

3.1 Apache Superset

Apache Superset [72] is an open-source software tool for data exploration and analysis that enables users to easily create and display data visualizations. Apache Superset has the ability to create various forms of visualizations, including bar charts, line charts, pie charts, histograms, scatter plots, and heatmaps. This tool provides a great way to see about the summary of all data, allowing users to quickly understand its distribution and relationship. Apache Superset also provides user-friendly interface for data exploration and analysis, and allows for the creation of dashboards that can be shared among users. It supports enterprise-ready authentication and high-granularity security/permission settings in order to protect data from unauthorized access by allowing only the users who are authorized. Apache Superset has been developed based on three main modules: (1) ***data visualization*** which creates visual representations of data in an easily understandable form; (2) ***data exploration*** which is the process of examining data from different perspectives to understand its content in new and creative ways; and (3) ***data analysis*** is the process of specifying metrics, verifying problems, and making decisions.

3.1.1 Main Features

The main features of Apache Superset are as follows.

- A rich set of data visualizations which offers a wide range of data visualization options such as bar charts, line charts, pie charts, histograms, scatter plots, heatmaps, etc. These visualizations help users to easily understand and insight about the complex data.
- Easy-to-use interface for exploring and visualizing data in order to provide user-friendly, enabling users to quickly create charts, add filters, share dashboards, and apply various settings to customize their data views.
- Enterprise-ready authentication which allows for integration with major authentication providers. For example, database, OpenID, LDAP, OAuth & REMOTE_USER through Flask AppBuilder.
- An extensible, high-granularity security/permission model for allowing intricate rules on who can access individual features and the dataset
- A simple semantic layer for allowing users to control how data sources are displayed in the UI by defining which fields should show up in which drop-down and which aggregation and function metrics are made available to the user
- Integration with most SQL-speaking RDBMS through SQLAlchemy
- The integration with Druid.io enables Apache Superset to handle complex data structures and schemas, making it a powerful tool for data analysis and exploration.

3.1.2 Databases

Apache Superset provides a wide range of functionalities for connecting to various databases and tools, seamlessly integrating with almost all major databases. This makes data visualization and analysis easier, ultimately leading to more efficient model development. Apache Superset is compatible with numerous databases such as Amazon Athena, Amazon Redshift, Apache Drill, Apache Druid, Apache Hive, Apache Impala, Apache Kylin, Apache Pinot, Apache Spark SQL, BigQuery, CockroachDB, Elasticsearch, IBM, Snowflake, SQLite, and SQL Server, among others. Other database engines that have a proper DB-API driver and SQLAlchemy dialect should also be supported by Apache Superset.

3.1.3 Types Of Visualization

Apache Superset offers a wide range of visualization types, including bar charts, line charts, area charts, pie charts, donut charts, time series charts, heatmaps, scatter plots, bubble charts, box plots, tree maps, sunburst charts, Sankey diagrams, chord diagrams, word clouds, Mapbox visualizations, scatter plot, grid, polygons, path, screen grid, and acrs. These visualizations can be customized and configured to suit specific data analysis needs. Example visualizations can be seen in Figure 27-28.

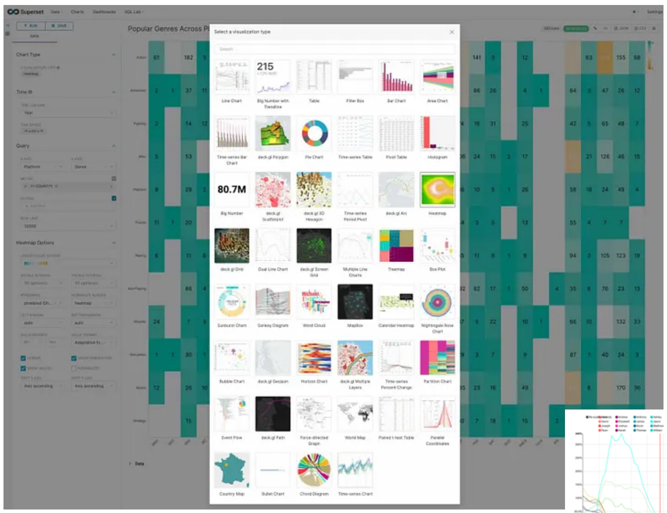


Figure 27: Type of virtualization

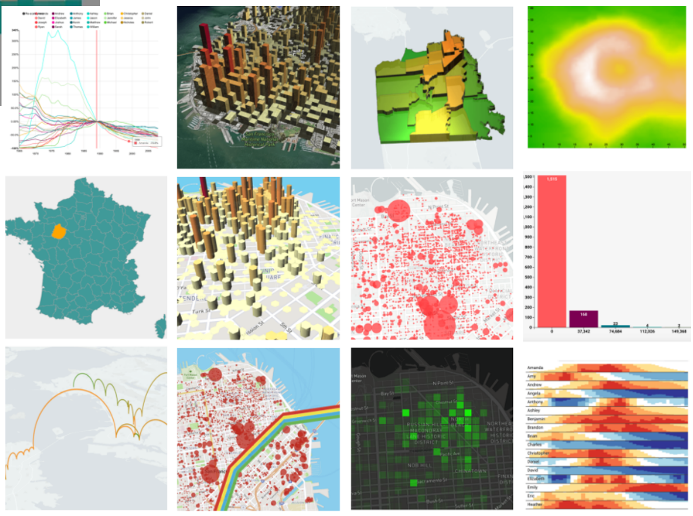


Figure 28: : Different types of Visualization offered by superset

3.2 Grafana

Grafana [73] is a powerful open-source tool that enables users to query, visualize, alert on, and explore metrics, logs, and traces from a variety of data sources. Using Grafana, users can easily transform their time-series database data into clear and insightful graphs and visualizations. Although it doesn't have built-in data collection capabilities, Grafana provides support for a wide range of data sources and offers a unique query editor for each of them, allowing for seamless integration and analysis of diverse data sets. Grafana is a running process that can be accessed via a web browser on your computer or server, providing easy and flexible access to your data. Additionally, users can configure visual alerts in their graphs, and set up Grafana to send notifications via email or other channels when alerts are triggered. The main features of Grafana include (1) user-friendly client charts and dashboard plugins for easily interpreting complex data, (2) supporting for multiple data sources with different time series, such as Graphite, InfluxDB, OpenTSDB, Prometheus, and Elasticsearch, (3) the ability to display data from multiple sources for the same business on a single dashboard, and (4) the ability to quickly share dashboards to facilitate collaboration in data analysis and troubleshooting.

3.2.1 Dashboards

A dashboard in Grafana [74-75] is a collection of one or more panels that are arranged into rows and columns. Grafana provides a wide range of panels that make it easy for users to build queries and customize visualizations according to their specific needs. In addition to these features, it includes a "*variables*" feature that allows the selection of a target in the dashboard, making it possible to use the same dashboard for multiple targets. It also offers a playlist option that allows users to create a playlist of pre-made dashboards. When the playlist is played, the dashboards change like a slideshow, with the time between each dashboard configurable. Moreover, Grafana provides several other display options, including statistical graphs that can display graphs based on statistics over a specified time period, tables for displaying data in tabular format, and 3D line charts (as shown in Figure 29). Furthermore, there are additional options called "*panel plugins*" that allow users to add additional display functions as needed. Users can also configure time settings to display data in a way that suits their needs. For example, users can specify the start and end times of the time range they wish to view, or they can specify the time range they want to display. Additionally, Grafana provides efficient display options that can be customized to meet specific needs.

3.2.2 Panels and Visualizations

The panel is the basic visualization building block in Grafana. Each panel has a query editor specific to the data source selected in the panel. The query editor allows users to build a query that returns the data they want to visualize. There are a wide variety of styling and formatting options for each panel. Panels can be dragged, dropped, and resized to rearrange them on the dashboard. Example of Gafana dashboard with different panels is shown in Figure 29. The built-in panels (shown in Figure 30) which are frequently used, can be categorized as follows:

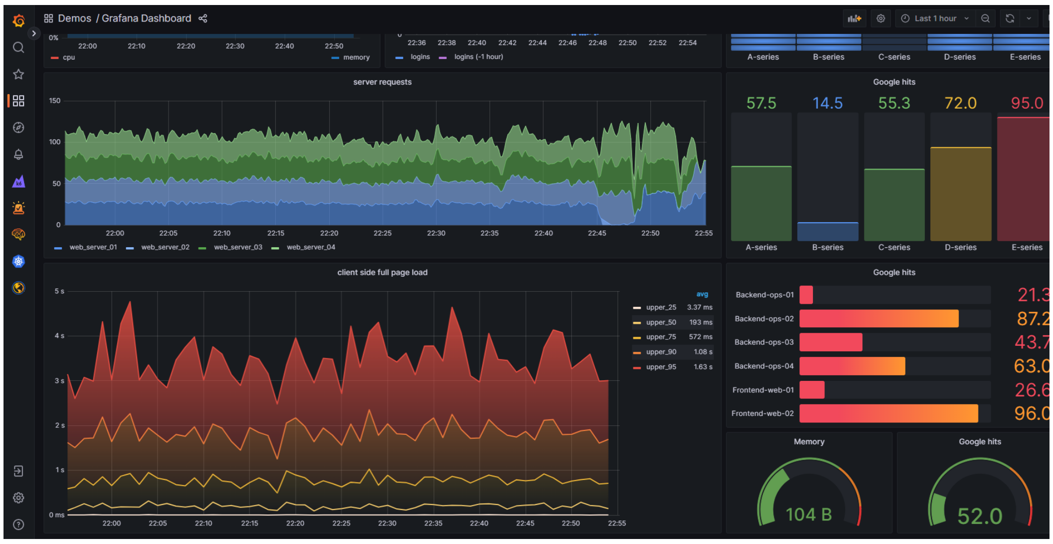


Figure 29: : Example dashboard with different panels.

