A

MAJOR PROJECT REPORT

ON

REAL - TIME MONITORING AND ALERTING USING SPLUNK

Submitted in partial fulfilment of the Academic requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

"CSE- CYBER SECURITY"

By

GANDLA RAJU (21D41A6226)

Under the esteemed guidance of

Mrs. N. MADHU BHAVANI

(Assistant Professor)

DEPARMENT OF CSE-CYBER SECURITY

SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY

(An Autonomous Institute)

(Approved by AICTE, New Delhi & Permanently Affiliated to JNTU Hyderabad)

Sheriguda(V), Ibrahimpatnam(M), R.R. Dist-501 510

2024-25

SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY

(An Autonomous Institute)

(Approved by AICTE, New Delhi & Permanently Affiliated to JNTU Hyderabad)

DEPARMENT OF CSE-CYBER SECURITY

CERTIFICATE

This is to certify that the Major Project entitled "**REAL - TIME MONITORING AND ALERTING USING SPLUNK**" is submitted in partial fulfilment for the award of **Degree of Bachelors of Technology** in **CSE-Cyber Security** of Sri Indu College of Engineering and Technology, Hyderabad for the academic year **2024-2025**. This is a record of the project work done by **Gandla Raju (21D41A6226)** which has been approved as it satisfies academic requirements in respect of the work prescribed for **IV YEAR, II-SEMESTER of B. TECH** course.

INTERNAL GUIDE

Asst. Prof N. Madhu Bhavani

HEAD OF THE DEPARTMENT

Prof. Uma Maheswari G

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

With great pleasure I want to take this opportunity to express our heartfelt gratitude to all the people who helped in making this project work a success. I thank the almighty for giving me the courage & perseverance in completing the project.

I am very thankful to Principal **Prof. Dr. G. SURESH sir**, for giving me the permission to carry out this project and for providing necessary infrastructure and labs.

I'm highly indebted to **Prof G. UMA MAHESWARI madam**, Head of the Department of CSE-Cyber Security for providing valuable guidance at every stage of this project.

I am grateful to our internal project guide **Asst. Prof N. Madhu Bhavani madam,** for her constant motivation and guidance given by her during the execution of this project work.

I would like to thank the Teaching & Non-Teaching staff of Department of CSE-Cyber Security for sharing their knowledge with me.

Last but not the least I express my sincere thanks to everyone who helped directly or indirectly for the completion of this project.

Gandla Raju

(21D41A6226)

DECLARATION

I declare the project work **"REAL - TIME MONITORING AND ALERTING USING SPLUNK"** was carried out by me and this work is not the same as that of any other and has not been submitted to anywhere else for the award of any other degree.

Signature of the candidate:

1. …….

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES **PAGE NO**

LIST OF TABLES

1. ABSTRACT

Real-time monitoring and alerting are crucial for maintaining the health and performance of modern IT systems. This abstract describes a system leveraging Splunk to achieve this. Splunk's ability to ingest and analyze diverse machine data – from application logs and system metrics to network traffic and security events – makes it ideal for comprehensive monitoring. This system utilizes Splunk's search processing language (SPL) to define complex alert conditions based on thresholds, trends, and correlations across multiple data sources. These actions can range from notifications via email, SMS, or integrations with collaboration platforms, to automated remediation steps like restarting services or scaling resources. This includes techniques for alert prioritization to focus on critical issues, correlation to reduce alert noise from related events, and suppression of redundant alerts. By implementing these strategies, the system empowers operations teams to proactively identify, diagnose, and resolve problems, minimizing downtime and ensuring optimal service delivery. Finally, the abstract may briefly touch upon the scalability and extensibility of the Splunk-based system, highlighting its ability to adapt to evolving monitoring needs and integrate with existing IT infrastructure.

2. INTRODUCTION

In this chapter it is described the work done in the context of the experimentation on Splunk tool in analysing Logs. The work starts from the initial point, the search for logs, and continues with the problems to face for manipulation of raw data, the control rules for detecting web attacks, ending with a Machine Learning and statistical approach to the determination of anomalies.

Description of the starting point

The starting point of my trip is a collection of files, the logs, within a medium-high complexity application infrastructure. My aim is to make these objects becoming a resource, more properly an asset of the organization.

The logs inside the Application Infrastructure (for the description of the architecture see the Figure 27) are many and often scattered in the meanders of the machines and their directories, and are often not known at company level except by individual users.

They are often used by individual developers who use them for troubleshooting purposes, or by system administrators who use them for management purposes, or by security officers for control purposes.

However, there is a lack of a unified vision of the resource, a lack of cross analysis between the various logs and of an in-depth look at the information they contain.

It often happens that these files, sometimes very voluminous, are not even read, before being forgotten or completely deleted. For this reason, it has origin the present work with the need to try to enhance, through an approach of “exploration and discovery”, of what is hidden in those logs. Then my activity consisted in what can be called “log mining”.

As being perfectly aware that logs, intended as a support to contain a record of the activities carried out in an information system, can come from very different sources (firewalls, servers, databases, networks, applications, etc..) and that my main purpose is to explore only the contents of logs of application origin.

Fragmentary knowledge of available logs

Often every single developer knows the location of their own logs. You have to look for a

little bit before you find what you are looking for, if, as in my case, you play a cross-cutting role in applications. In an Infrastructure like the one under consideration (which is based on the J2EE platform and which is built with a SOA approach, using IBM Websphere Application Server), it is well known in advance the position and the format of the Application Server logs, since they are standardized.

As far as the logs of the single applications of which the infrastructure is composed are concerned, it could be searched in the directories of the respective applications, where the logs are produced.

These logs are the ones on which the attention is focused since they are of direct competence of a software development unit in which this work was carried out.

The evaluation of the format of the individual logs made in the first analysis let me believe my work could pursue several objectives:

(1) to proceed to an automatic extraction of targeted information
(2) measure quantities, evaluate trends
(3) detecting errors, reporting
(4) exploring relations between information

It should be decided to adopt a specific or general solution? Make or buy? Open source or commercial?

A possible choice was to build a log analysis [1] solution starting from opensource resources, freely available, and of limited cost.

In particular, the “grep” commands could be used. This command is a Linux or Unix utility that searches file input patterns or regular expressions.

This type of command could allow us to focus on a single problem at a time and examine the results in detail.

How ever it seemed to be more targeted for small cases or for analysis that are carried out with a high degree of manual skills and also have a low “friendly” type degree of usability. It also seemed that the integration into an infrastructure was restrictive and cumbersome, and complex, above all in the case of multiple types of different logs.

In addition, there are the following factors that are detrimental to this “grep utility” tool:

- It is suitable when you know perfectly what you want to achieve, and you focus on very specific text, on the whole file.
- You should not overlook the time it takes to search for large files.
- It seems often endless as you search all over the file (which can be terabytes!).
- It is not easy for correlating searches on multiple log files at the same time, or even on different devices (it may be useful to extend the search on the Web server of the Web Farm, or on the HTTP server, or on a Firewall).

Splunk [2] commercial tool immediately seemed to me the most viable solution, providing this company the Enterprise distributions, freely downloadable, with licenses valid for 60 days, or, alternatively, Developer licenses, valid for 6 months.

This tool offers a series of features that reduce the time needed to analyze the logs on a periodic basis, as there are activatable files/directories monitors, which update the analysis. It also makes available the generation of alerts at the occurrence of predefined events, the generation of reports that allow corporate compliance, the centralization of the logs to be analyzed on which to make further actions or investigations. It should not be underestimated that several add-ons do exist, and also apps and APIs that allow Splunk to be integrated with the surrounding infrastructure.

Its use allows the easy centralization of log management, through different instances of log collector (Forwarders), or log analyzer (Head Searches). It is possible to analyze multiple sources simultaneously, without affecting performance, as Splunk is based on the concept of data indexing (Indexer) that optimizes search times.

The latter is an important aspect in log management because, in a multi-application and multi-distributed infrastructure, the application log is produced on the machine on which the application is hosted and executed, and in the case of cross analysis of logs from different sources, it is necessary to centralize it as the next Figure 1:

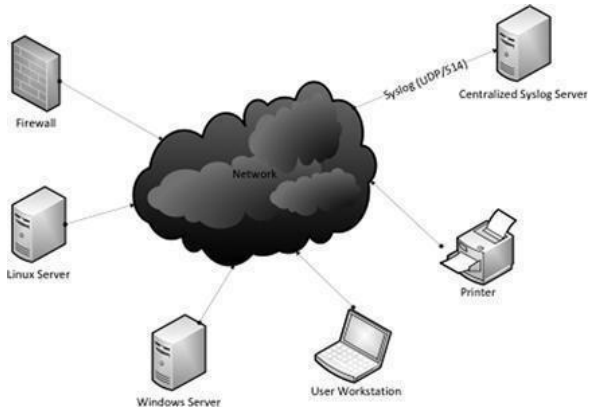


Figure 1: Centralization of Logs.

It is also easy to report and display in a Dashboard the summary data of the search results, thanks to a rich availability of graphs, which can be used from a web interface or even via a command line or web services.

2.1. Motivation

The driving force behind implementing real-time monitoring and alerting with Splunk stems from the critical need for proactive, data-driven decision-making in today's dynamic digital landscape. Here's a breakdown of the key motivations:

- **Proactive Issue Detection and Resolution:**
    - Real-time monitoring allows organizations to identify and address issues before they escalate into major disruptions. By analyzing data as it's generated, potential problems like server outages, security breaches, or application errors can be detected early.
    - This proactive approach minimizes downtime, reduces operational costs, and enhances overall system reliability.
- **Enhanced Security Posture:**
    - Real-time security monitoring with Splunk enables organizations to detect and respond to security threats in near-real-time.
    - By correlating security logs and events, suspicious activity can be identified, and immediate action can be taken to mitigate potential damage.
    - This includes detecting intrusion attempts, anomalous user behavior, and data exfiltration.
- **Improved Operational Efficiency:**

    - Real-time monitoring provides valuable insights into system performance, application behavior, and user activity.
    - This data can be used to optimize resource allocation, identify bottlenecks, and improve overall operational efficiency.
    - Automation of alerts, also allows for rapid responses to known issues, allowing for less human interaction, and more efficiency.
- **Data-Driven Decision-Making:**
    - Real-time dashboards and reports provide a comprehensive view of critical metrics and KPIs.
    - This information empowers organizations to make informed decisions based on accurate and up-to-date data.
    - For example, during peak usage times, resources can be dynamically allocated.
- **Compliance and Auditing:**
    - Real-time monitoring helps organizations maintain compliance with regulatory requirements and industry standards.
    - Splunk's audit trail capabilities provide a detailed record of system activity, which is essential for compliance reporting and auditing.
- **Enhanced User Experience:**
    - By monitoring application performance and user behavior in real-time, organizations can identify and address issues that may impact the user experience.
    - This leads to improved customer satisfaction and increased user engagement.
- **Reduced Mean Time To Resolution (MTTR):**
    - The speed of real time alerting, allows for a much quicker response to any detected anomolies. This quick response time, reduces the MTTR, and keeps systems running.
- **Predictive Analysis:**
    - Splunk allows for the analysis of trends, and the ability to predict future issues. This allows for preventative maintanence, and the prevention of future outages.

2.2. Existing System

Before implementing real-time monitoring and alerting with Splunk, organizations often rely on a variety of existing systems and practices, which may exhibit limitations and inefficiencies. Here's a general overview of common pre-Splunk scenarios:

- **Log Files and Manual Analysis:**
    - Many systems rely on traditional log files for troubleshooting and performance monitoring.
    - Analyzing these logs often involves manual searching, grepping, and parsing, which is time-consuming, error-prone, and inefficient, especially in large-scale environments.
    - This manual process inhibits the ability to detect issues in real time.
- **Disparate Monitoring Tools:**
    - Organizations may use a collection of separate monitoring tools for different systems and applications.
    - This can result in fragmented data, inconsistent reporting, and a lack of a unified view of the environment.
    - Correlation of events across different systems becomes difficult, hindering root cause analysis.
- **Reactive Troubleshooting:**
    - Troubleshooting is often reactive, meaning issues are addressed only after they have caused a disruption.
    - This can lead to significant downtime, data loss, and customer dissatisfaction.
    - The lack of real-time insights prevents proactive issue prevention.
- **Limited Alerting Capabilities:**
    - Existing alerting systems may be limited in their capabilities, providing basic notifications based on simple thresholds.
    - They may lack the ability to correlate events, suppress noise, or provide detailed contextual information.
    - Alerts are often sent to a single location, and do not provide options for escalation.

- **Lack of Centralized Data:**
    - Data is often siloed in different systems, making it difficult to gain a holistic view of the organization's operations.
    - This lack of centralized data hinders the ability to identify trends, patterns, and anomalies.
- **Inadequate Security Monitoring:**
    - Security monitoring may be limited to basic log analysis and intrusion detection systems.
    - These systems may not be able to detect advanced threats or provide real-time correlation of security events.
    - Auditing and compliance reporting may be difficult to produce.
- **Spreadsheet Reporting:**
    - Many reports are generated manually, and placed into spreadsheets. This is a very time consuming, and error prone method of reporting.
- **Lack of Predictive Analysis:**
    - Most existing systems lack the ability to perform predictive analysis, which can help organizations anticipate and prevent future issues.

2.3. Literature Survey

Real-time monitoring and alerting have become essential components in modern IT operations, particularly for ensuring system availability, performance, and security. Splunk, known for its advanced log analysis and machine data processing capabilities, has emerged as a key tool in this domain.

1. **Real-Time Monitoring Capabilities**

   According to Patel et al. (2020), Splunk provides a scalable solution for monitoring complex IT infrastructures. Its ability to index, search, and visualize large volumes of data in real time makes it ideal for proactive system management. The authors emphasized the use of dashboards and scheduled reports to track key performance indicators (KPIs) and identify bottlenecks before they impact business services.

2. **Alerting and Anomaly Detection**

   Singh and Sharma (2019) evaluated Splunk’s alerting framework, highlighting how it

enables organizations to define thresholds and trigger automated responses to unusual activity. They demonstrated use cases in cloud environments where real-time alerts reduced downtime and accelerated incident resolution.

3. **Integration with Security Operations**

   Gupta et al. (2021) focused on Splunk's application in Security Information and Event Management (SIEM). Their research showed that Splunk can correlate events from multiple sources such as firewalls, intrusion detection systems, and antivirus tools, enabling real-time threat detection. They also discussed the integration of Splunk Enterprise Security (ES) to enhance incident response workflows.

4. **Machine Learning and Predictive Analytics**

   In a study by Kumar and Rao (2022), the use of the Splunk Machine Learning Toolkit (MLTK) was explored for anomaly detection and predictive alerting. Their experiments in monitoring server CPU loads and network traffic patterns demonstrated how predictive models could reduce false positives and improve the accuracy of alerts.