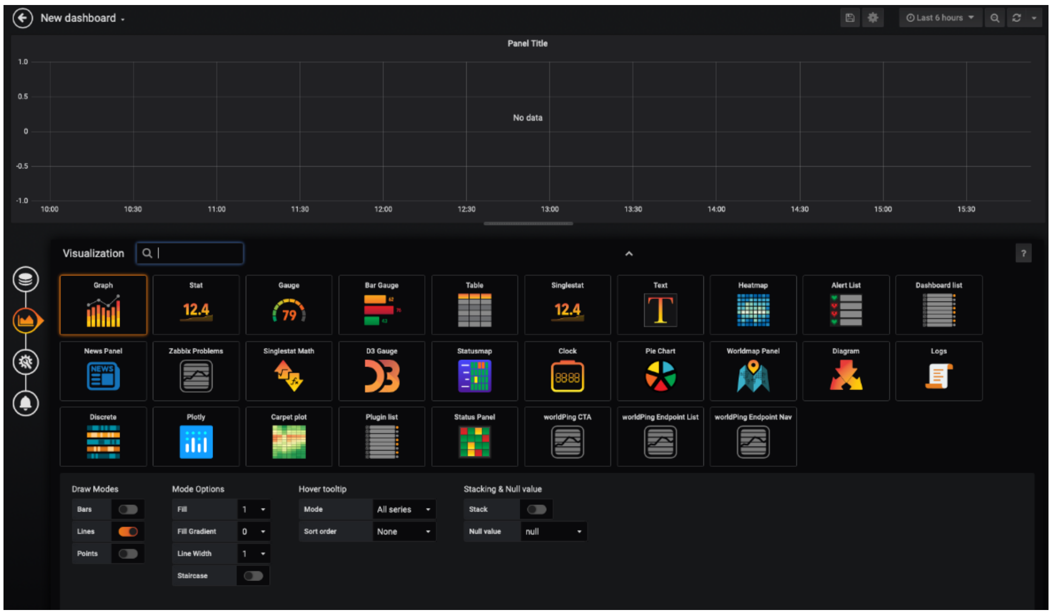


Figure 30: : : List of panel options and visualization

- Graphs & charts
    - Time series is the default and main Graph visualization.
    - State timeline for state changes over time.
    - Status history for periodic state over time.
    - Bar chart shows any categorical data.
    - Histogram calculates and shows value distribution in a bar chart.
    - Heatmap visualizes data in two dimensions, used typically for the magnitude of a phenomenon.
    - Pie chart is typically used where proportionality is important.
    - Candlestick is typically for financial data where the focus is price/data movement.
- Stats & numbers
    - Stat for big stats and optional sparkline.
    - Bar gauge is a horizontal or vertical bar gauge.
- Misc
    - Table is the main and only table visualization.
    - Logs is the main visualization for logs.
    - Node Graph for directed graphs or networks.
    - Traces is the main visualization for traces.
    - Flame Graph is the main visualization for profiling.
- Widgets
    - Dashboard list can list dashboards.
    - Alert list can list alerts.
    - Text panel can show markdown and html.
    - News panel can show RSS feeds.

In addition to the features mentioned earlier, Grafana also includes a search tool that allows users to find dashboards with names or descriptions that match specific keywords. Advanced search is also available, where users can specify different conditions such as time, tool name, or data source. Additionally, Grafana has an Alerting system that makes it easy and convenient for users to set up alert conditions. Users can set up alert conditions in various ways, such as configuring Grafana to send email notifications when a metric exceeds a specified value or when errors occur in creating or displaying dashboards. Grafana provides users with flexible alerting channels to receive notifications, including email and other channels that users can define.

3.2.2 Alert configuration and usage

Grafana has the ability to connect and display data from various sources on a dashboard, and provides tools for setting up alerts to notify users of any events that require attention. These alerts can be sent through various channels such as email, Line or Slack, and users can customize the conditions and notification methods according to their needs. To set up an alert in Grafana, users need to follow these steps:

- Alert condition: users need to define the conditions for triggering an alert in Grafana. For example, they may want to be alerted when there is an abnormal amount of access attempts, such as an increase in the number of users accessing a website beyond the daily average, or when there is access from an IP address that has never been used before. Users can use Grafana's query tools to create equations or conditions for monitoring.
- Notification method: users need to set up the notification method to send an alert message to the recipient. For example, they may want to send an email or a notification message to a mobile app. Users can customize the communication channels and details according to their needs.
- Alert settings: users need to configure the settings to enable Grafana to work according to the conditions and notification methods set. Users can customize the title and message of the alert, the status check interval, and whether to turn the alert on or off. Additionally, Grafana keeps a history of alerts and users can view the status and history of alerts on the Alert tab of the dashboard or the Alert List page.

3.3 Metabase

Metabase [76] is an open-source business intelligence and data visualization tool that enables connectivity to databases, execution of SQL queries, and creation of charts, dashboards, and reports. It is designed to be user-friendly and accessible to non-technical users. One of Metabase's key features is its ability to connect to a wide range of data sources, including SQL databases such as MySQL and PostgreSQL, cloud storage services like Amazon S3 and Google Drive, and even APIs and webhooks. This enables easy integration of data from multiple sources to create a consolidated view of business metrics. Metabase also boasts a powerful query builder that enables the creation of complex SQL queries using a visual interface, without the need to write any code. This simplifies data exploration and allows non-technical users to ask questions without relying on data analysts or developers. After data acquisition, Metabase allows for easy creation of beautiful and interactive charts and

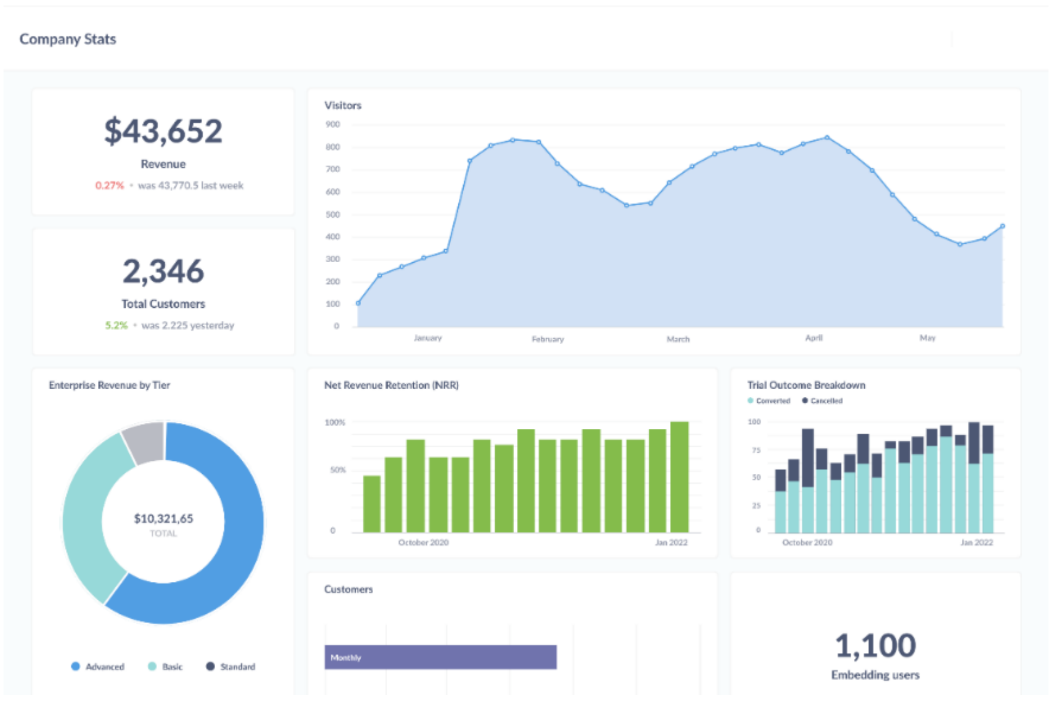


Figure 31: Metabase dashboard

dashboards. With a range of chart types, including line, bar, and pie charts, and a drag-and-drop interface for customization, users can create personalized visualizations. Additionally, interactive dashboards enable real-time data filtering and exploration. The example of Metabase dashboard is shown in Figure 31.

3.3.1 Main Features

The main features of Metabase can be described as follows.

- Database connectivity: This feature allows users to connect to the desired database, such as MySQL, PostgreSQL, Amazon Redshift, Google BigQuery, MongoDB, Presto, Athena, Snowflake, Oracle, and others, enabling users to access data as needed.
- Query Building: This feature allows users to easily create SQL queries using the user interface without having to write SQL themselves, making it easier for users to understand the data.
- Visualization: The visualization feature in Metabase allows users to create comprehensive graphs and dashboards that display data accurately. Users can select from various chart formats, such as line charts, bar charts, pie charts, and others, and customize them according to their preferences, including color, size, position, format, and even export them as images or embed them in websites and applications.
- Real-time data display: The real-time data display feature enables users to display data in real-time using the dashboard, allowing them to monitor changes in their data promptly. The auto-refresh feature updates and displays new data automatically at a set interval, such as every 5 seconds or 10 minutes. Furthermore, Metabase supports real-time data display using the Streaming API connector, which connects to the streaming system of each database, such as Apache Kafka or AWS Kinesis, to receive and display data automatically. The Streaming API reduces latency when displaying data during changes in the system.
- Permission: Metabase also offers a robust permission setting feature that enables users to control data access rights for different users. Users can assign access rights at various levels, such as viewing only their own dashboards or accessing specific rows in a database table. Users can manage access rights by defining roles and permissions in Metabase, where roles are user groups, and permissions are access rights assigned to each role as determined by the user.

3.4 Kibana

Kibana [18,24,77] is an open-source data visualization and exploration tool designed to work with Elasticsearch. It allows users to create interactive dashboards and visualizations to analyze data and gain insights. With Kibana, users can easily navigate through large data sets and create various types of charts such as line graphs, bar graphs, and pie charts. It works in conjunction with Elasticsearch and Logstash form the ELK stack, which is commonly used to manage and analyze large volumes of log data. Logstash collects data from various sources and sends it to Elasticsearch, which then serves as a database where data can be queried and retrieved by Kibana for visualization. In addition to its visualization capabilities, Kibana offers features such as index pattern management, mapping, and search functionality. It provide user-friendly interfaces, making easy for users to customize and configure their dashboards according to their specific needs. This feature makes Kibana a powerful tool for data analysis and visualization that can help organizations to obtain a better-informed decision based on their data.

When considering the Kibana visualizations, it has been designed and can be integrated with Elasticsearch. Kibana simply supplies the UI to use these aggregations for defining the different dimensions on display in visualizations. There are two types of aggregations: (1) bucket aggregations groups documents together in one bucket according to your logic and requirements; and (2) metric aggregations are used to calculate a value for each bucket based on the documents inside the bucket. Each visualization type presents buckets and their values in different ways. So in a pie chart for example, the number of slices is defined by the Buckets aggregation while the size of the slice is defined by the Metric aggregation. Kibana supports quite a large number of Elasticsearch aggregation types, each with specific configuration options and field type limitations. Histogram and Date Histogram Bucket aggregations, for example, will only work on integers. The Min and Max Metric aggregations will only work on number or date fields while the Unique Count Metric aggregation works on any field type.

3.4.1 Kibana features

Kibana offers several features which are described as follows.

- Visualization: Kibana has a lot of ways to visualize data in an easy way. Some of the ones which are commonly used are vertical bar chart, horizontal bar chart, pie chart, line graph, heat map etc.
- Dashboard: When we have the visualizations ready, all of them can be placed on one board- the Dashboard. Observing different sections together gives you a clear overall idea about what exactly is happening.

- Dev Tools: User can work with your indexes using dev tools. Beginners can add dummy indexes from dev tools and also add, update, delete the data and use the indexes to create visualization.
- Reports: All the data in the form of visualization and dashboard can be converted to reports (CSV format), embedded in the code or in the form of URLs to be shared with others.
- Filters and Search query: Users can make use of filters and search queries to get the required details for a particular input from a dashboard or visualization tool.
- Plugins: You can add third party plugins to add some new visualization or also other UI addition in Kibana.
- Coordinate and Region Maps: A coordinate and region map in Kibana helps to show the visualization on the geographical map giving a realistic view of the data.
- Timelion: It also called as timeline is yet another visualization tool which is mainly used for time based data analysis. To work with timeline, we need to use simple expression language which helps us connect to the index and also perform calculations on the data to obtain the results we need. It helps more in comparison of data to the previous cycle in terms of week , month etc.
- Canvas: Canvas is yet another powerful feature in Kibana. Using canvas visualization, you can represent your data in different colour combinations, shapes, texts, multiple pages basically called as workpad.

3.4.2 Types Of Visualization

Visualizations in Kibana are categorized into five different types (as shown in Figure 32), which are grouped based on their characteristics and usage. These types include:

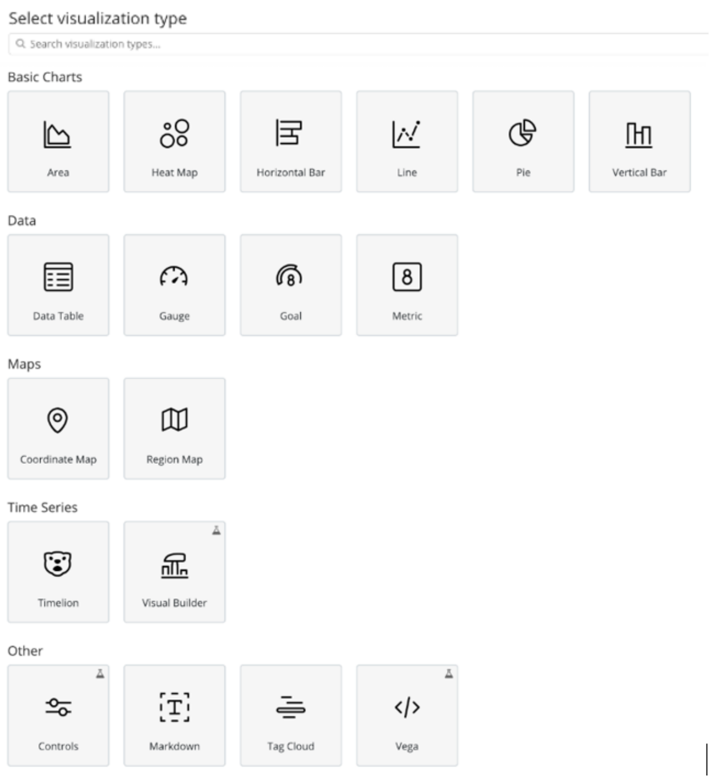


Figure 32: Type of visualization

- Basic Charts: This category includes commonly used visualizations such as Area, Heat Map, Horizontal, Bar, Line, Pie, and Vertical Bar, which are used to display data in various forms, especially related to trends and data relationships.
- Data: This category includes visualizations related to numeric data, such as Data Table, Gauge, Goal, and Metric, used to display numerical data and data relationships.
- Map: This category includes visualizations related to maps, such as Coordinate Map and Region Map, used to display data according to geographic coordinates and different regions.
- Time Series: This category includes visualizations related to time-based data, such as Timelion and Visual Builder, used to display trends and changes in data over a specified period of time.
- Other Visualizations This category includes other visualization types, such as Markdown, Tag Cloud, Vega, Vega-Lite, and Graph, used to display data in different forms and formats that are distinct from the other visualization types.

3.4.3 Security Analytics

Security analytics with Kibana is the use of Kibana to analyze and monitor security-related data, using logs and Elasticsearch stored in an ELK stack, to investigate attack details and manage the results of the received data analysis. This increases the efficiency in responding to security risks in an organization. Kibana can be used in security analytics through the following functions.

- Dashboard: Use the Dashboard to display information related to security, such as the number of detected attacks and attack statistics.
- Search: Use the Search function to find log data related to security, such as searching for IP addresses that have unauthorized access.
- Filter: Use the Filter function to filter log data related to security, such as selecting logs that contain keywords like "attack" or "malware."
- Visualization: Use Visualization to present log data in an easy-to-analyze format, such as using Pie Chart to show attack statistics by attack type.
- Machine Learning: Use Machine Learning to automatically analyze and verify log data. Machine Learning uses log data to create a model to detect and identify security risks, separating logs into anomalies and normal data.

After the model has been created, it can be used to check log data in real-time automatically. Machine Learning examines each log item to determine whether it falls into the risk group based on the created model. If there is any risk, the system will display an alert to the administrator, allowing them to respond to the risk promptly. The alerting system in Kibana includes three primary components.

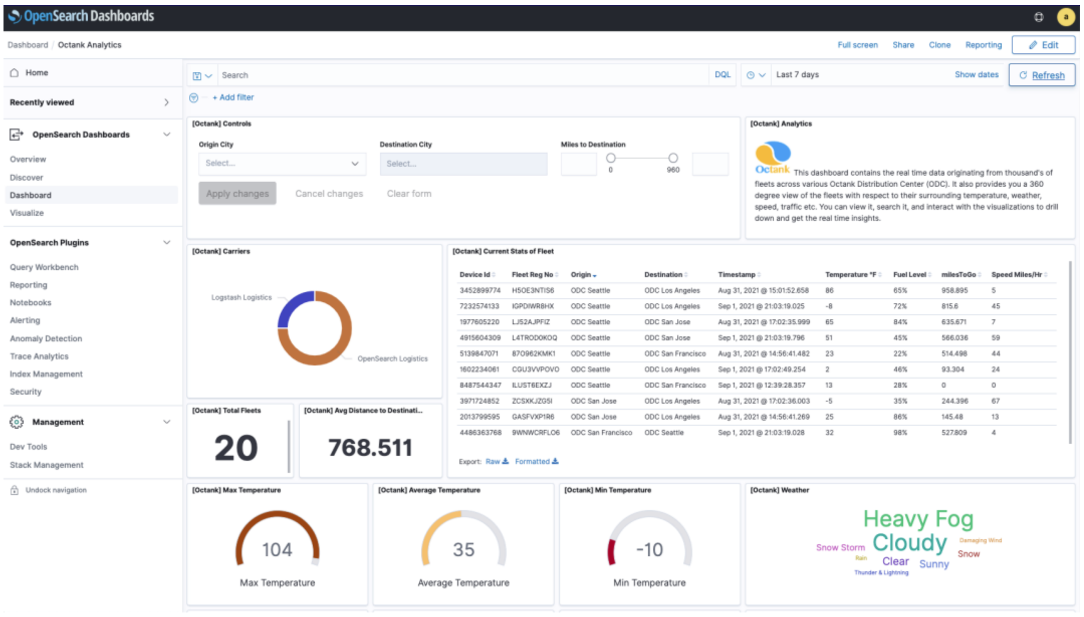
- Alerting Framework: Collects and manages notification channel and condition settings for alerts.
- Notification Channels: Notification channels that users can set up, such as email, Slack, or webhook, are used to send alert messages to designated users.
- Alerting rules: The conditions set up to monitor data in Elasticsearch. When the established conditions are met, the system sends an alert to the designated notification channel.