5. **Performance and Scalability**

   Verma et al. (2021) conducted a performance evaluation of Splunk in large-scale enterprise settings. They concluded that Splunk can efficiently handle terabytes of data daily while maintaining near real-time alerting capabilities. Their study also emphasized best practices such as data model acceleration and summary indexing for optimized performance.

6. **Case Studies and Industry Applications**

   Several real-world case studies, such as those published in Splunk's whitepapers (2020-2022), describe implementations across sectors like finance, healthcare, and e-commerce. For example, in the banking sector, Splunk has been used to monitor transaction logs to detect fraud in real time.

2.4. Challenges in Existing System

Organizations that rely on traditional or legacy monitoring systems for real-time monitoring and alerting face several limitations:

1. **Lack of Real-Time Data Processing**

   Many conventional systems process logs in batches rather than in real time. This delay

in data analysis can lead to missed critical alerts and slow incident response.

2. **Limited Scalability**

   Legacy tools often struggle to scale with the growing volume, variety, and velocity of machine data. As a result, performance deteriorates when monitoring large or distributed infrastructures.

3. **Poor Visualization and Reporting**Traditional monitoring lacks dynamic dashboards and visual analytics. This makes it difficult for administrators to gain insights quickly or share findings with non-technical stakeholders.

4. **Minimal Automation**

   Without intelligent alerting and automation, older systems rely heavily on manual log reviews and static threshold-based alerts, which are prone to errors and delays.

5. **Inadequate Correlation and Analysis**

   Legacy tools often can't correlate logs from diverse sources (e.g., servers, firewalls, databases). This leads to fragmented monitoring and incomplete root cause analysis.

6. **Higher Risk of Alert Fatigue**

   Due to the lack of machine learning or adaptive thresholds, traditional systems often generate a high volume of irrelevant alerts, overwhelming IT teams and delaying genuine issue resolution.

7. **Integration Issues**

   Non-modern systems typically have limited APIs or plugin support, making it hard to integrate them with modern DevOps tools, cloud platforms, or incident management systems.

8. **Security Blind Spots**

   Without centralized logging and intelligent analysis, potential security threats (like unusual login patterns or DDoS attempts) can go undetected until significant damage is done.

2.5. Proposed System

Solution Overview:

- Implement Splunk to aggregate data from multiple sources and provide a single-pane view of real-time metrics, logs, and events.

- Leverage Splunk's alerting and reporting features to automate monitoring and troubleshooting.

Key Benefits:

- **Real-time insights:** Instant visibility into system health and performance.
- **Automation:** Alerts and notifications reduce manual intervention.
- **Scalability:** Splunk can handle large volumes of data and scale as needed.
- **Enhanced security:** Proactive detection of suspicious activities with security event monitoring.

2.6. Objectives

The main objectives of this project, aimed at developing a real-time monitoring and alerting system using Splunk, are as follows:

1. **To enable real-time monitoring:**

   Continuously observe system logs, performance metrics, and event data from diverse sources to ensure timely detection of issues.

2. **To automate alerting mechanisms:**

   Set up intelligent alerts that notify IT personnel immediately when predefined conditions or anomalies occur.

3. **To centralize data collection:**

   Aggregate and index machine data from various platforms into a single, accessible repository for streamlined analysis.

4. **To enhance decision-making:**

   Provide actionable insights via interactive dashboards and real-time visualizations to support technical and managerial decision-making.

5. **To improve incident response:**

   Reduce system downtime and accelerate troubleshooting through early detection, automated alerts, and rapid root cause identification.

6. **To integrate predictive analytics:**

   Use machine learning models for detecting anomalies and forecasting future issues based on historical data trends.

7. **To strengthen security monitoring:**

   Monitor security-related logs and detect potential threats through correlation and behavior analysis.

8. **To ensure scalability and flexibility:**

   Develop a system architecture capable of handling increasing data volumes and adapting to evolving infrastructure needs.

2.7. Methodology

The methodology for developing the real-time monitoring and alerting system using Splunk follows a systematic, phased approach. Each phase ensures efficient data handling, timely alerting, and actionable insights.

1. **Requirement Analysis**

   This initial phase involves identifying the monitoring goals, log sources, alerting thresholds, and performance/security requirements. Inputs are gathered from stakeholders to define key performance indicators (KPIs), data types, and integration needs.

2. **Data Collection and Ingestion**

   a. Data is collected from various sources using:

   b. Splunk Universal Forwarders for server and application logs

   c. Syslog servers for network devices

   d. APIs and cloud connectors for cloud service logs

   e. Ingested data is routed to indexers for storage and analysis.

3. **Indexing and Parsing**

   Splunk indexes raw machine data, parses events, and extracts relevant fields. Key steps include:

   a. Field extraction using regular expressions or in-built field extractors

   b. Data normalization using the Common Information Model (CIM)

   c. Organizing data by source/type for optimized searching

4. **Search and Alert Configuration**

   Using Splunk's Search Processing Language (SPL), custom searches are built to:

   a. Detect anomalies and threshold breaches

b. Configure real-time and scheduled alerts

c. Trigger actions such as email notifications, script execution, or ticket generation

5. **Dashboard Development**

   Dashboards are created to visualize:

   a. System health and performance metrics

   b. Alert summaries and trends

   c. Real-time updates with user-friendly charts and drill-down capabilities

6. **Machine Learning Integration (Optional)**

   Where required, Splunk MLTK is used to:

   a. Apply unsupervised or supervised models

   b. Detect abnormal behaviors and forecast resource usage

   c. Reduce false positives in alerting through model-based thresholds

7. **Testing and Optimization**

   The system undergoes rigorous testing to validate:

   a. Alert accuracy and timeliness

   b. Data ingestion speed and query performance

   c. Dashboard responsiveness and resource efficiency

   d. Optimization techniques include summary indexing, search acceleration, and index time tuning.

8. **Deployment and Continuous Monitoring**

   The final system is deployed with:

   a. Continuous monitoring of Splunk's internal logs and performance

   b. Routine checks and updates to adapt to infrastructure or application changes

   c. Periodic reviews to refine dashboards and alerting logic

2.8. Hardware and Software Requirements

Hardware Requirements:

Components	Minimum Requirement	Recommended for Production
CPU	Quad-Core (4 Cores)	8-Core or Higher

RAM	8 GB	16-32 GB or more
Storage	100 GB (SSD preferred)	512 GB+ SSD with RAID configuration
Network	1 Mbps Ethernet/Wi-Fi	1 or 10 Gbps (for high ingestion rates)
Servers	Single server for testing	Distributed setup (Indexer, Search Head, Forwarders)

Software Requirements:

Category	Description
Operating System	Windows Server or Linux (Ubuntu/CentOS/RHEL)
Splunk Enterprise	Version 9.0 or Latest
Web Browser	Chrome, Firefox or Edge (for accessing UI)
Add-ons & Apps	Splunk Common Information Model (CIM), Splunk Machine Learning Toolkit (MLTK), Splunk Enterprise Security (optional)
Supporting Tools	Python 3.x (for ML models), APIs and CLI tools

2.9. Organization of the Project

This project report is organized into several chapters, each addressing a critical aspect of the development and implementation of a real-time monitoring and alerting system using Splunk:

- **Chapter 1: Introduction**

  Provides an overview of the project, including background, objectives, problem statement, scope, and significance. It also covers the literature survey, challenges in existing systems, proposed system, hardware/software requirements, methodology, and project structure.

- **Chapter 2: System Design**

  Describes the system architecture, data flow, component design, and module-level functionality. It outlines how the different parts of the system interact to achieve real-time monitoring and alerting.

- **Chapter 3: Implementation**

  Details the step-by-step implementation of the system, including Splunk installation,

data ingestion setup, alert creation, dashboard development, and integration of optional machine learning models.

- **Chapter 4: Testing and Evaluation**

  Presents the results of system testing, discusses performance metrics, evaluates the effectiveness of alerts, and validates the system against the defined objectives.

- **Chapter 5: Conclusion and Future Work**

  Summarizes the outcomes of the project, highlights the contributions, and suggests possible improvements or future enhancements such as advanced analytics, multi-cloud integration, or extended alert automation.

- **References**

  Lists all the sources cited throughout the report in APA format.

- **Appendices**

  Includes supplementary materials such as sample dashboards, SPL queries, system configuration details, and screenshots.

3. PROPOSED SYSTEM

3.1. System Architecture

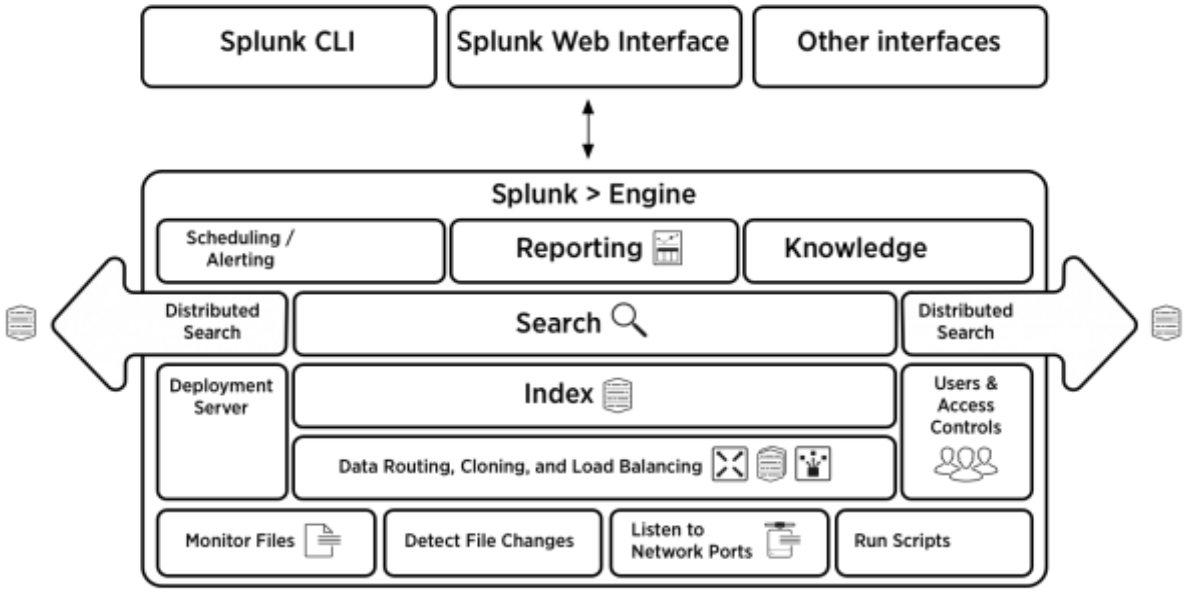


Figure 2: Splunk Architecture

3.2. UML Diagrams

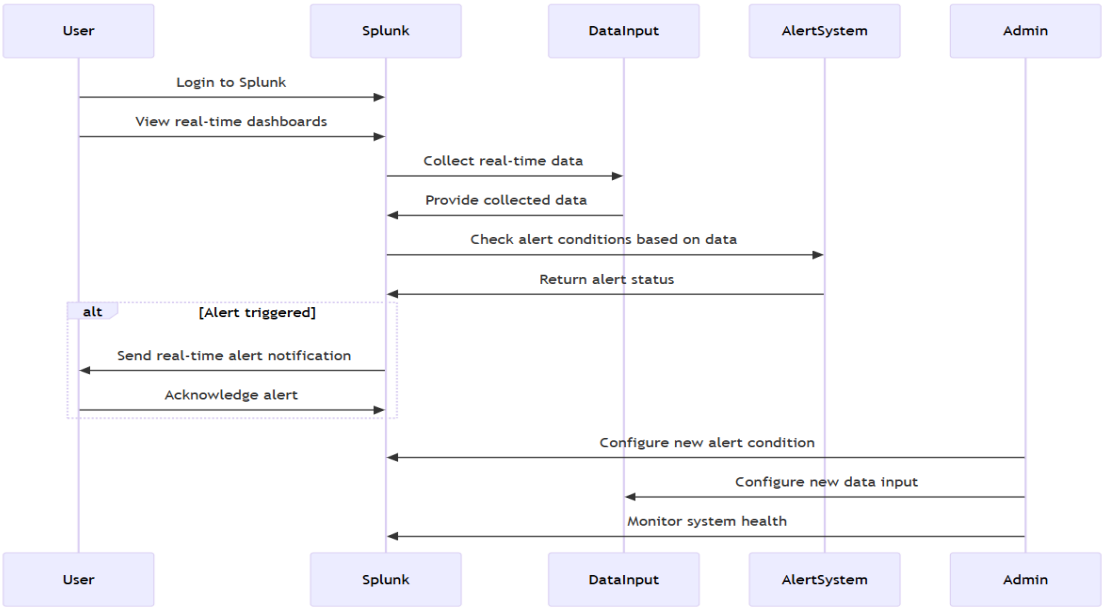


Figure 3: Sequence Diagram

3.3. Flow Chart

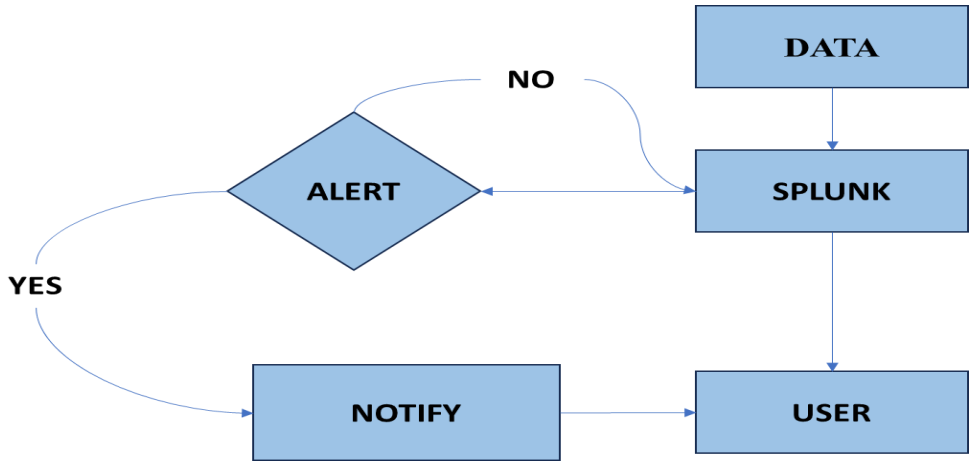


Figure 4: Flow Chart of Splunk

3.4. Description of Algorithms

Although Splunk primarily operates through configuration and query languages rather than traditional algorithms, certain logic-based processes and optional machine learning models are employed to enhance real-time monitoring and alerting. This section outlines the key algorithms and methods used.