Once alert conditions are set up, Kibana constantly monitors related data, and Machine Learning automatically improves its model to enhance the accuracy of risk detection in the future.

3.5 OpenSearch

Dashboards are indeed a valuable tool in OpenSearch [78-79] for visualizing data without the need for extensive coding. They provide a user-friendly interface for creating and displaying visualizations and interactive charts based on the data stored in OpenSearch. With dashboards, users can easily explore and analyze data, create informative visual representations, and gain insights without the requirement of coding a complete framework from scratch. Figure 33 and Figure 34 exemplified the OpenSearch dashboards and several types of visualizations that can be utilized within the dashboards.

- Area controls: Visual representations that allow users to interact with a set of controls affecting other visualizations on the dashboard.
- Coordinate map: It displays data on a map, allowing users to visualize spatial patterns and relationships.
- Data table: Data table presents data in a tabular format, making it easy to view and analyze structured information.

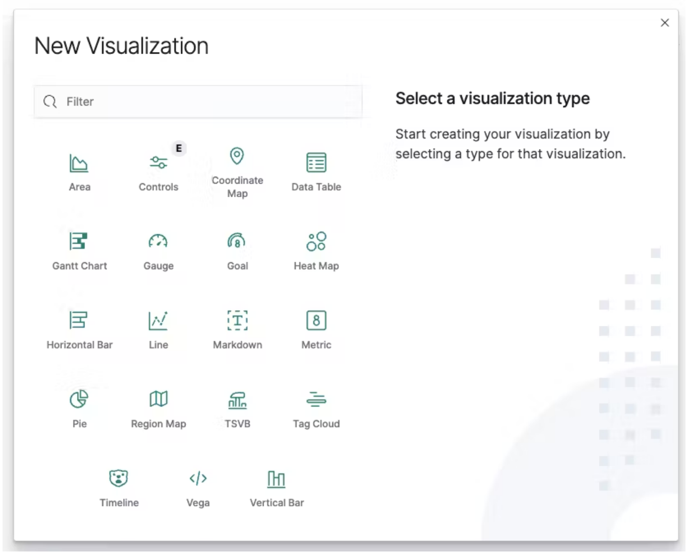


This dashboard contains the real time data originating from thousand's of fleets across various Octank Distribution Center (ODC). It also provides you a 360 degree view of the fleets with respect to their surrounding temperature, weather, speed, traffic etc. You can view it, search it, and interact with the visualizations to drill down and get the real time insights.

[Octank] Current Stats of Fleet

Device Id	Fleet Reg No	Origin	Destination	Timestamp	Temperature °F	Fuel Level	milesToGo	Speed Miles/Hr
3452899774	H5OE3NTIS6	ODC Seattle	ODC Los Angeles	Aug 31, 2021 @ 15:01:52.658	86	65%	958.895	5
7232574133	IGPDIWR8HX	ODC Seattle	ODC Los Angeles	Sep 1, 2021 @ 21:03:19.025	-8	72%	815.6	45
1977605220	LJ52AJPFIZ	ODC Seattle	ODC San Jose	Aug 31, 2021 @ 17:02:35.999	65	84%	635.671	7
4915604309	L4TRODOKOQ	ODC Seattle	ODC San Jose	Sep 1, 2021 @ 21:03:19.796	51	45%	566.036	59
5139847071	87O962KMK1	ODC Seattle	ODC San Francisco	Aug 31, 2021 @ 14:56:41.482	23	22%	514.498	44
1602234061	CGU3VVPOVO	ODC Seattle	ODC Los Angeles	Sep 1, 2021 @ 17:02:49.254	2	46%	93.304	24
8487544347	ILUST6EXZJ	ODC Seattle	ODC San Francisco	Sep 1, 2021 @ 12:39:28.357	13	28%	0	0
3971724852	ZCSXKJZG5I	ODC San Jose	ODC Los Angeles	Aug 31, 2021 @ 17:02:36.003	-5	35%	244.396	67
2013799595	GASFVXP1R6	ODC San Jose	ODC Los Angeles	Aug 31, 2021 @ 14:56:41.269	25	86%	145.48	13
4486363768	9WNWCRFLO6	ODC San Francisco	ODC Seattle	Sep 1, 2021 @ 21:03:19.028	32	98%	527.809	4

Figure 33: OpenSearch dashboards



Select a visualization type

Start creating your visualization by selecting a type for that visualization.

Figure 34:Types of visualization

- Gantt chart: which shows the duration and scheduling of tasks or events over a specific timeframe.
- Gauge: It used to represent a measurement, typically displaying a single value within a defined range.
- Goal: The objective is to visualize progress towards a specific target or objective.
- Heat map: It depicts data using color intensity to highlight patterns and variations.
- Horizontal bar: which displays categorical data using horizontal bars to compare values across different categories.

- Line: which illustrates data trends over time, showing the relationship between variables.
- Markdown: Enabling users to include text and formatting within the dashboard to provide explanations or additional information.
- Metric: It shows a single value or measurement, often used to display key performance indicators (KPIs).
- Pie: which represents data as a circular chart divided into slices, representing proportions or percentages.
- Region map: which displays data on a geographic map, emphasizing different regions or areas.
- TSVB (Timelion, Time Series Visual Builder): OpenSearch fffers advanced time series visualizations, such as forecasting and anomaly detection.
- Tag cloud: which represents text data where the importance or frequency of terms is depicted using font size or color.
- Timeline: which provides a chronological representation of events or activities.
- Vega: Enabling users to create custom visualizations using the Vega specification language.
- Vertical bar: Similar to the horizontal bar charts, which displays categories vertically.

3.6 Functionality analysis

Table 6 provides key features comparison between Apache superset, Grafana, Metabase, Kibana, and OpenSearch.

Table 6: Functionality analysis for different open sources data virtualization platforms.

Feature	Apache Superset	Grafana	Metabase	Kibana	OpenSearch
License Model	Apache License 2.0	AGPLv3	AGPLv3	Elastic License	Apache License 2.0
Data sources	Supports various data sources such as SQL databases, Druid, and CSV files	Supports various data sources including Graphite, Prometheus, Elasticsearch, and InfluxDB	Supports data sources such as MySQL, PostgreSQL, and MongoDB	Supports data sources such as Elasticsearch	OpenSearch supports data sources such as Elasticsearch
Data Visualization	Provides a variety of visualization options such as charts, tables, heatmaps, and geospatial maps	Offers a wide range of visualization options including graphs, charts, gauges, and heatmaps	Offers visualization options through charts and tables and provides a simple drag-and-drop dashboard builder	Offers visualization options through graphs, charts, and tables, and is particularly useful for analyzing log data from Elasticsearch	Offers visualization options through graphs, charts, and tables, and is particularly useful for analyzing log data from Elasticsearch
Data Filtering	Allows users to filter data through a range of options such as sliders, checkboxes, and date pickers	Provides a flexible filtering system that allows users to filter data by time range, tag, and field value	Provides a range of filtering options such as filters, segments, and custom questions	Provides a powerful filtering system that allows users to filter data through a range of options including queries and time ranges	Provides a powerful filtering system that allows users to filter data through a range of options including queries and time ranges
Alerting	Has built-in alerting and notification system that sends notifications through email or Slack	Has a built-in alerting system that can send notifications through email, PagerDuty, and other third-party integrations	Has built-in alerting that can send notifications through email or Slack	Has built-in alerting that can send notifications through email and webhooks	Has built-in alerting that can send notifications through email and webhooks
Data Comparison	Offers the ability to compare data through time-series or scatter plots	Supports data comparison through multiple data sources and time ranges	Does not support data comparison	Does not support data comparison	Does not support data comparison

Report Creation	Allows users to create reports using a simple drag-and-drop interface	Allows users to create customized reports using Grafana panels and templates	Allows users to create reports through a range of options including questions, dashboards, and pulses	Allows users to create reports through a range of options including dashboards, visualizations, and canvas	Allows users to create reports through a range of options including dashboards, visualizations, and canvas
Ad-hoc Querying	Provides ad-hoc querying capabilities through SQL Lab and Druid	Provides ad-hoc querying through Grafana's Explore feature	Provides ad-hoc querying through a custom SQL editor and by using the "Ask a question" feature	Provides ad-hoc querying through the Query Bar and the Discover feature	Provides ad-hoc querying through the Query Bar and the Discover feature
Machine learning support	Superset does not have direct machine learning support. However, it can connect to databases with ML data, enabling users to manage and display the data through the platform.C16	Grafana Enterprise, offered by Grafana Labs, provides machine learning support for predictive analytics and anomaly detection. Supports Prometheus, Graphite, Elasticsearch, InfluxDB, and OpenTSDB data sources.	Metabase does not have direct machine learning support. However, it can connect to databases with ML data, enabling users to manage and display the data through the platform.	Kibana offers machine learning support through its X-Pack feature for automated analysis and prediction, such as anomaly detection and model forecasting. This feature is only available in the paid version of X-Pack. The free version does not include this feature.	OpenSearch does not have direct machine learning support. However, it can connect to databases with ML data, enabling users to manage and display the data through the platform.

4. Signal alerting service

A Signal alerting service is a critical tool for managing computer systems and networks, providing valuable benefits in notifying system administrators of important events on the system. For example, cyber attacks or system hacks. With the ability to manage large amounts of data from multiple sources in a Security Information and Event Management (SIEM) system and alert system administrators to important events, system administrators can be informed of significant events immediately and manage them as quickly as possible. The programs used to create Signal alerting services have different capabilities and features. They can be explained as follows:

4.1 ElastAlert

ElastAlert [80-81] is an open-source tool that provides a practical example for detecting and alerting on events in Elasticsearch. Elasticsearch is a high-performance, distributed, NoSQL database and search engine. It is developed using the Elasticsearch open command and written in Python to enable users to define rules or conditions for monitoring data in Elasticsearch and receive notifications when these conditions are met. For example, it can detect unauthorized access attempts in Elasticsearch, identify log patterns, or capture values that exceed specified limits. Additionally, ElastAlert adopts a YAML-like rule format, making it convenient for users to define monitoring rules and conditions. When ElastAlert detects a match based on the defined rules, it passes the match dictionary to the specified enhancements. This allows users to easily customize or enrich the notification before sending it to the desired alerting channels. For instance, additional data can be added from other data sources, data formats can be modified, or the alert message content can be customized as show in figure 35 [80].

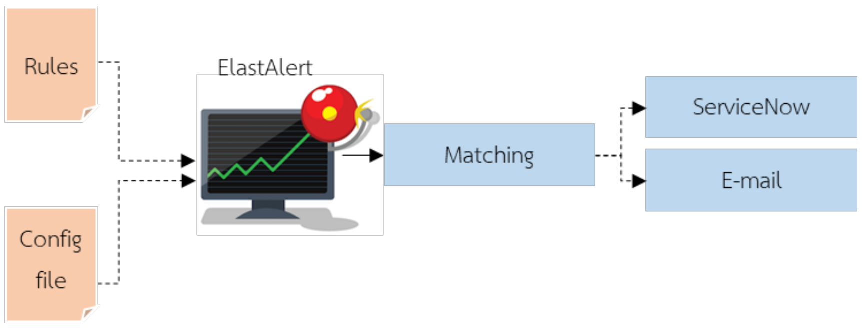


Figure 35: ElastAlert workflow

The workflow of ElastAlert [80] involves using Elasticsearch along with two essential components: rule types and alerts. Elasticsearch is queried periodically to check data within the specified timeframe. The returned data is then processed by the defined rule types to determine matches based on the specified search conditions. Once matches are found, they are sent to the designated alerting system or multiple systems for further action based on the specified conditions. Also, ElastAlert supports various rule types that are familiar and available in ElastAlert. For example, frequency type (i.e., match where there are X events in Y time), spike type (i.e., match when the rate of events increases or decreases), flatline type (i.e., match when there are less than X events in Y time). Additionally, it provides support for a wide range of alerting channels, including Command, Email, JIRA, OpsGenie, SNS, HipChat, Stride, MS Teams, Slack, Mattermost, Telegram, Google Chat, PagerDuty, PagerTree, Exotel, Twilio, VictorOps, Gitter, ServiceNow, Debug, Alerta, Stomp, theHive, HTTP POST, Alerter, Line Notify, Zabbix, and more. Users can easily import or customize rule types and alerts to meet specific requirements. ElastAlert offers flexibility and extensibility, allowing users to tailor their alerting workflows accordingly. Additionally, it provides additional features to enhance the usefulness of alerts, such as linking alerts to Kibana dashboards, aggregating field values, generating periodic reports, separating alerts using unique key fields, and intercepting and enhancing the matching data. The following are some examples of alert types commonly encountered in ElastAlert:

- Frequency alert: This type of alert checks for the occurrence of events within a specified time period and generates an alert when events meet the defined conditions. For instance, detecting excessive unauthorized access attempts within a short period of time.
- Flatline alert: This alert monitors activity or events that deviate significantly from the normal baseline and generates an alert when no events or significant changes occur within a specified time frame. For example, detecting a sudden drop in the frequency of important server activities.
- Blacklist/Whitelist alert: This alert type checks data against a list of allowed or disallowed items (blacklist/whitelist) and generates an alert when the data matches the defined list. For instance, detecting the presence of dangerous filenames in a list of operations.

- Spike alert: This alert type detects events that exhibit a significant increase within a specified time period and generates an alert when there is a rapid surge. For example, detecting a notable surge in Distributed Denial of Service (DDoS) attack requests within a short period.
- Any/Threshold alert: This alert type checks for events that meet or exceed a defined threshold or criteria and generates an alert when the conditions are met. For example, detecting more than three instances of incorrect password attempts within a specified time frame.
- Composite alert: This alert type involves creating complex rules and conditions by combining multiple data elements to detect and generate alerts. For example, detecting abnormal access attempts from multiple locations along with suspicious login attempts using compromised user accounts.

4.2 Alertmanager

Alertmanager [82-83] is an open-source program developed by the Prometheus project team, which is a system for event and alert management in IT infrastructure systems. Alertmanager is a part of the Prometheus ecosystem and is used to receive and manage alerts sent by event detection and system status monitoring systems in Prometheus. Typically, Alertmanager is used to connect with Prometheus and receive alert data from Prometheus and connected event detection systems [82]. Then, Alertmanager handles these alerts and sends them to recipients or communication channels such as email, Slack, PagerDuty, or custom-defined webhooks. Alertmanager supports deduplication of redundant alerts and repetitive notifications. Additionally, it supports grouping, silencing, and customizable alert template configuration, providing flexibility in managing alerts. When Prometheus detects

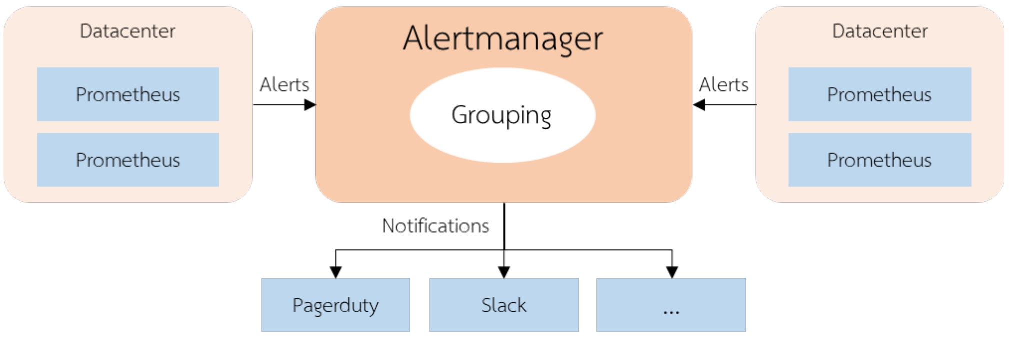


Figure 36: The architecture of Alertmanager with multiple Prometheus Servers

abnormal or exceeding values in the collected metrics, Alertmanager takes responsibility for deduplicating, grouping, and sending appropriate notifications to recipients or notification channels. The separation between monitoring (Prometheus) and alerting (Alertmanager) enables efficient tracking and management of notifications in complex environments, as depicted in Figure 36 [83].

- Alertmanager Server: It serves as the core component of Alertmanager, responsible for receiving alert data and executing defined rules and configurations. This central server manages incoming alert data and processes the alert status. The Alertmanager server can be configured to operate in a single-node mode or high-availability mode, where multiple Alertmanager servers work together to handle failures and ensure availability.
- Alerting Rules: Alerting rules are crucial in Alertmanager as they define conditions and the order of alerts based on their importance. Alerting rules utilize a Boolean expression language to evaluate events and select appropriate alerting scopes.
- Notification Channels: Alertmanager supports multiple diverse notification channels, including email, Slack, PagerDuty, webhooks, and more. Users can configure and customize notification channels according to their organizational requirements.
- Grouping: Alertmanager can group closely occurring alerts to reduce redundant notifications and enhance alert management efficiency. Grouping can be done based on user-defined criteria, such as event sources (e.g., alerts from different sources), event importance levels, or predefined notification plans.
- Silencing: Alertmanager features silencing capabilities that help reduce unnecessary or repetitive notifications. Users can define a duration to silence specific alert conditions and prevent notifications during specific time periods.
- Notification Templates: Alertmanager allows users to define and customize notification formats and templates. Users can tailor the content of notifications, such as text, links, or additional information, to be displayed in the notifications.

4.3 Shinken

Shinken [84] is an open-source monitoring tool written in Python under the GNU Affero General Public License. It is designed to notify users or system administrators about critical events or issues detected by the monitoring system. When a problem is detected, such as a service outage or a performance degradation, Shinken Signal sends alerts through various channels such as email, SMS, or integrations with messaging platforms like Slack or PagerDuty.

The architecture of Shinken follows a distributed and modular design, allowing for scalability, flexibility, and customization. Here are the key components and their roles in the Shinken architecture:

- Shinken Core: The core is the central component responsible for coordinating all monitoring activities. It handles the scheduling of checks, event processing, and notification distribution. It also manages the configuration files and communicates with other components.
- Pollers: Pollers are responsible for executing monitoring checks. They collect data from hosts and services by running plugins or scripts and send the results back to the core for processing. Pollers can be distributed across multiple machines to handle a large number of checks efficiently.
- Reactionners: Reactionners handle the event-driven actions triggered by the monitoring system. They process events such as service alerts, downtime scheduling, or rechecks. Reactionners can perform actions like sending notifications, executing recovery scripts, or triggering automated responses.
- Arbiters: Arbiters manage the configuration and monitoring objects. They validate the configuration files, resolve dependencies between hosts and services, and distribute the configuration changes to the relevant components. They ensure that the monitoring setup remains consistent and up to date.
- Brokers: Brokers handle the communication between the core and external systems. They receive monitoring events, metrics, and status updates from the core and forward them to external systems for processing or visualization. Brokers can integrate with various tools like databases, message queues, or dashboard applications.
- Web UI: Shinken provides a web-based user interface for monitoring administration and visualization. The UI allows users to configure monitoring objects, view status information, acknowledge alerts, and access historical data. It provides a convenient way to manage the monitoring infrastructure.
- Modules: Shinken supports various modules that extend its functionality. These modules can provide additional checks, integrations with external tools, or custom

actions. Modules are developed separately and can be easily integrated into the Shinken framework.