1. 2.X.1 Search Processing Language (SPL) Logic

Splunk's core logic revolves around Search Processing Language (SPL), a domain-specific language used for data querying, filtering, correlation, and transformation. While not an algorithm in the conventional sense, SPL acts as the control logic to:

- **a.** Filter events using conditionals (e.g., status=500 for error logs)
- **b.** Aggregate metrics using statistical functions (stats, avg, count)
- **c.** Detect thresholds using conditional comparisons (if, eval)
- **d.** Join or correlate data using join or lookup

Example SPL query for error rate detection:

index=web_logs status=500 | stats count by host | where count > 10

2. Threshold-Based Alerting

This rule-based algorithm checks if real-time metrics exceed predefined static thresholds. When the condition is met, an alert is triggered.

Logic:

IF (metric_value > threshold) THEN trigger_alert()

3. Time Window Monitoring

Alerts can also be configured based on moving time windows. This logic uses rolling aggregates to detect patterns across time intervals.

Example:

Calculate average error rate over last 10 minutes.

IF (average_error_rate > expected_rate) THEN trigger_alert()

4. Anomaly Detection (Machine Learning Toolkit - Optional)

If enabled, Splunk's Machine Learning Toolkit (MLTK) supports supervised and unsupervised algorithms, such as:

a. K-Means Clustering: To identify outlier patterns in server metrics.

b. Density Function Estimation: To spot unusual deviations in log volumes.

c. Linear Regression: For trend forecasting of resource usage.

Example Pseudocode (Outlier Detection with Z-Score):

```
mean = avg(log_count)

std_dev = stddev(log_count)

z_score = (current_value - mean) / std_dev

if abs(z_score) > threshold:

   trigger_anomaly_alert()
```

5. Correlation Logic

Used primarily in security monitoring (via Splunk Enterprise Security), this logic correlates multiple events across different sources to identify complex incidents.

Example Logic:

IF (failed_login from multiple IPs) AND (suspicious file execution) THEN trigger_security_alert()

4. SPLUNK FUNCTIONAL ARCHITECTURE

4.1. Splunk Functional Architecture

It is possible to install the version of Splunk Enterprise on a single instance of a machine, in such a way to have on a single site all the peculiarities of Splunk, namely “data input”, “indexing”, “search”. In the case of more complex organizations, the typical architecture of Splunk is the type shown in the following Figure 2. The architecture is functional to the organization in which you want to install Splunk.

4.1.1. The Forwarders

Forwarders are 20 mb software distributions which can be installed directly on the site where the data need to be collected. They only monitor files or directories and send the collected data to Splunk instances (Enterprise type) that provide indexing of data.

4.1.2. The Indexers

While the number of Forwarders is equivalent to the number of machines on which there are logs that need to be analyzed, the number of Indexers is calculated on the basis of a capacity plan depending on the size of the logs that are estimated to be handled. It is necessary to have approximately 100GB for each Indexer, in a configuration that meets the hardware requirements provided by Splunk. On the Indexers takes place data parsing and data indexing.

4.1.3. The Search Heads

As far as concerns the Search Heads, which are the input end points for user interactions, their number depends on the number of Users expected for that configuration. It should be noted that a Search Head User means both an interactive user and REST application or also a web service user. Splunk search heads serve as the user's gateway to data analysis, acting as the centralized hub for initiating searches and visualizing results. When a user submits a query through the Splunk Web interface, the search head takes on the task of distributing that search across the indexers, which hold the actual data. In distributed deployments, this coordination is critical, as the search head effectively orchestrates the retrieval and merging of data from multiple indexers into a unified output. Beyond simple search execution, search heads also manage "knowledge objects," encompassing saved searches, reports, and dashboards that empower users to leverage Splunk's capabilities fully. To enhance performance and ensure

high availability, Splunk allows for search head clustering, where multiple search heads work in unison, sharing configurations and distributing workloads, thereby providing redundancy and uninterrupted access. Essentially, the search head handles the intellectual aspect of data analysis, leaving the heavy lifting of data storage and retrieval to the indexers, making it the primary point of interaction for those seeking insights from their machine data.

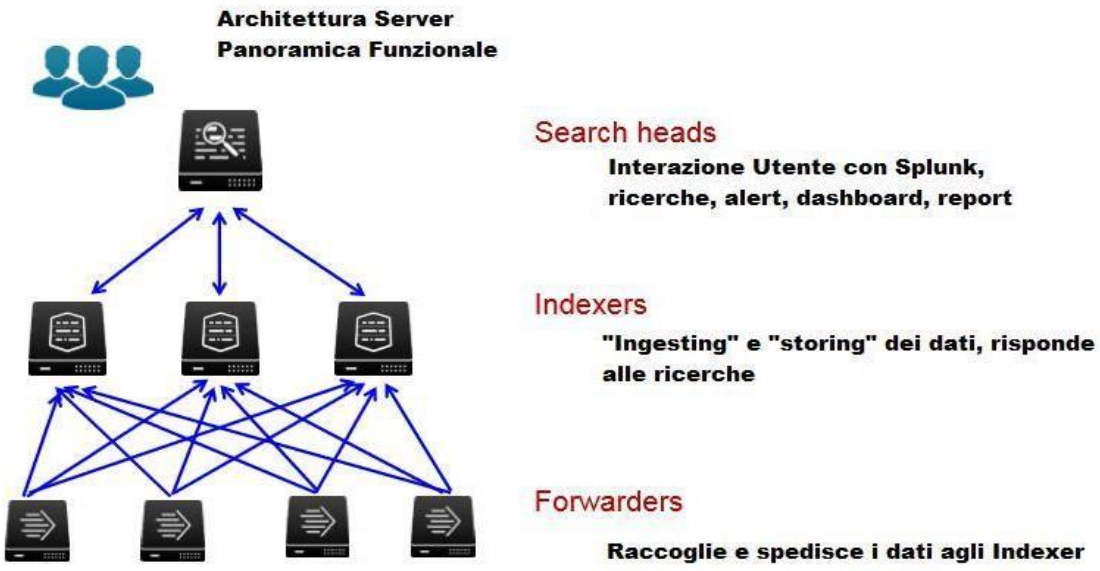


Figure 5: Splunk Functional Architecture Overview.

4.2. Initial Architecture

This architecture has been realized using Virtual Machines created on the Aruba Cloud VPS. The instances were created by installing Splunk and transferring the logs which were withdrawed from the production environment of the Infrastructure. The instances of Splunk which were created are reported below:

Universal Forwarderh: 1 of GB RAM, 1 single core CPU, 20 GB SDD, Linux system

Indexer: 2 GB RAM, 1 CPU single core, 40 GB SDD, Linux system

Search Head: 2GB RAM, 1 single core CPU, 40GB SDD, Linux system

Forwarder machines communicate with Indexer machines via a TCP connection on port 9997, through which they send data to the Indexer.

This minimum configuration is adequate to guarantee acceptable performance with a maximum of 5-10 logs, of about 300 mb size each and 1-2 million events. The architecture implemented in this project, is described in the following Figure 6.

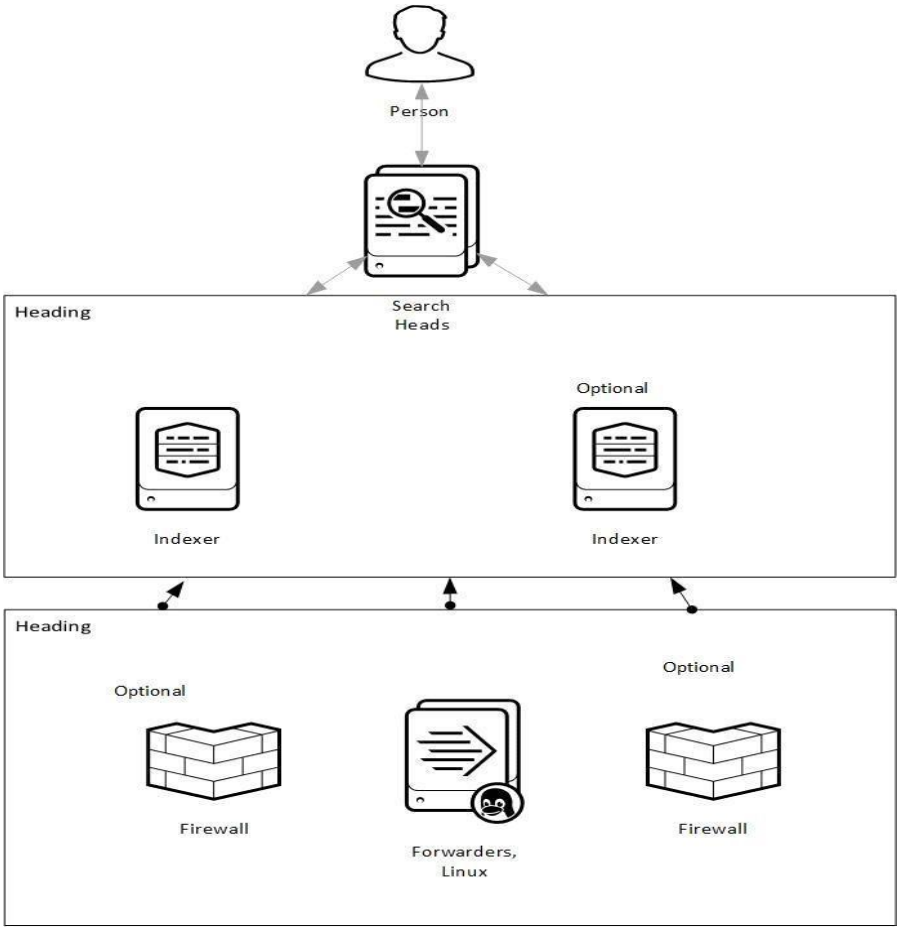


Figure 6: Initial Architecture

4.3. HW System Requirements

The hardware requirements reported in Splunk Documentation necessary to be an adequate configuration to ensure efficient log management and searching are listed here below:

Basic Level – Single Instance Host

Intel x86 64-bit chip architecture 12 CPU cores a 2Ghz (at least 12GB RAM Standard 1Gb Ethernet NIC 800 IOPS (on average) Performance of Indexing Max: 20MB/sec Performance of Search Max: 50.000 events/sec

Medium Level – Distributed Multi Hosts

Instance Intel x86 64-bit chip architecture 12-16 CPU cores 2Ghz (at least) 12GB RAM Standard 1Gb Ethernet NIC 800 IOPS (on average)

High Level – Distributed Multi Hosts

Instance Intel 64-bit chip architecture 48 CPU cores at 2GHz (at least) 128GB RAM SSD 1Gb Ethernet NIC plus NIC optional SAN

4.4. Splunk processing: data pipeline

The initial architecture distinguishes the functional activities of Splunk, which take place as a data processing chain. Regardless of whether a single-instance or multi-instance architecture is adopted, the "data pipeline" follows three distinct stages:

1. Input: where the data is preloaded, divided into 64K blocks and annotated with the metadata: host, source, sourcetype. At this stage, the concept of event does not yet exist.
2. Parsing: In this phase Splunk segments the data into events, sets the timestamps, notes the metadata it obtains from the previous input phase, possibly transforms the event data, according to the transformation rules (regex).
3. Indexing: the events obtained by parsing are written to the disk in compressed and indexed form.
4. Search: this phase concerns the user's access to the indexed data (in the form of search, reports, alerts, dashboards).

In the following Figure 7 is showed the data pipeline process:

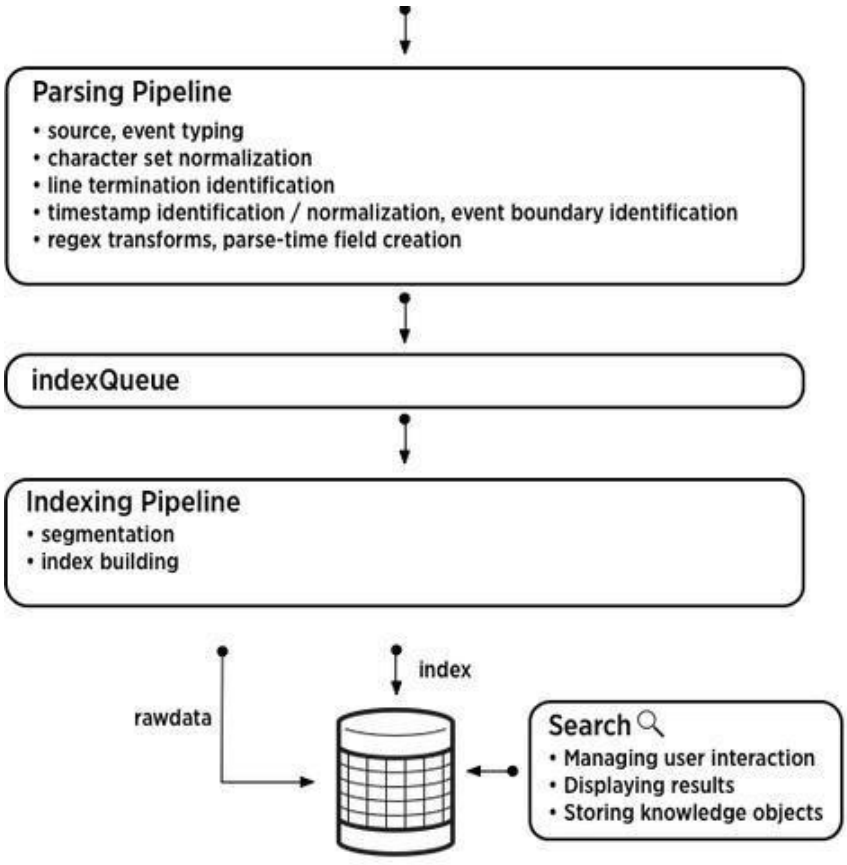


Figure 7: Data Pipeline.

Although in the present project there is a single source of data, it can be the most varied as Splunk is predisposed to all sources and formats (Figure 8).

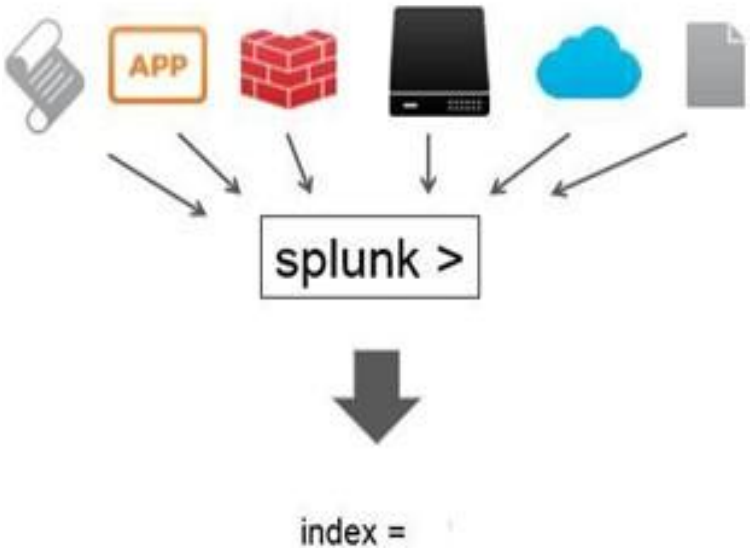


Figure 8: Heterogeneity of Data Sources.