The architecture of Shinken adheres to the Unix philosophy of "one tool, one task." Each component in Shinken is designed to fulfill a specific role and operates independently, communicating with other components through standardized interfaces. Shinken utilizes an HTTP backend, which simplifies the creation of highly available and distributed monitoring architectures. In summary, Shinken's architecture follows:

- Shinken gets data IN: passively, actively and Networked API
- Shinken acts on the data: Correlation, Event suppression, Event handlers, Adding new poller daemons, and Runtime interaction
- Shinken gets data OUT: Networked API, Notifications, Logging, Web/Mobile Frontend (via API and Native WebUI), and Metrics databases
- Shinken manages configurations: Discovery manager SkonfUI, Multi-level discovery engine, Configuration Packs (commands, config templates, graph templates, etc.), and Text file management via configuration engines (cfengine, chef, puppet, salt)

4.4 Functionality analysis

Table 7 provides key features comparison between ElastAlert, Alertmanager, and Shinken.

Tables 7: The features comparison between ElastAlert, Alertmanager, and Shinken.

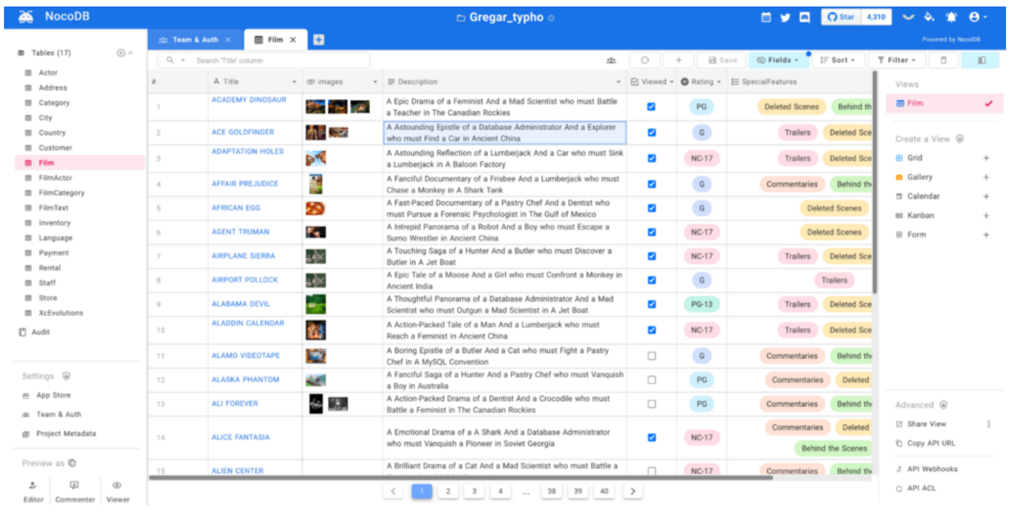
Program	ElastAlert	Alertmanager	Shinken
Monitoring	✓	-	✓
Alerting	✓	✓	✓
Integration Options	✓	✓	✓
Scalability	✓	✓	✓
Community Support	✓	✓	✓
Centralized Management	-	✓	✓
Signal Alerting	✓	✓	✓
Rule-based Alerts	✓	✓	✓
Real-time Notifications	✓	✓	✓
Multiple Notification Channels	✓	✓	✓
Customizable Alerts	✓	✓	✓
Scheduled Alerts	✓	✓	✓

Alert Escalation	✓	✓	✓
Integration Options	✓	✓	✓
Open Source	✓	✓	✓
Community Support	✓	✓	✓
Alert types	Command, Email, JIRA, OpsGenie, SNS, HipChat, Slack,Mattermost, Telegram, GoogleChat, PagerDuty,PagerTree,Exotel,Debug, Stomp, theHive,HTTP POST, Alerter,Line Notify,Zabbix	Email, Webhook, PagerDuty, Slack, OpsGenie, VictorOps, Pushover, Telegram, Discord, Microsoft Teams, Talk, WeChat, Hangouts, Line, Threema, Gitter, IRC, Mattermost, Rocket.Chat, SMS	Email, JIRA, Slack, OpsGenie, Telegram, AWS SNS, Microsoft Teams, PagerDuty, ServiceNow, Azure Logic Apps, VictorOps, Webhook, Splunk, Elasticsearch, Custom Actions

5. Database Management Systems

5.1 NocoDB

NocoDB [85-86] is a no-code database platform that enables teams to collaborate and build applications easily using a familiar spreadsheet interface. It can connect to any relational database and transform those databases into intelligent spreadsheet interfaces. This allows you to create no-code applications collaboratively with your team. Currently, NocoDB can work with databases such as MySQL, PostgreSQL, Microsoft SQL Server, SQLite. Additionally, the NocoDB app store allows you to create business workflows by integrating applications with Slack, Microsoft Teams, Discord, Twilio, WhatsApp, Email, and third-party APIs. It also



#	Title	images	Description	Viewed	Rating	SpecialFeatures
1	ACADEMY DINOSAUR		A Epic Drama of a Feminist And a Mad Scientist who must Battle a Teacher in The Canadian Rockies	✓	PG	Deleted Scenes, Behind th
2	ACE GOLDFINGER		A Astounding Epistle of a Database Administrator And a Explorer who must Find a Car in Ancient China	✓	G	Trailers, Deleted Sce
3	ADAPTATION HOLES		A Astounding Reflection of a Lumberjack And a Car who must Sink a Lumberjack in A Baloon Factory	✓	NC-17	Trailers, Deleted Sce
4	AFFAIR PREJUDICE		A Fanciful Documentary of a Frisbee And a Lumberjack who must Chase a Monkey in A Shark Tank	✓	G	Commentaries, Behind th
5	AFRICAN EGG		A Fast-Paced Documentary of a Pastry Chef And a Dentist who must Pursue a Forensic Psychologist in The Gulf of Mexico	✓	G	Deleted Scenes
6	AGENT TRUMAN		A Intrepid Panorama of a Robot And a Boy who must Escape a Sumo Wrestler in Ancient China	✓	NC-17	Deleted Scenes
7	AIRPLANE SIERRA		A Touching Saga of a Hunter And a Butler who must Discover a Butler in A Jet Boat	✓	NC-17	Trailers, Deleted Sce
8	AIRPORT POLLOCK		A Epic Tale of a Moose And a Girl who must Confront a Monkey in Ancient India	✓	G	Trailers
9	ALABAMA DEVIL		A Thoughtful Panorama of a Database Administrator And a Mad Scientist who must Outgun a Mad Scientist in A Jet Boat	✓	PG-13	Trailers, Deleted Sce
10	ALADDIN CALENDAR		A Action-Packed Tale of a Man And a Lumberjack who must Reach a Feminist in Ancient China	✓	NC-17	Trailers, Deleted Sce
11	ALAMO VIDEOTAPE		A Boring Epistle of a Butler And a Cat who must Fight a Pastry Chef in A MySQL Convention	☐	G	Commentaries, Behind th
12	ALASKA PHANTOM		A Fanciful Saga of a Hunter And a Pastry Chef who must Vanquish a Boy in Australia	☐	PG	Commentaries, Deleted
13	ALI FOREVER		A Action-Packed Drama of a Dentist And a Crocodile who must Battle a Feminist in The Canadian Rockies	☐	PG	Commentaries, Behind th
14	ALICE FANTASIA		A Emotional Drama of a A Shark And a Database Administrator who must Vanquish a Pioneer in Soviet Georgia	✓	NC-17	Commentaries, Deleted, Behind the Scenes
15	ALIEN CENTER		A Brilliant Drama of a Cat And a Mad Scientist who must Battle a	☐	NC-17	Commentaries, Behind th

< 1 2 3 4 ... 38 39 40 >

Figure 37: NocoDB Dashboard

provides programmatic access via APIs for you to create integrations with Zapier/Integromat and custom applications (as shown in Figure. 37). The following are the lists of NocoDB main features.

- **Database Creation:** Create databases using popular SQL databases such as MySQL, PostgreSQL, and SQLite. Additionally, it also supports NoSQL databases like MongoDB.
- **Data Structure Design:** You can design the structure of your database using a drag-and-drop interface. You can define tables, establish relationships between tables, and specify various field types such as text, number, date, file, and more.
- **Data Input and Management:** Provides a user-friendly form builder to create custom input forms for data entry. It allows you to easily add, edit, and delete records.
- **Multiple View Formats:** Supports different views of your data, including grid view, gallery view, form view, and kanban view, among others. These views determine how your database is displayed.
- **View Permissions:** Can define different permissions for views in NocoDB. Collaborative views allow multiple users to work together on the same view, while locked views restrict access to only specified users or roles.
- **Database/View Sharing:** Can share your databases or views in NocoDB publicly or privately with password protection. This gives you control over who can access and view your database application.
- **Various Cell Types:** Offers a variety of cell types for different data fields, such as ID, LinkToAnotherRecord, Lookup, Rollup, Single Line Text, and more.
- **Role-based Access Control:** Provides fine-grained access control at different levels. You can assign roles to users and define their access privileges. This allows you to authorize or restrict access to different features and data.
- **App Store for Workflow Automation:** NocoDB has an app store that offers integration with other systems. It falls into three main categories:
    - Chat : Slack, Discord, Mattermost, and etc
    - Email : AWS SES, SMTP, MailerSend, and etc
    - Storage : AWS S3, Google Cloud Storage, Minio, and etc

- **Programmatic Access:** NocoDB allows access through programs using its REST API. You can use these APIs to perform actions and interact with your NocoDB application programmatically. Additionally, NocoDB provides SDKs for system integration and development purposes.

References

[1] S. Kowtha, L. A. Nolan and R. A. Daley, "Cyber security operations center characterization model and analysis," *2012 IEEE Conference on Technologies for Homeland Security (HST)*, Waltham, MA, USA, 2012, pp. 470-475, doi: 10.1109/THS.2012.6459894.

[2] M. Vielberth, F. Böhm, I. Fichtinger and G. Pernul, "Security Operations Center: A Systematic Study and Open Challenges," in *IEEE Access*, vol. 8, pp. 227756-227779, 2020, doi: 10.1109/ACCESS.2020.3045514.

[3] N. Miloslavskaya, "Security Operations Centers for Information Security Incident Management," *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*, Vienna, Austria, 2016, pp. 131-136, doi: 10.1109/FiCloud.2016.26.

[4] A. Perera, S. Rathnayaka, N. D. Perera, W. W. Madushanka and A. N. Senarathne, "The Next Gen Security Operation Center," *2021 6th International Conference for Convergence in Technology (I2CT)*, Maharashtra, India, 2021, pp. 1-9, doi: 10.1109/I2CT51068.2021.9418136.

[5] F. Ahmed, U. Jahangir, H. Rahim, K. Ali and D. -e. -S. Agha, "Centralized Log Management Using Elasticsearch, Logstash and Kibana," *2020 International Conference on Information Science and Communication Technology (ICISCT)*, Karachi, Pakistan, 2020, pp. 1-7, doi: 10.1109/ICISCT49550.2020.9080053.

[6] V. D. Kumar, R. R. Shah and A. Philip, "Centralized log management for pepper," *2011 IEEE Third International Conference on Cloud Computing Technology and Science*, Athens, Greece, 2011, pp. 1-3, doi: 10.1109/CloudCom.2011.128.

[7] Fluentd. "Output plugin overview." Fluentd documentation. [Online]. Available: https://docs.fluentd.org/output. [Accessed: Feb 22, 2023].

[8] Fluent Bit. "Fluentd and Fluent Bit." [Online]. Available: https://docs.fluentbit.io/manual/about/fluentd-and-fluent-bit. [Accessed: Jun 1, 2023].

[9] Graylog. "Planning Your Deployment." [Online]. Available: https://go2docs.graylog.org/5-0/planning_your_deployment/planning_your_deployment.html. [Accessed: Feb. 19, 2023].

[10] Apache Kafka. "Documentation." Apache Kafka, 2019. [Online]. Available: https://kafka.apache.org/documentation/. [Accessed: Feb 20, 2023].

[11]

IBM Cloud Architecture. (n.d.). Kafka Overview. Retrieved February 23, 2023, from https://ibm-cloud-architecture.github.io/refarch-eda/technology/kafka-overview/

[12] Cloudera, Inc. "Apache Kafka - Overview." Hortonworks Data Platform, version 2.6.1. [Online]. Available: https://docs.cloudera.com/HDPDocuments/HDP2/HDP-2.6.1/bk_kafka-component-guide/content/ch_overview_kafka.html. [Accessed: Feb 20, 2023].

[13] R. Singh, "Getting Started with rsyslog in Linux," GeeksforGeeks, 17 March 2021. [Online]. Available: https://www.geeksforgeeks.org/getting-started-with-rsyslog-in-linux/. [Accessed: 22 February 2023].

[14] Rainer Gerhards. (2021). rsyslog Architecture [Image], Available: https://www.rsyslog.com/. [Accessed: Feb 19, 2023].

[15] J. Partain, "Rsyslog Plugin Overview," [Online]. Available: https://www.rsyslog.com/plugin-overview/. [Accessed: Feb 20, 2023].

[16] Elastic Stack, “Welcome to Elastic Docs”, [Online]. Available: https://www.elastic.co/elastic-stack/. [Accessed: Feb. 28, 2023].

[17] M. A. Fathahillah, "An Overview on Elasticsearch and its Usage," Towards Data Science, 12 September 2021. [Online]. Available: https://towardsdatascience.com/an-overview-on-elasticsearch-and-its-usage-e26df1d1d24a. [Accessed: Feb 20, 2023].

[18] A. Verma, "ELK (Elasticsearch, Logstash, Kibana) Conceptual Tutorial for Beginners," Medium, 19 October 2020. [Online]. Available: https://faun.pub/elk-elasticsearch-logstash-kibana-conceptual-tutorial-for-beginners-2a7a827305b8. [Accessed: Feb 19, 2023].

[19] DBI services. "Elastic (ELK) Stack: Elasticsearch Terminologies". dbi services Blog, 15 June 2018. [Online]. Available: https://www.dbi-services.com/blog/elastic-elk-stack-elasticsearch-terminologies/. [Accessed: Feb 18, 2023].

[20] Elastic Beats, [Online]. Available: https://www.objectrocket.com/resource/what-are-elasticsearch-beats/. [Accessed: Feb 27, 2023]

[21] Elastic Logstash, [Online]. Available: https://www.elastic.co/logstash/. [Accessed: Feb 27, 2023]

[22] Elastic. (n.d.). Beats reference. Retrieved February 24, 2023, from https://www.elastic.co/guide/en/beats/libbeat/current/beats-reference.html

[23] M. Amer, "Logstash for Synchronize Elasticsearch with DBs," Medium, 28 June 2020. [Online]. Available: https://mohaamer5.medium.com/logstash-for-synchronize-elasticsearch-with-dbs-e5dda7cea930. [Accessed: 20 February 2023].

[24] Kibana, [Online]. Available: https://www.elastic.co/kibana/. [Accessed: Mar 1, 2023]

[25] Threat Intelligence Platform, [Online]. Available: https://en.wikipedia.org/wiki/Threat_Intelligence_Platform. [Accessed: Mar 4, 2023]

[26] Tim Wilson, “Threat Intelligence Platform: The Next Must-Have for Harried Security Operations Teams, Jun 2015. [Online]. Available: https://www.darkreading.com/attacks-breaches/threat-intelligence-platforms-the-next-must-have-for-harried-security-operations-teams. [Accessed: Mar 04, 2023].

[27] M. Senftleben and A. Willemsen, "Introduction to Information Sharing with MISP," MISP Project, [Online]. Available: https://www.misp-project.org/documentation. [Accessed: Mar 18, 2023].

[28] S. Zeb, M. Shahid, R. Ahmad, N. Javaid, "MISP vs. OpenCTI: A Comparative Study of Cyber Threat Intelligence Sharing and Analysis Platforms," Computers & Security, vol. 110, p. 102310, Jan. 2022.

[29] CIRCL (Computer Incident Response Center Luxembourg). (n.d.). MISP - Malware Information Sharing Platform & Threat Sharing. In MISP (Malware Information Sharing Platform) User Guide (p. 64), [Online]. Available: https://www.circl.lu/doc/misp/book.pdf. [Accessed: Mar 19, 2023].

[30] A. Al Jallad, I. Aloulou, and M. Samet, "A Comparative Study of OpenCTI and MISP for Cyber Threat Intelligence Sharing and Analysis," Journal of Cybersecurity and Privacy, vol. 1, no. 1, pp. 1-16, Dec. 2021.

[31] Filigran. (n.d.). Open CTI Architecture. [Online]. Available: https://filigran.notion.site/Architecture-5ce8241eac7e4e24906249e9595314cd [Accessed: Mar 20, 2023].

[32] ANSSI, “OpenCTI – The Open Source Solution for Processing and Sharing Threat intelligence Knowledge”, [Online]. Available: https://www.ssi.gouv.fr/actualite/opencti-the-open-source-solution-for-processing-and-sharing-threat-intelligence-knowledge/ [Accessed: Apr 7, 2023].

[33] MITRE ATT&CK. [Online]. Available: https://attack.mitre.org/ [Accessed: Apr 7, 2023].

[34] Structured Threat Information Expression (STIX). [Online]. Available: https://oasis-open.github.io/cti-documentation/ [Accessed: Apr 7, 2023].

[35] GraphQL, “A Query Language for your API”. [Online]. Available: https://graphql.org/ [Accessed: Apr 5, 2023].

[36] TheHive. [Online]. Available: https://github.com/TheHive-Project/TheHive [Accessed: Apr 5, 2023].

[37] MISP, “Threat Intelligence Sharing Platform”. [Online]. Available: https://github.com/MISP/MISP [Accessed: Apr 5, 2023].

[38] Alves, M. M. (2021). OpenCTI as Threat Intelligence Platform for Security Operation Centers. In Proceedings of the 16th International Conference on Evaluation of Novel Approaches to Software Engineering (pp. 679-686).

[39] Filigran. (n.d.). OpenCTI. Data visualization is a feature of OpenCTI, a cyber threat intelligence platform offered by Filigran. [Online]. Available: https://www.filigran.io/en/products/opencti/. [Accessed: Mar 19, 2023].

[40] CRITs. (n.d.). What Is CRITs? . [Online]. Available: https://github.com/crits/crits. [Accessed: Mar 21, 2023].

[41] Schrottner, J., et al. (2021). The Landscape of Cyber Threat Intelligence Platforms: An Evaluation Framework for Researchers and Practitioners. 10th International Conference on Decision Support System Technology, ICDSST 2021. [Online]. Available: https://diglib.uibk.ac.at/ulbtirolhs/download/pdf/6676638?originalFilename=true. [Accessed: Mar 19, 2023].

[42] CRITs. [Online]. Available: https://crits.github.io/ [Accessed: Apr 7, 2023].