5. IMPLEMENTATION & TESTING

5.1. Analysis of the Infrastructure Logs

In this project were analyzed sample portions of the Infrastructure Application Logs. In particular those related to a couple of applications.

Three main Indexes (for the sake of simplicity we will denominate them index A, index B and Index C) have been created, on which all the searches have been carried out.

The set of search queries submitted to the system are still general and can be repeated in multiple contexts. In order to increase the generalization, the ease of use and the reusability of the search queries, the corresponding macros have been stored (although not reported in this paper).

After loading, parsing, and indexing the data, it was necessary to pre-treat the data, in order to extract and better define the fields that are used in the subsequent stages of search.

5.2. Data Loading and Extraction

Data Loading into Splunk is an important step for the subsequent analysis and visualization phases. If the data are not correctly analyzed, time-stamped, and divided into events, it will be difficult in the next stages to carry out a 'proper analysis of the data and get insights into them'.

Since the data may come from many sources as web servers, application servers, operating systems, network security, services or applications (although in our case they will be exclusively of application origin) it is still important that the format makes all the subsequent analytical phases easily searchable.

In general, Splunk is equipped with internal facilities that recognize the most common formats from various sources (Figure 6) and then apply the preconfigured settings already stored. If Splunk recognizes the data source, it will apply the settings and definitions for that data source. The data formats can be divided into three types: **structured**, **semi-structured**, and **unstructured ones**.

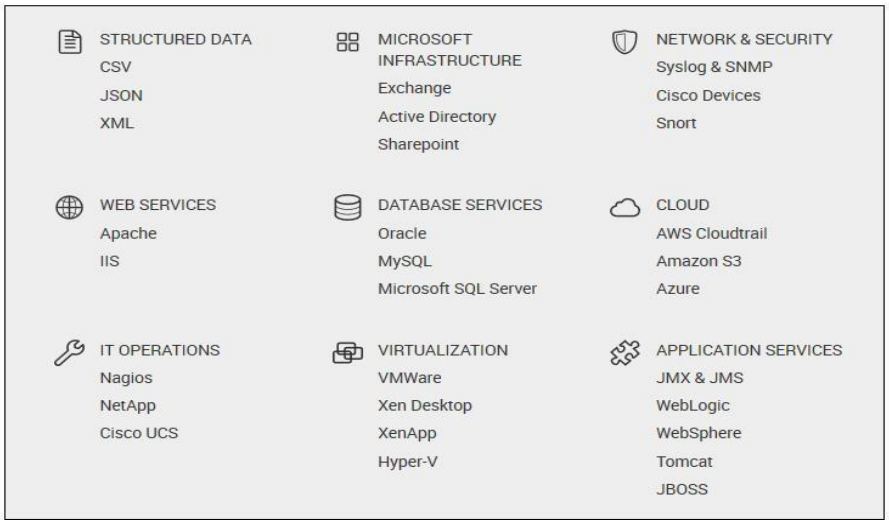


Figure 9: Preconfigured Splunk Data Formats.

If structured formats occur maybe additional settings need to be specified for the custom format in question. This can be done in the field extraction phase.

5.2.1. Application Logging

In the case of the application infrastructure under consideration, the logs considered are basically related to applications developed in-house. In this case, logging is delegated to developers for troubleshooting reasons, or for determining a certain level of quality service or determine system availability (through measurements of systems interruptions).

Most of the time they are just the developers who understand the meaning of the application logs as they are ultimately the only users to determine software bugs or make a fixing in case of malfunctions.

In the kind of infrastructure under consideration, which is a medium-high complexity system, with about 20 machines cooperating in the architecture, the resolution of application problems is a complex and time- consuming activity.

The adoption of a tool like Splunk, thanks to the use of queries and reports, alerts and messaging, can be of fundamental help to maintenance and operational activities.

This requires an activity of extraction of the fields, in order to fully analyze data, and thus making data to become an asset for the entire organization.

In fact, because it makes sense to believe that ’garbage in ... garbage out’, it cannot be

expected to extract value if into Splunk is introduced only information of little value. In this sense we should follow as much as possible an approach in which there is a minimum gap between the development and operation of the system. Basically you have to try to get a log that is ’machine friendly’ and ’human readable’.

5.2.2. The Loggers

At the application level, the developer uses loggers (i.e. java log4j) that allow to write messages in multithreaded mode and in a typical standard format. In this way, if there is an exception, some peculiar aspects can be checked, while others are established and delegated to the Java framework.

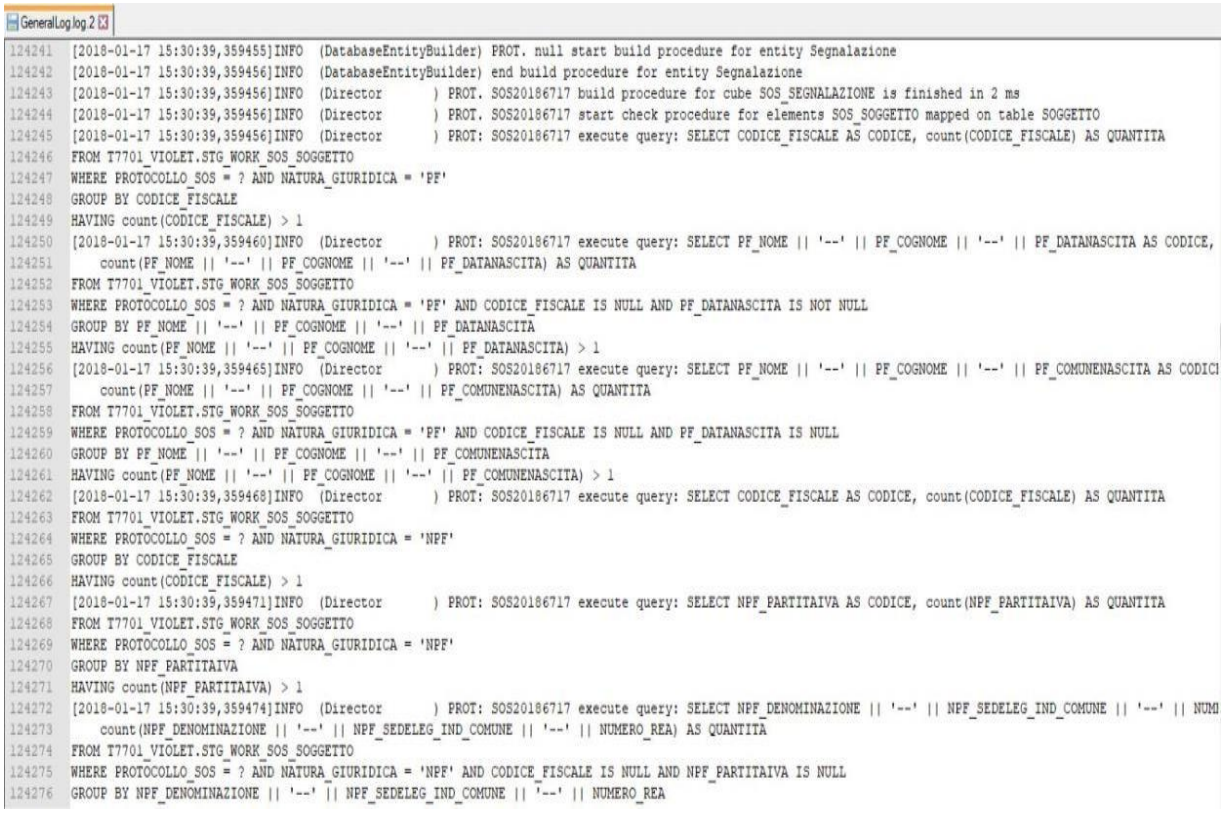
In other words, the logger keeps track of what happened in the program, so if a crash or bug occurs, you can trace the cause.

We can say that the events produced by the application loggers are in the form of ’**structured data’** while those that come from the proprietary applications developed ad hoc generally have a form of **’semi- structured’** data and in other cases of **’unstructured data’**. The latter may vary, and may also extend over several lines, while the event they report may be in a format without precise separators, and without the obligation of a start or an end clearly identified or reported explicitly.

The logs here reported and examined, relevant to the Infrastructure, are mostly structured data, although in some sections of the logs there may appear unstructured ones, as shown in Figure 10:

This latter type of log is much more complicated because all knowledge must be extracted manually and it requires collaboration and information exchange between the Splunk engineer and the programmer or vendor. In such cases, if Splunk is allowed to operate automatically, it will try to extract stuff that it wrongly believes to be data, and the whole process will not bring added value in the following phases of analysis and reporting.

To make sure that we are loading or extracting correctly, we must put order in the initial chaos and this usually begins by separating an event from the next one, by means of a line or a character. There are many ways in which you can separate events from each other (see Splunk/Admin/Propsconf Documentation).



```
GeneralLog.log.2
124241 [2018-01-17 15:30:39,359455]INFO  (DatabaseEntityBuilder) PROT. null start build procedure for entity Segnalazione
124242 [2018-01-17 15:30:39,359456]INFO  (DatabaseEntityBuilder) end build procedure for entity Segnalazione
124243 [2018-01-17 15:30:39,359456]INFO  (Director       ) PROT. SOS20186717 build procedure for cube SOS_SEGNALAZIONE is finished in 2 ms
124244 [2018-01-17 15:30:39,359456]INFO  (Director       ) PROT. SOS20186717 start check procedure for elements SOS_SOGGETTO mapped on table SOGGETTO
124245 [2018-01-17 15:30:39,359456]INFO  (Director       ) PROT: SOS20186717 execute query: SELECT CODICE_FISCALE AS CODICE, count(CODICE_FISCALE) AS QUANTITA
124246 FROM T7701_VIOLET.STG_WORK_SOS_SOGGETTO
124247 WHERE PROTOCOLLO_SOS = ? AND NATURA_GIURIDICA = 'PF'
124248 GROUP BY CODICE_FISCALE
124249 HAVING count(CODICE_FISCALE) > 1
124250 [2018-01-17 15:30:39,359460]INFO  (Director       ) PROT: SOS20186717 execute query: SELECT PF_NOME || '--' || PF_COGNOME || '--' || PF_DATANASCITA AS CODICE,
124251     count(PF_NOME || '--' || PF_COGNOME || '--' || PF_DATANASCITA) AS QUANTITA
124252 FROM T7701_VIOLET.STG_WORK_SOS_SOGGETTO
124253 WHERE PROTOCOLLO_SOS = ? AND NATURA_GIURIDICA = 'PF' AND CODICE_FISCALE IS NULL AND PF_DATANASCITA IS NOT NULL
124254 GROUP BY PF_NOME || '--' || PF_COGNOME || '--' || PF_DATANASCITA
124255 HAVING count(PF_NOME || '--' || PF_COGNOME || '--' || PF_DATANASCITA) > 1
124256 [2018-01-17 15:30:39,359465]INFO  (Director       ) PROT: SOS20186717 execute query: SELECT PF_NOME || '--' || PF_COGNOME || '--' || PF_COMUNENASCITA AS CODICI
124257     count(PF_NOME || '--' || PF_COGNOME || '--' || PF_COMUNENASCITA) AS QUANTITA
124258 FROM T7701_VIOLET.STG_WORK_SOS_SOGGETTO
124259 WHERE PROTOCOLLO_SOS = ? AND NATURA_GIURIDICA = 'PF' AND CODICE_FISCALE IS NULL AND PF_DATANASCITA IS NULL
124260 GROUP BY PF_NOME || '--' || PF_COGNOME || '--' || PF_COMUNENASCITA
124261 HAVING count(PF_NOME || '--' || PF_COGNOME || '--' || PF_COMUNENASCITA) > 1
124262 [2018-01-17 15:30:39,359468]INFO  (Director       ) PROT: SOS20186717 execute query: SELECT CODICE_FISCALE AS CODICE, count(CODICE_FISCALE) AS QUANTITA
124263 FROM T7701_VIOLET.STG_WORK_SOS_SOGGETTO
124264 WHERE PROTOCOLLO_SOS = ? AND NATURA_GIURIDICA = 'NPF'
124265 GROUP BY CODICE_FISCALE
124266 HAVING count(CODICE_FISCALE) > 1
124267 [2018-01-17 15:30:39,359471]INFO  (Director       ) PROT: SOS20186717 execute query: SELECT NPF_PARTITAIVA AS CODICE, count(NPF_PARTITAIVA) AS QUANTITA
124268 FROM T7701_VIOLET.STG_WORK_SOS_SOGGETTO
124269 WHERE PROTOCOLLO_SOS = ? AND NATURA_GIURIDICA = 'NPF'
124270 GROUP BY NPF_PARTITAIVA
124271 HAVING count(NPF_PARTITAIVA) > 1
124272 [2018-01-17 15:30:39,359474]INFO  (Director       ) PROT: SOS20186717 execute query: SELECT NPF_DENOMINAZIONE || '--' || NPF_SEDELEG_IND_COMUNE || '--' || NUMI
124273     count(NPF_DENOMINAZIONE || '--' || NPF_SEDELEG_IND_COMUNE || '--' || NUMERO_REA) AS QUANTITA
124274 FROM T7701_VIOLET.STG_WORK_SOS_SOGGETTO
124275 WHERE PROTOCOLLO_SOS = ? AND NATURA_GIURIDICA = 'NPF' AND CODICE_FISCALE IS NULL AND NPF_PARTITAIVA IS NULL
124276 GROUP BY NPF_DENOMINAZIONE || '--' || NPF_SEDELEG_IND_COMUNE || '--' || NUMERO_REA
```

Figure 10: Format of the Log index A.

In such type of logs, in order to be able to carry out a correct extraction of significant fields, it must taken into account the **'event breaking'**.

5.2.3. Data Extraction

The step mentioned above is called **'event extraction'**. It consists of identifying the line or group of lines that constitutes the information relating to an action. The **'extraction of a field'** consists in identifying precisely what can be considered a field, for its sequential position in a list of values, assuming the field is not already in the form of a key=value. This is a process for the that Splunk administrator, which creates value in the data sets.

5.2.4. Metadata fields

It should be noted that there are three most important default fields: host, source, and sourcetype which are also called metadata. They describe where the event originated.

'Host' – it indicates the name of the host, the device, the IP address, or the domain name of the network from which the event originates.

‘Source’ – it indicates the name of the File, stream, or any other input from which the event originates. In the case that the data comes from directories and files specified by the configuration files, the source value is the path, e.g. /archive/server1/var/log/messages.0 or /var/log/. If the data source was based on data from a network, such as a protocol or port, it would be UDP:514.

‘Source type’ – this field stands for the format of the data of the source from which it originates. This value determines how the data will be formatted. Other default fields include date and time fields, which add more search ’granularity’ to event timestamps. Finally ‘Splunk Enterprise’ adds other default fields classified as ’internal’ fields.