[43] Cheng, F., Li, W., Li, W., Li, Y., Li, H., & Li, Y. (2021). A Risk Management System for Hotels Based on Threat Intelligence. Journal of Ambient Intelligence and Humanized Computing, 11(12), 5521-5532.

[44] Center for Internet Security. (2021). Collective Intelligence Framework (CIF). [Online]. Available: https://csirtgadgets.com/collective-intelligence-framework/ [Accessed: Mar 22, 2023].

[45] Greg Farnham, “Tools and Standards for Cyber Threat Intelligence Projects”, Jun 2012. Online]. Available: https://www.giac.org/paper/gcpm/134/tools-standards-cyber-threat-intelligence-projects/108367 [Accessed: Apr 10, 2023].

[46] Extensible Markup Language (XML). [Online]. Available: https://www.w3.org/XML/ [Accessed: Apr 7, 2023].

[47] JSON. [Online]. Available: https://www.json.org/json-en.html [Accessed: Apr 7, 2023].

[48] CSV – Common Separated Values. [Online]. Available: https://datahub.io/docs/data-packages/csv [Accessed: Apr 7, 2023].

[49] Melo e Silva, A. de, Gondim, J.J.C., Oliveira Albuquerque, R. de, Villalba, L.J.G.: A Methodology to Evaluate Standards and Platforms within Cyber Threat Intelligence. Future Internet, 1-23 (2020)

[50] Tounsi, W., Rais, H.: A survey on technical threat intelligence in the age of sophisticated cyber attacks. In: Computers & Security 72, 212-233 (2018)

[51] Staiger, T. (2021). Cyber Threat Intelligence Sharing Platforms: A Comprehensive Analysis of Software Vendors and Research Perspectives. Master's thesis, University of Innsbruck.

[52] Lodi, M. (2023). IntelOwl. [Online]. Available: https://intelowl.readthedocs.io/ [Accessed: Mar 20, 2023].

[53] Intel Owl. (2021, September 13). Intel Owl Release v3.0.0. The Honeynet Project. [Online]. Available: https://www.honeynet.org/2021/09/13/intel-owl-release-v3-0-0/. [Accessed: Mar 21, 2023]

[54] F. Sabahi and A. Movaghar, "Intrusion Detection: A Survey," *2008 Third International Conference on Systems and Networks Communications*, Sliema, Malta, 2008, pp. 23-26, doi: 10.1109/ICSNC.2008.44.

[55] Palo Alto Networks, "What is an Intrusion Detection System?", Mar 2022. [Online]. Available: https://www.paloaltonetworks.com/cyberpedia/what-is-an-intrusion-detection-system-ids [Accessed: May 15, 2023].

[56] Waleed, A., Jamali, A. F., & Masood, A. (2022). Which open-source IDS? Snort, Suricata or Zeek. Computer Networks, 213, 109116.

[57] The Snort Project. (2020, April 8). Snort Users Manual (Version 2.9.16). [Online]. Available: https://snortorgite.s3.amazonaws.com/production/document_files/files/000/000/249/original/snort_manual.pdf. [Accessed: Mar 29, 2023].

[58] Gavrilović, N., Ćirić, V., & Lozo, N. (2022). Snort IDS System Visualization Interface for Alert Analysis. Serbian Journal of Electrical Engineering, 19(1), 67-78. DOI: https://doi.org/10.2298/SJEE2201067G.

[59] Shuai, L., & Li, S. (2021). Performance optimization of Snort based on DPDK and Hyperscan. Procedia Computer Science, 183, 837-843. doi: 10.1016/j.procs.2021.02.172

[60] Abdulrezzak, S., & Sabir, F. A. (2023). Enhancing Intrusion Prevention in Snort System. In Proceedings of the 2023 15th International Conference on Developments in eSystems Engineering (DeSE) (pp. 1-5). IEEE. DOI: 10.1109/DESE58274.2023.10099757

[61] OISF, "Suricata User Guide Release 5.0.3." [Online]. Available: https://suricata.readthedocs.io/_/downloads/en/suricata-5.0.3/pdf/. [Accessed: April 7, 2023].

[62] I. Ghafir, V. Prenosil, J. Svoboda, & M. Hammoudeh, "A survey on network security monitoring systems," 2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW).

[63] Fekolkin, R. (2015). Intrusion Detection and Prevention Systems: Overview of Snort and Suricata. Internet Security, A7011N. Luleå University of Technology.

[64] Alves, A. F. P. (2020). Integrating an Intrusion Detection System with Heterogeneous IoT Endpoint Devices (Thesis). Universidade do Minho, Escola de Engenharia. Retrieved from. DOI: 10.13140/RG.2.2.21711.92328.

[65] Fadhilah, D., & Marzuki, M. I. (2020). Performance Analysis of IDS Snort and IDS Suricata with Many-Core Processor in Virtual Machines Against DoS/DDoS Attacks. In Authorized licensed use limited to: National Science & Technology Development Agency. In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 78-1-7281-7450).

[66] Hua, Q., Yu, S.-Y., & Asghar, M. R. (2020). Analysing performance issues of open-source intrusion detection systems in high-speed networks. Journal of Information Security and Applications, 51, 102426. DOI: 10.1016/j.jisa.2019.102426

[67] Zeek, 2022. “The zeek network security monitor”. [Online]. Available: https://zeek.org/. [Accessed: April 25, 2023].

[68] The Cyber Center for Security and Analytics. (2020). Zeek Intrusion Detection Series: Lab 1 - Introduction to the Capabilities of Zeek. Document Version: 03-13-2020.

[69] Muhammad, A.R., Sukarno, P., Wardana, A.A. (2023). Integrated Security Information and Event Management (SIEM) with Intrusion Detection System (IDS) for Live Analysis based on Machine Learning. Procedia Computer Science, 217, 1406-1415.

[70] Elvira Nassirova, “Data Visualization Dashboard: Benefits, Types, and Examples”, Jan 2023. [Online]. Available: https://blog.coupler.io/what-is-data-visualization-dashboard/ [Accessed: May 18, 2023].

[71] Sameer Bhale, “Dashboard Data Visualization: Benefits, Types, and Examples”, Feb 2023. [Online]. Available: https://www.knowledgehut.com/blog/business-intelligence-and-visualization/dashboard-data-visualization [Accessed: May 18, 2023].

[72] Apache Superset. (2021). Introduction to Apache Superset - Apache Superset documentation. [Online]. Available: https://superset.apache.org/docs/intro/. [Accessed: Mar 21, 2023].

[73] Grafana. (n.d.). Dashboards. Retrieved March 29, 2023, [Online]. Available: https://grafana.com/docs/grafana/latest/dashboards/ [Accessed: Mar 21, 2023].

[74] Grafana documentation. (n.d.). Demo Dashboards. [Online]. Available: https://play.grafana.org/d/000000012/grafana-play-home?orgId=1. [Accessed: Mar 21, 2023].

[75] Leppänen, T. (2021). Implementation of Grafana monitoring tool for containerized microservices (Bachelor's thesis, Turku University of Applied Sciences). [Online]. Available: https://www.theseus.fi/bitstream/handle/10024/512860/Turkuamk_Bachelors_Thesis_Leppanen_Tiia.pdf?sequence=2&isAllowed=y . [Accessed: Mar 21, 2023].

[76] Metabase. (n.d.). Tour of Metabase. Retrieved from https://www.metabase.com/learn/getting-started/tour-of-metabase#how-to-query-and-visualize-your-data

[77] Elasticsearch BV. (2021). Kibana: Introduction. [Online]. Available: https://www.elastic.co/guide/en/kibana/current/introduction.html. [Accessed: Mar 24, 2023].

[78] OpenSearch contributors. (2023). Introduction to OpenSearch [Documentation]. Retrieved from https://opensearch.org/docs/latest/. [Accessed: Jun 1, 2023].

[79] Opster Expert Team - Gustavo. "OpenSearch Observability Visualizations: How to Use Notebooks and Operational Panels." [Online]. Available: https://opster.com/guides/opensearch/opensearch-basics/opensearch-dashboards-visualizations-notebooks-operational/. Updated: Oct 30, 2022. [Accessed: June 1, 2023].

[80] Chatzichrysou, D. (2019). Evaluate ElastAlert for IT-DB use cases. CERN IT Department – Database Group – Infrastructure & Automation. Supervisors: Tsouvelekakis, A., & Coterillo Coz, I.

[81] Long, Q. (2019). ElastAlert Documentation Release 0.0.1. [Online]. Available: https://elastalert.readthedocs.io/_/downloads/en/latest/pdf/.[Accessed: May 7, 2023].

[82] Pivotto, J. (2019, November 8). Improved alerting with Prometheus and Alertmanager [Conference presentation]. PromCon Munich. [Online]. Available: https://promcon.io/2019-munich/slides/improved-alerting-with-prometheus-and-alertmanager.pdf. [Accessed: May 1, 2023].

[83] Dubey, A. (2018, March 25). AlertManager Integration with Prometheus. [Online]. Available: https://iamabhishek-dubey.medium.com/alertmanager-integration-in-prometheus-197e03bfabdf. [Accessed: May 2, 2023].

[84] Shinken Documentation. (Release 2.4). Shinken Team. August 14, 2015. [Online]. Available: https://readthedocs.org/projects/shinken/downloads/pdf/latest/. [Accessed: May 7, 2023].

[85] NocoDB Documentation. (0.108.1: Bug Fix Release) [Online]. Available: https://docs.nocodb.com/. [Accessed: June 1, 2023].

[86] Menor, D. (May 30, 2022). Best Open-Source Airtable Alternatives. [Online]. Available: https://hashdork.com/best-open-source-airtable-alternatives/.[Accessed: June 1, 2023].