6. RESULTS & DISCUSSIONS

6.1. Examples of searches on indexes

Once the meaningful fields have been extracted, using the interactive extractor and the regular expressions, the search query can be run. This is possible using SPL (Search Processing Language) and after the extraction of of the relevant fields. Here below will be reported the main searches compiled for the present work.

Query 1: SQL Query Times

This search query asks the system, to list the first 10-time intervals spent in running the sql queries of the application, sorted by percentage distribution, from the first, most frequent, to the last. As it can be seen from the result of the query in Figure 11, 47% of SQL queries last 1 second; about 29% last 2 seconds.

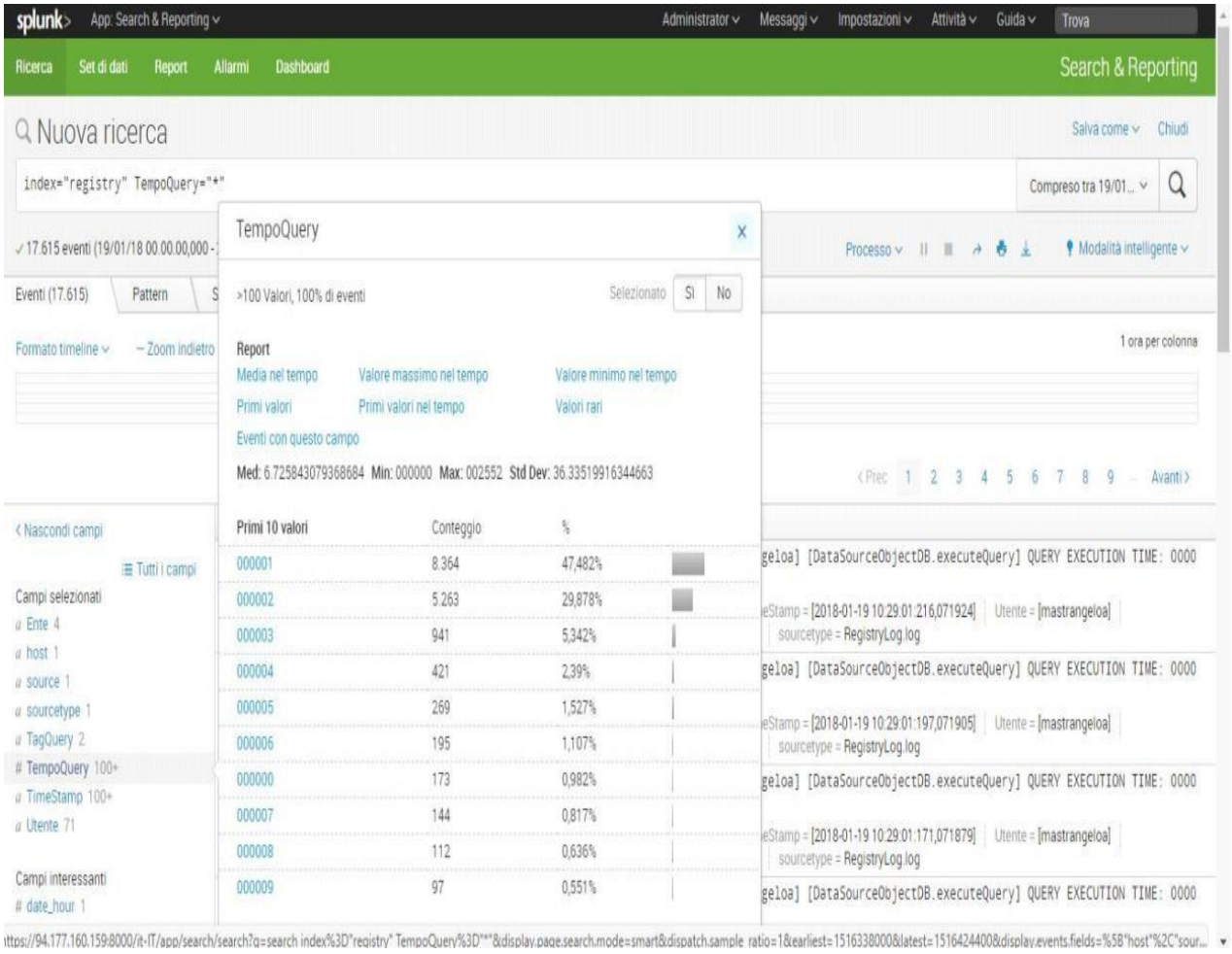


Figure 11: results of Query 1

Query 2: SQL Query Times (actualized)

In this query, carried out on the log, we asked the system to make a ’transaction’, that is to group events having in common certain characteristics. In particular, it was asked to display all the queries on the axis of the abscissae, and the relative times taken to run that query, on the axis of the ordinates. By passing the mouse on the single line of the histogram, the query in question can be displayed. You can see, in the Figure 12, for the purpose of determining possible anomalies, a query whose time is more than 1500ms.

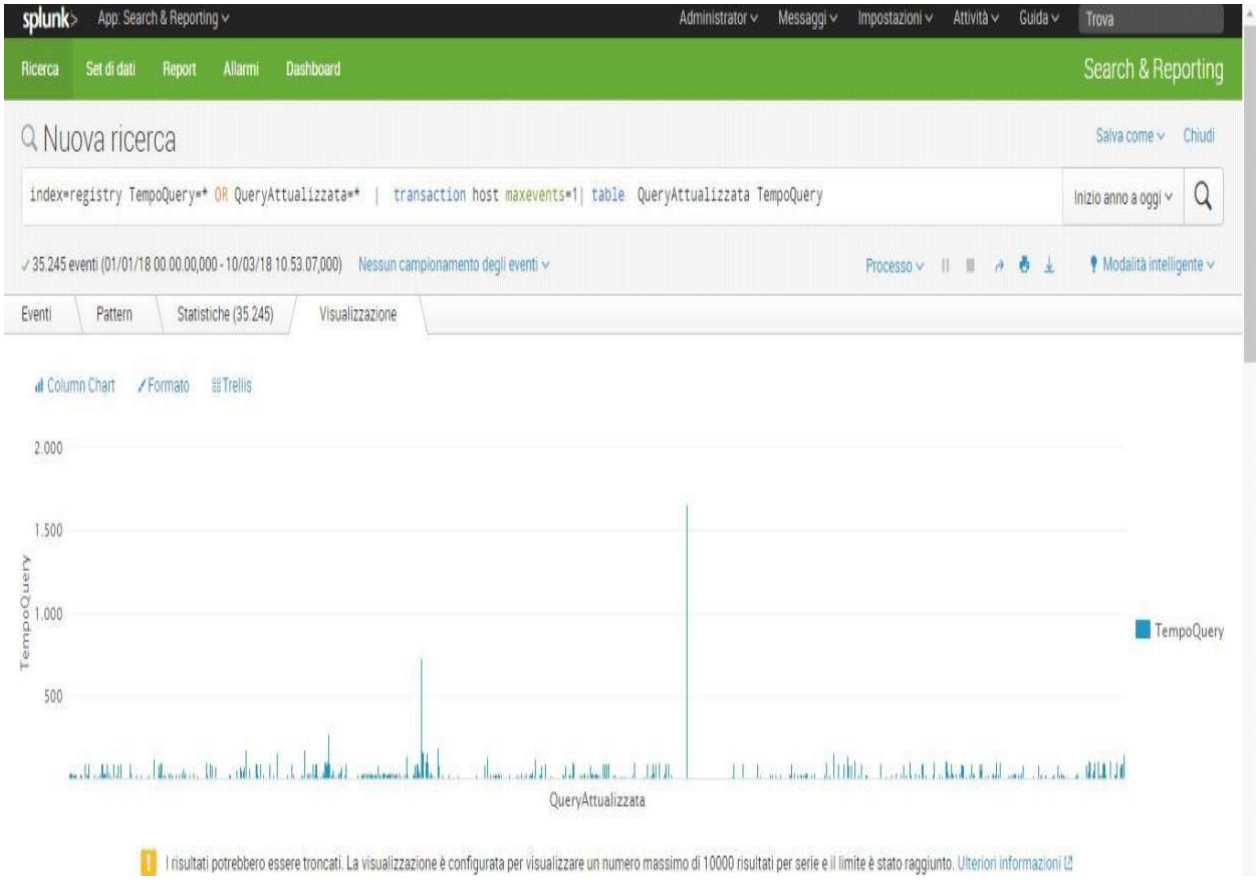


Figure 12: Results of Query 2

Query 3: SQL Query Times (anomalies)

In this query shown in the Figure 13, the maximum value recorded on all queries ran on the system is asked. This value is enough high to be considered an ‘outlier’.

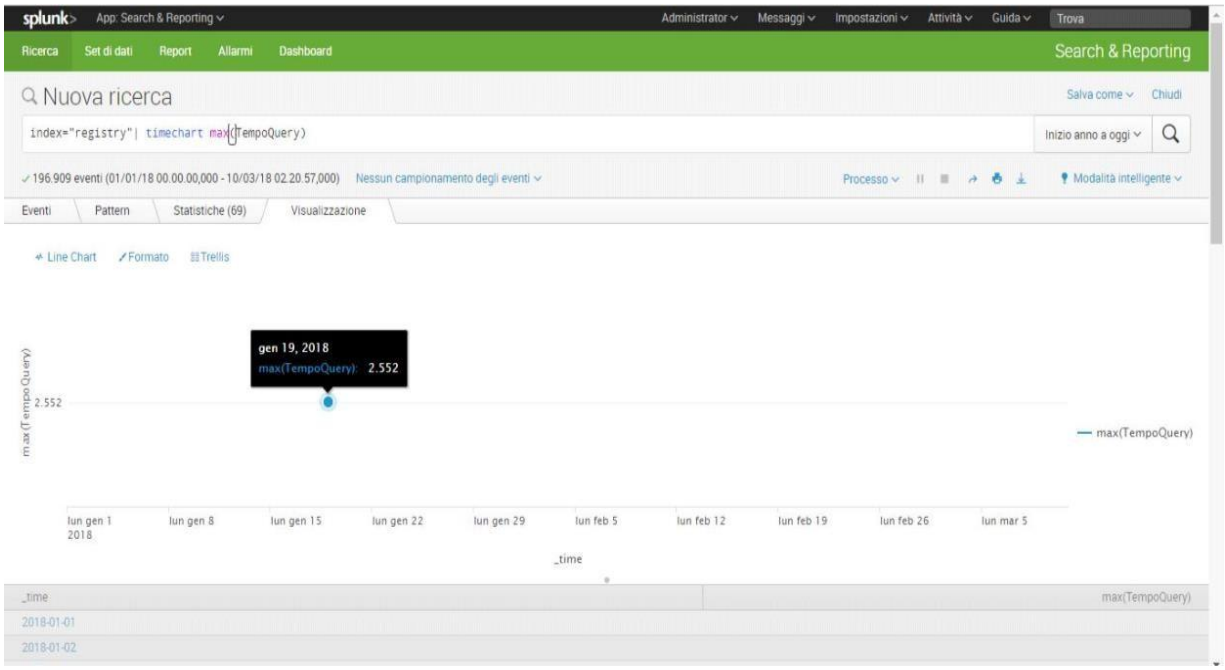


Figure 13: Results of Query 3

Query 4: Incomplete Transactions on Services

In the following search query, carried out on the index log, it was asked to display all incomplete transactions, i.e. all those that have a service login that does not match a service logout, and/or vice-versa. In the pie chart in the Figure 14, there are 1271 total events found, distributed on two dates, 17/1/2018 and 18/1/2018.

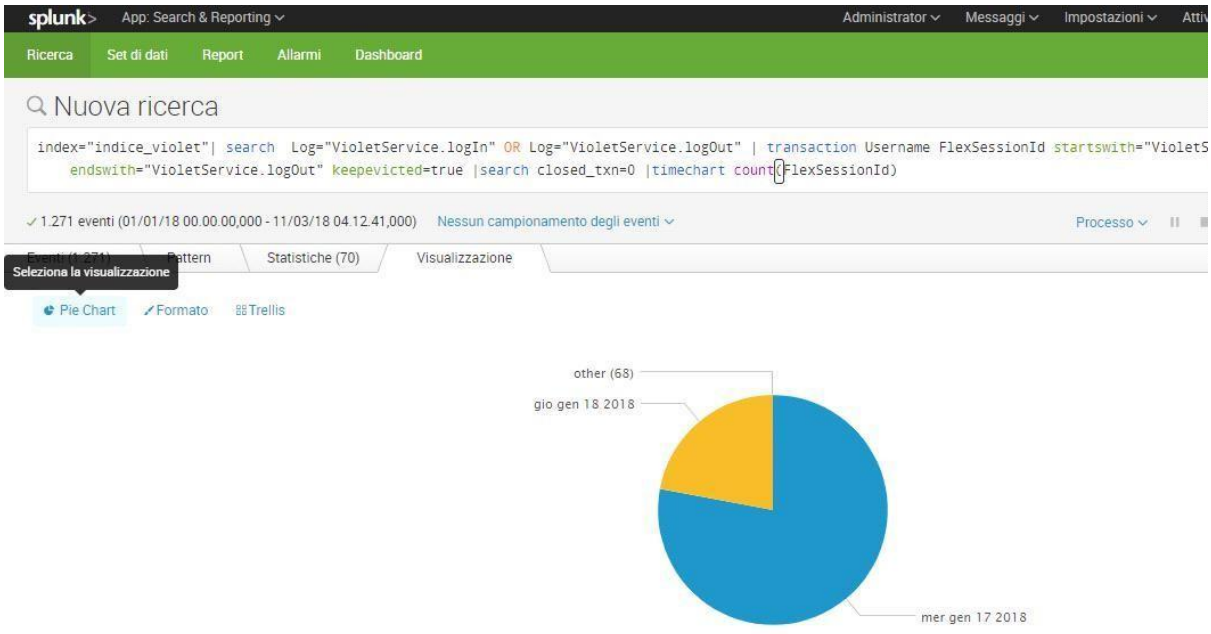


Figure 14: Results of Query 4

Query 5: Measure of Application Load: 'Transaction'

In this query (Figure 15), an estimation of the application load is inferred, by assuming that the "Log" field, gives us an approximation of this measure. Since we can't directly measure CPU cycles, the number of service operations, indicated by "Log" field, divided by time slot, is likely one of the possible load measurements.

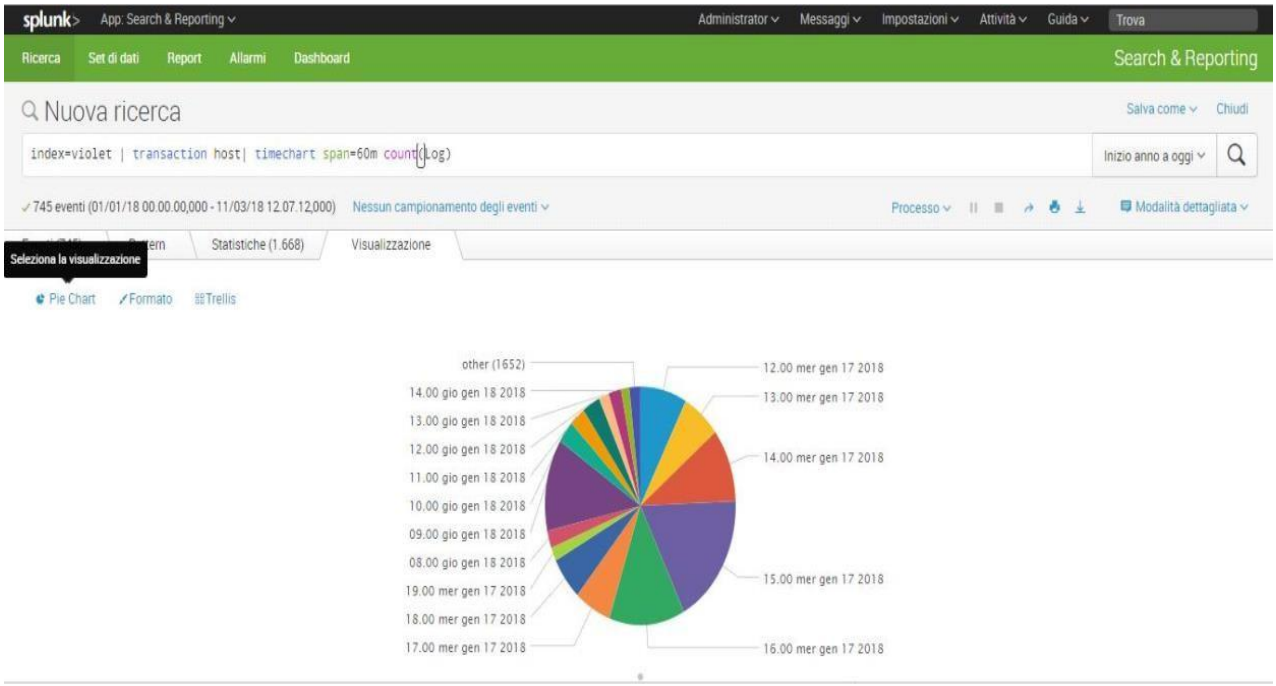


Figure 15: Results of Query 5

Query 6: Application Load Measurement: 'System pauses'

In the following query carried out on the log (Figure 16), it was asked to highlight all system pauses longer than 2 seconds, and report them with respect to the time on the abscissae axis. The pauses are deduced from the event timestamp. The interpretation given to the result of the query is that there are periodic instants of time, coinciding with the hours or half hours, in which there are breaks greater than 2 seconds. It can be supposed that these application pauses may coincide with higher-level tasks, which the system or application must perform in those deadlines.

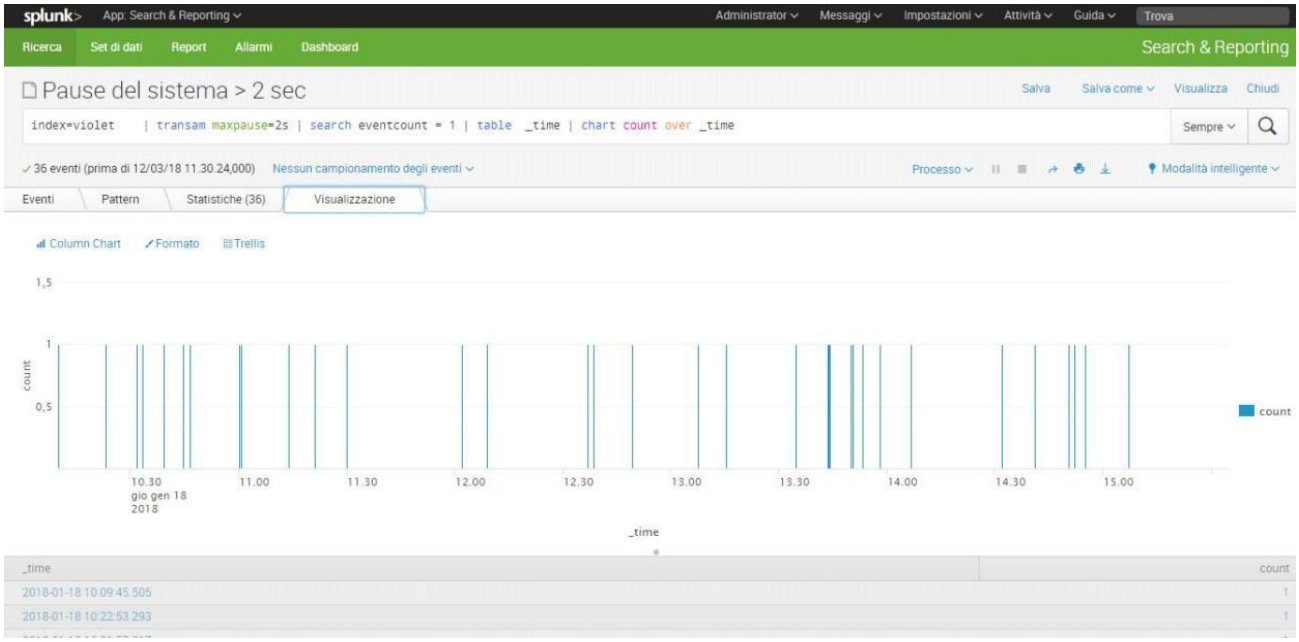


Figure 16: Results of Query 6

Query 7: Application Load Measurement: 'event concentration'

This query also can be considered a measure of the application effort of our system, as it is a distribution of the time intervals between two successive events. As you can see in Figure 17, there are more events at a distance of 0ms, in respect to those at 0.1ms, and to those at 0.2ms time distance. This fact highlights a big commitment of the system also, and probably, in a multithreading mode.

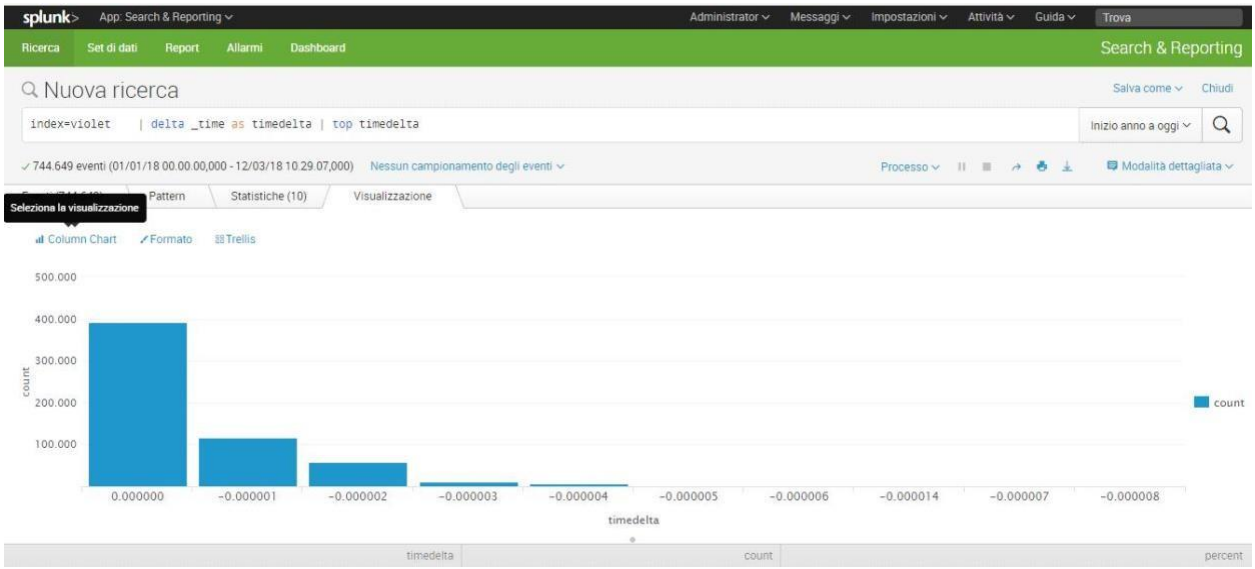


Figure 17: Results of Query 7

Query 8: Broken Authentication and Session Management Control

This query is genuinely a security check to ensure against attacks such as 'Broken Authentication and Session Management'. This security check is done in the web server log file and could be useful for both external and internal access verification. There is a type of attack of this type called 'session fixation attack' that occurs by passing on the URL, in the request, the session identifiers preset and invalid. In addition, in the search query, neither the user nor the password should be passed in full, as this constitutes an exposure of sensitive data in the URLs logged into the proxie web or stored in the cache. The security control contains the following control strings:

1) **"/login\.jsp.*\?.*(userId | password)=./"**

Which is the login.jsp script followed by '?', User and password

2) **"/;jsessionid=./"**

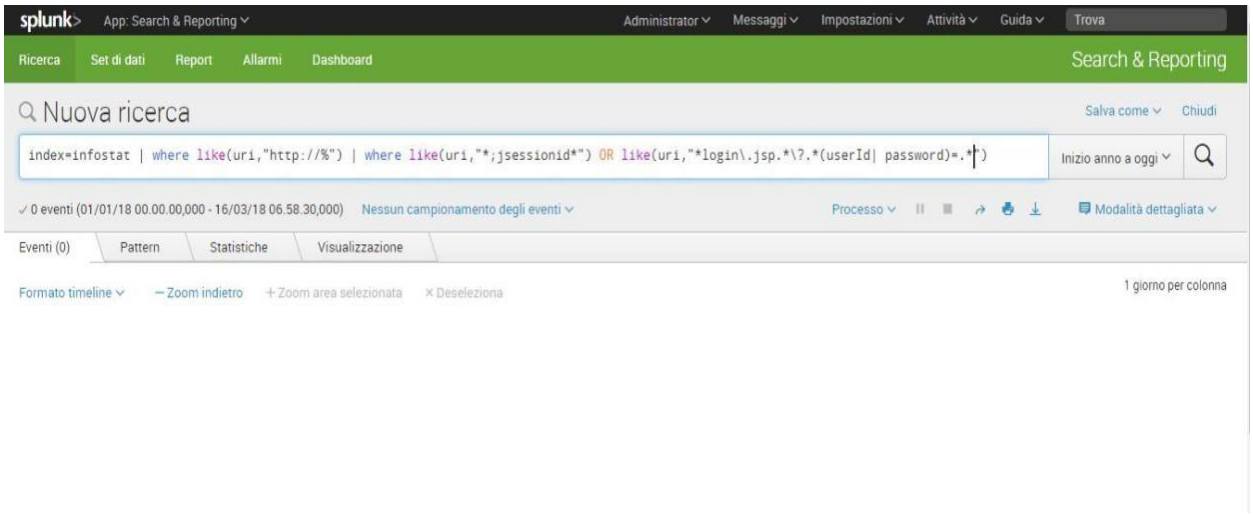


Figure 18: Results of Query 8

In the following pages will be reported further control rules against possible attacks, according to the OWASP classification, potentially taking place in the web farm, and recorded on the logs.

7. QUERY EXECUTION TIMES AND PERFORMANCE

Here below are reported some execution times in order to evaluate Splunk performances and its degree of interactivity with the user. Obviously, it should be taken into account that the machines have been designed with very low power compared to what is required by the HW requirements. This is because of cost limitation reasons, and also because of the stress test of the system was not in the main objectives of the present work.

Not all search queries ran in this research prototype are the same in terms of power and resource requirements. For example, the commands such as 'cluster', 'transaction', 'correlated', 'transam', as well as the search combined with pipes in sequence, absorb a lot of CPU and therefore increase waiting times, which rarely go above a minute even for the most complex queries.

Queries can be optimized to increase performance. In addition, if called in program mode, they are further optimized as they do not engage the web interface.

To examine the time taken just inspect the process performed for the query, as shown below. For the execution of the query: "ERROR" the inspection of the process reports the following times:

1,018 seconds for the execution of the query:

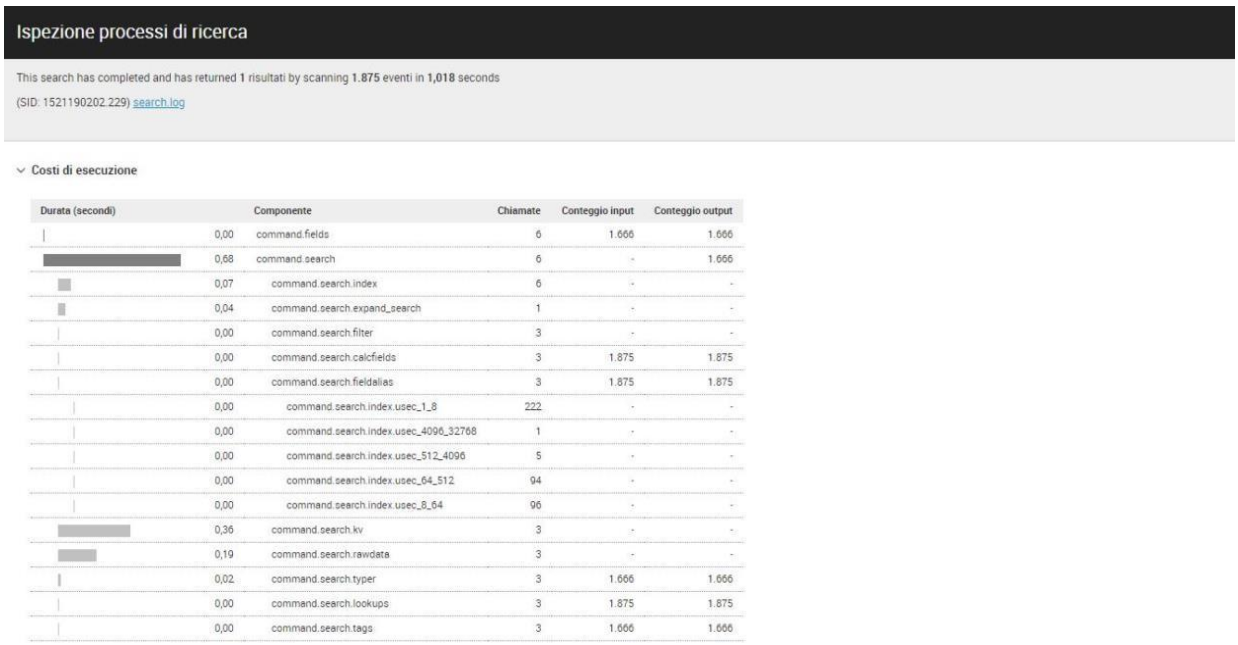
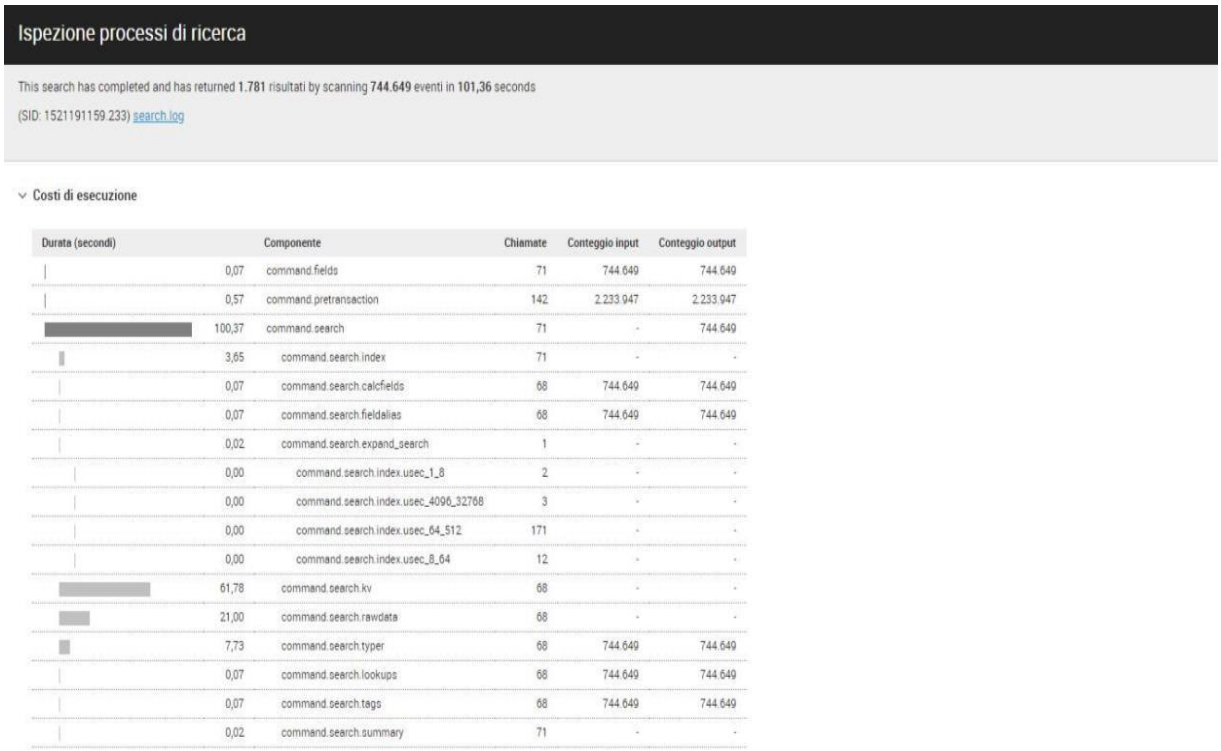


Figure 19: Error Process Inspection

to execute the query: 'CaricoApplicativoLog' the process inspection reports the following times:

101.36 seconds for the execution of the query:



Durata (secondi)	Componente	Chiamate	Conteggio input	Conteggio output
0,07	command.fields	71	744.649	744.649
0,57	command.pretransaction	142	2.233.947	2.233.947
100,37	command.search	71	-	744.649
3,65	command.search.index	71	-	-
0,07	command.search.calcfields	68	744.649	744.649
0,07	command.search.fieldalias	68	744.649	744.649
0,02	command.search.expand_search	1	-	-
0,00	command.search.index.usec_1_8	2	-	-
0,00	command.search.index.usec_4096_32768	3	-	-
0,00	command.search.index.usec_64_512	171	-	-
0,00	command.search.index.usec_8_64	12	-	-
61,78	command.search.kv	68	-	-
21,00	command.search.rawdata	68	-	-
7,73	command.search.typer	68	744.649	744.649
0,07	command.search.lookups	68	744.649	744.649
0,07	command.search.tags	68	744.649	744.649
0,02	command.search.summary	71	-	-

Figure 20: CaricoApplicativoLog Process Inspection

8. REPORT

8.1. The Infrastructure Application Dashboard

A Dashboard has been created (Figure 21), that summarizes the data obtained from previous search queries, inserting the results in quadrants that rapresent the number of errors or warnings, the performances, the application load, the security alerts. The idea is that these values could eventually be summarized in a single value, the Key Performance Indicator (KPI), which indicates the overall performance of the infrastructure application system and should be linked to the Service Level Agreements established by the organization, to internal and external users.

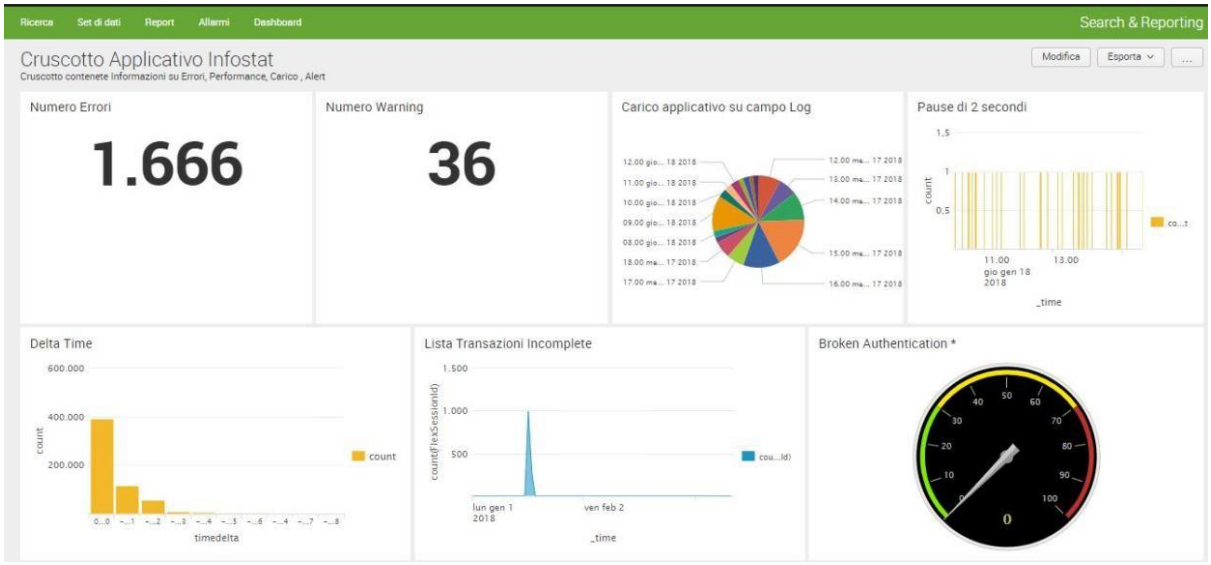


Figure 21: Infrastructure Application Dashboard

8.2. Bringing Splunk to the Web

The Dashboard created so far is a synthesis of information extracted from the logs of the infrastructure application. As it has been realized, it suffers from a certain staticity and therefore the aim of the work was to make its use more interactive in order to allow those users who consult it to have more freedom of action and deepening or drilling into its values. It was considered also as an important point the portability aspect and the easy installation on any machine of the infrastructure. For this reason, it was decided to make it an app.

An app in Splunk is nothing more than a set of configuration files.

Even if the dashboard is a way of rendering a lot of information in a synthetic, visual and strategic way, without the need to use SPL commands, it is important to give the user the possibility to explore it further.

For these reasons this APP was created which has its own navigation menu and which displays the application dashboard.

The app, set in a package, has been evaluated as meeting the criteria defined by Splunk with the AppInspect control, and therefore ensures its quality and robustness standards.

It can be activated by clicking on the icon of the Splunk apps installed and it has been 'branded' by inserting the logo of the Department of Informatics. By clicking on this icon the application dashboard, shown in the following image, can be accessed (Figure 22)

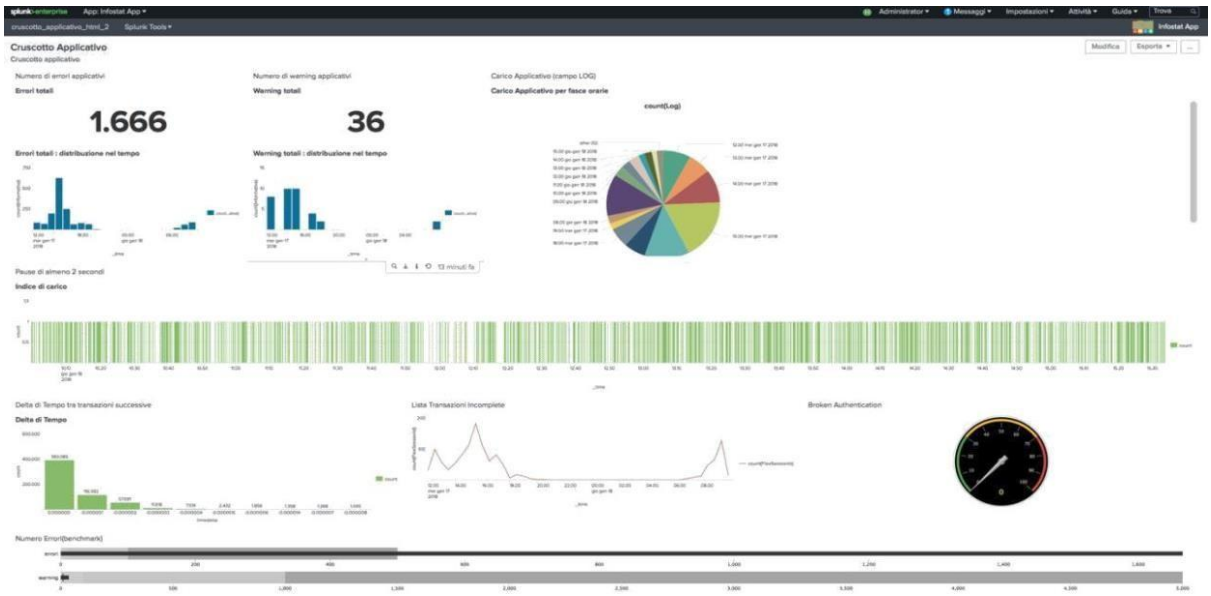


Figure 22: Dynamic Application Dashboard

9. HACKER'S ATTACK CONTROL RULES

9.1. Control rules for the safety of the infrastructure

The infrastructure architecture we have taken under consideration in our research project is a distributed one, with multiple applications cooperating. It is a Service Oriented Architecture and implements parallelism for the purpose of load balancing and increase of reliability. The infrastructure is divided into a web farm front, i.e. the component exposed to the Internet, and a server farm, i.e. the internal back-end. The logs analyzed so far are only a very small part of the logs of the infrastructure. This choice has been made because the logs selected have been considered relevant to the chosen portion of time (two days), and for they were extracted from a single machine, since the attention was limited to a couple of applications, the ones under developer's responsibility, who are committed to troubleshooting activities. Only application logs were examined because this was the context in which the author's work takes place, being himself an application developer.

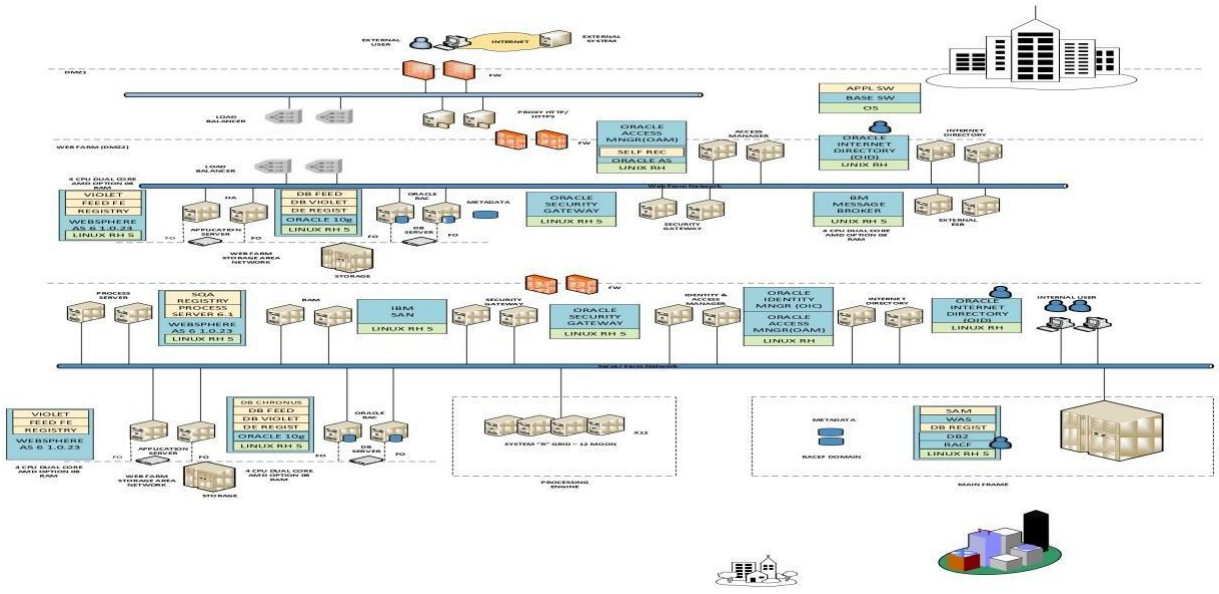


Figure 23: Infrastructure Architecture

The total size of the logs analyzed is about 2GB, covering hundreds of millions of events. The analysis which was carried out, mainly revealed anomalies, and strings were also scanned in order to identify fraudulent authentication attempts.

There is a distinction between the log of a front end and the log of a back end, since as the former has information about connections from external clients (Http), the latter mainly

operates connections with databases or web services (SOAP). Obviously, the logs of the applications in the web farm can allow to identify the attacks coming from outside, while the logs of the applications in the server farm allows to identify attacks form inside.

In the case of attacks from outside, because the field of intervention is level 7 of the OSI architecture (that is the application layer), and because the information used to determine security attacks consists of logs, the rules that will be activated will be based on the 'uri' field, in order to identify the major **security attacks** as cataloged by the organization **OWASP**. These type of attacks, in general, have the purpose of sabotage, or theft of information, or the compromise of the integrity of the data itself. In this case data can be changed and not only stolen. It should be assumed that, in addition to the application logs, should also be scanned the access logs of the Application Server.

9.1.1. Cross Site Scripting (XSS) Control Rule

The XSS attack can be identified by detecting HTML tags relevant to the execution of scripts in the uri field. Therefore, it is a matter of entering a uri field search with the following SPL commands:

'index=index C | where like(uri,'http:/%') | where like(uri,'*<*>*</*>') | chart count'

this search should identify all suspicious requests like: <script>alert('XSS') </script> or <h1> alert('XSS') </h1>

An alternative method is to explicitly list expressions that can activate active codes or javascript. The SPL control rule in this case should be:

'index=indexC | where like (uri, 'http:/%') | where like (uri, '*javascript*') like (uri, '*vbscript*') OR like (uri, '*applet*') like (uri, '*script*') like (uri, '*frame*') |chart count'

9.1.2. Cross Site Request Forgery (CSRF) Control Rule

There are many applications that are vulnerable to CSRF attack. They are those based on the automatic submission of credentials, as session cookies. The CSRF attack, which therefore presupposes an application with vulnerabilities (as an example we suppose it is hosted on the site www.example.com), can consist of putting a post in a forum with a link of the following type:

‘<img src=https:/www.example.comtrx.do? amt=100&toAcct=1234 width= ‘0’ height= ‘0’>’

The request will start from the forum site and will be directed to the destination site (trying to download the image size 0).

The attack therefore comes from the forum site, at the implicit request of the attacker, who does not perform the action directly.

To identify this attack in an access log file, is possible through the use of the referer. In the log, in fact, it will be present the referrer, that the client reports to be the site from which the reference or the link to the resource, requested. The one which the GET has started. From the examination of the log relative to an application that suffers this type of attack we can note something that usually does not happen, that means that the URL (or URI) request, that carries out the transaction, and the referrer, are belonging to two different sites. The SPL commands will be as follows:

‘index=indexC | where like(uri, ‘http:/%’) | where IP != referer’

The referrer value will also be an IP as a result of a lookup table or a workflow action that translates the domain into its IP address.

9.1.3. SQL Injection Control Rule

Let’s consider one of the most recurrent injections, even if many others can be identified. Injection is made using the characters () or (), which are delimiters for queries and comments, and also the character (#). The SPL commands used to check the attack will be:

‘index=indexC | where like (uri, ‘http:/%’) | where like (uri, ‘*‘*’) OR like (uri, ‘*(userId)=*’)’) | OR like (uri, ‘*(password)= ‘1’= ‘1’-’) | chart count’

A possible attack vector, among many possible, could be as follows: or ‘1’= ‘1’. Of course, there are endless variations of these values as 1’ or 1<2. The constant part is the single apex followed by the term OR.

9.1.4. Mischievous File Execution Check Rule

Since applications can allow the user to provide a filename, or part of a filename, this constitutes

a vulnerability in case the input is not validated. Therefore, the attacker can manipulate the filename to run a system program or an external URL. For example, the attacker may attempt to upload an executable file, or a file that calls other files to run or parse. The SPL commands to chech this attack will be :

'index=index C | where like (uri, 'http:/%') | where like (uri, '*.jsp*') OR like (uri, '*.xml*') | chart count'

9.2. Control points

Attacks can be identified in different areas of the Infrastructure architecture, and in each of these areas the available logs can be used. For example, logs originating from firewall, IDS, DBMS and application firewalls (WAF) can be used to complete the knowledge about an attack.

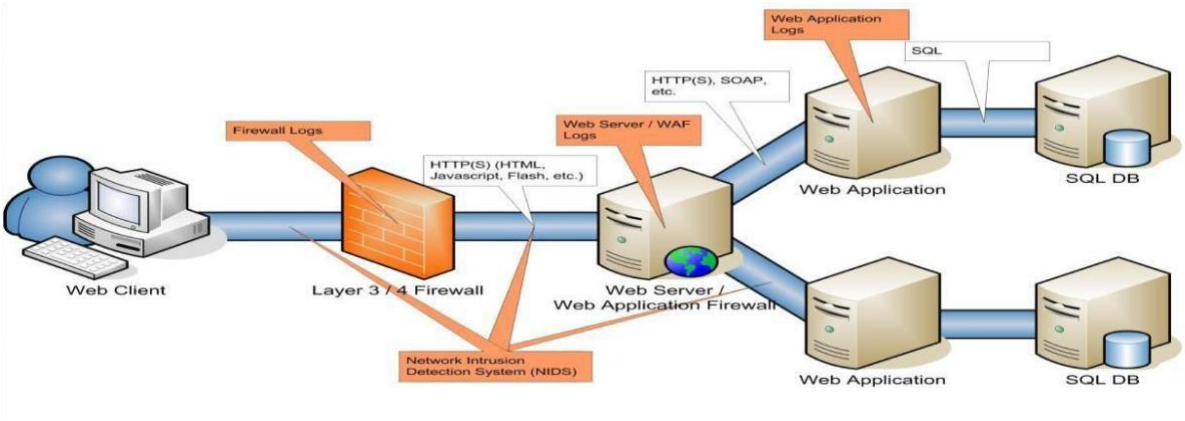


Figure 24: Infrastructure Architecture Control Points

All these rules can only be written if the attack vectors are known before writing down the rules. In this way, we can identify the attacks considered by OWASP to be among the 10 most critical attacks for the applications exposed on the Internet [2]. On the contrary, an approach to identifying statistical anomalies does not require prior knowledge of attack vectors. The nature of the work carried out anyway is experimental and indicative of the potential of the Splunk tool under investigation. It cannot be considered in terms of a product ready for the production environment.

10. PERFORMANCE CONSIDERATIONS

10.1. Further Performance Considerations on Search Queries

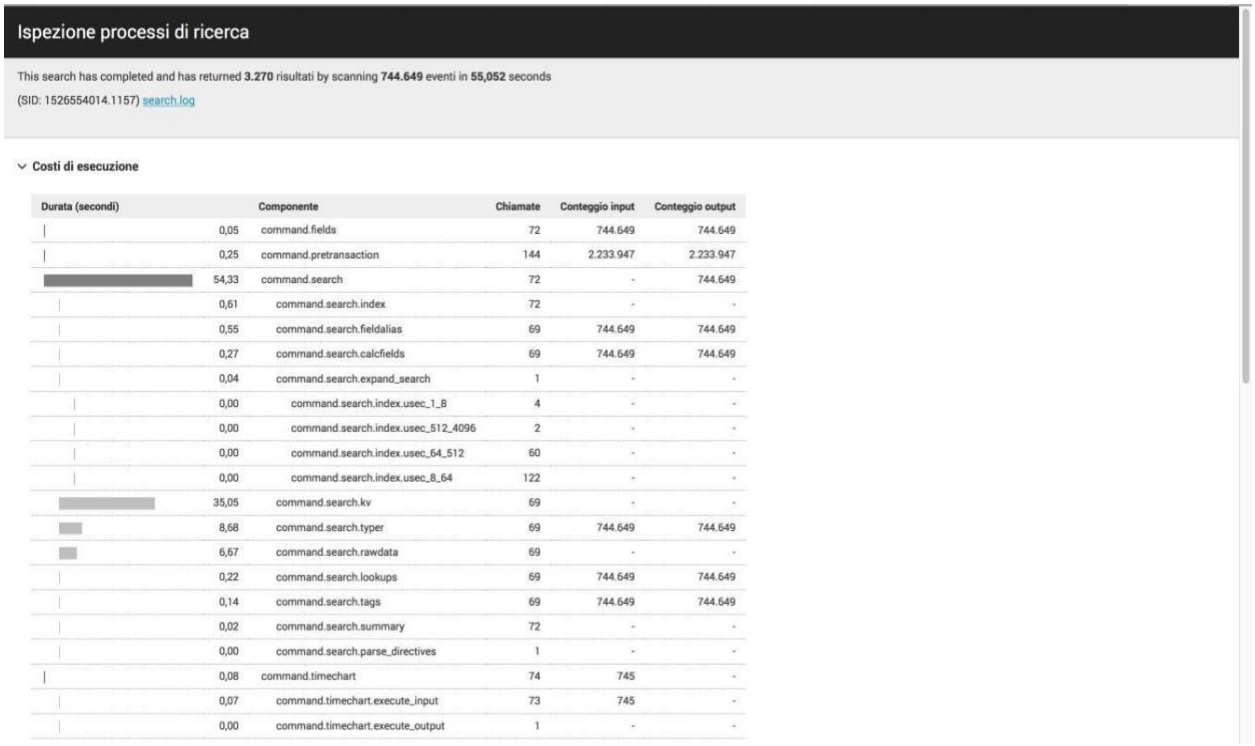
All the searches carried out in this project work have been found to have search times limited to 20-30 seconds. Excepted for the calculation of the load, the search spent 101.36 seconds, compared to 744,649 events consulted. It can be asserted that the execution times depend on the complexity of the search query, the total data to consult, the resources available on the machine (number of CPUs, RAM size, type of secondary memory) and also on the configuration of the Splunk platform. In order to experience the real dependence of execution times on HW resources the RAM size has been expanded, and the number of Virtual CPUs on the cloud has been increased, creating two new indexes: Indexer and Indexer Large (Figure 25):

NOME SERVER	IP PUBBLICO	HYPERV.	SO	VCPU	RAM	HD
UniversalForwarder SMART	80.211.155.253 0% traffico usato	vmware	CentOS 7.x 64bit	1	1 GB	20 GB
Indexer SMART	94.177.160.159 0% traffico usato	vmware	CentOS 7.x 64bit	2	4 GB	80 GB
IndexerLarge SMART	188.213.174.205 0% traffico usato	vmware	CentOS 7.x 64bit	4	8 GB	160 GB

Figure 25: Installed Virtual Machines

After reinstalling on Indexer and IndexerLarge two new configurations of data and programs, it was continued the work consisted on running search queries on the application load made previously. They were obtained the following times results (Figure 26).

What emerges from this experimentation is that by doubling and quadrupling the power of the machine, i.e. going from 2GB of RAM to 4GB and then to 8GB of RAM, and also by doubling the virtual CPUs from 1 to 2, and finally to 4 CPUs, the execution time of the search query falls significantly. In the Figure 23 we can see the data resulting from the execution of the 'application load' query on the Indexer machine, (4GBs of Ram, 2 VCPUs).



Ispezione processi di ricerca

This search has completed and has returned **3.270** risultati by scanning **744.649** eventi in **55,052** seconds

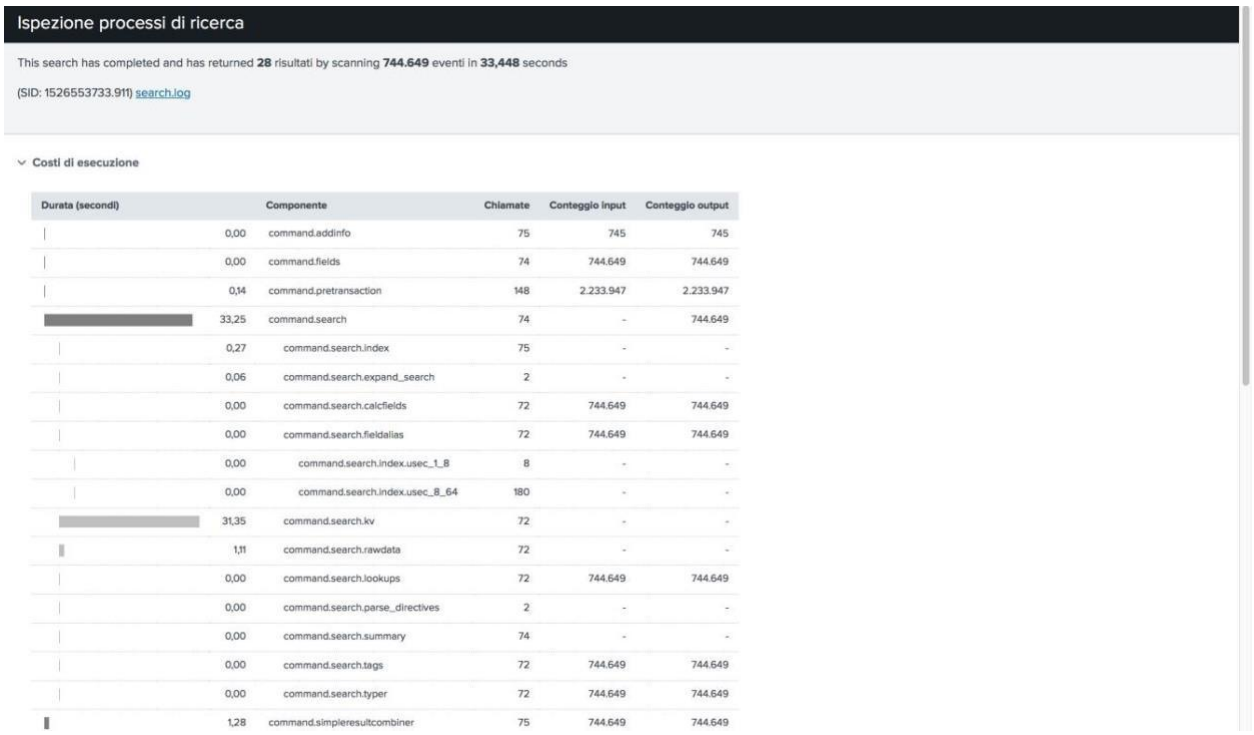
(SID: 1526554014.1157) search.log

Costi di esecuzione

Durata (secondi)	Componente	Chiamate	Conteggio input	Conteggio output
0,05	command.fields	72	744.649	744.649
0,25	command.pretransaction	144	2.233.947	2.233.947
54,33	command.search	72	-	744.649
0,61	command.search.index	72	-	-
0,55	command.search.fieldalias	69	744.649	744.649
0,27	command.search.calcfields	69	744.649	744.649
0,04	command.search.expand_search	1	-	-
0,00	command.search.index.usec_1_8	4	-	-
0,00	command.search.index.usec_512_4096	2	-	-
0,00	command.search.index.usec_64_512	60	-	-
0,00	command.search.index.usec_8_64	122	-	-
35,05	command.search.kv	69	-	-
8,68	command.search.typer	69	744.649	744.649
6,67	command.search.rawdata	69	-	-
0,22	command.search.lookups	69	744.649	744.649
0,14	command.search.tags	69	744.649	744.649
0,02	command.search.summary	72	-	-
0,00	command.search.parse_directives	1	-	-
0,08	command.timechart	74	745	-
0,07	command.timechart.execute_input	73	745	-
0,00	command.timechart.execute_output	1	-	-

Figure 26: Process Inspection on the Indexer Virtual Machine

In the following Figure 27 we can see the data resulting from the overall execution of the ‘application load’ query on the Indexer Large machine (8GBs of Ram, 4 VCPUs).



Ispezione processi di ricerca

This search has completed and has returned **28** risultati by scanning **744.649** eventi in **33,448** seconds

(SID: 1526553733.911) search.log

Costi di esecuzione

Durata (secondi)	Componente	Chiamate	Conteggio input	Conteggio output
0,00	command.addinfo	75	745	745
0,00	command.fields	74	744.649	744.649
0,14	command.pretransaction	148	2.233.947	2.233.947
33,25	command.search	74	-	744.649
0,27	command.search.index	75	-	-
0,06	command.search.expand_search	2	-	-
0,00	command.search.calcfields	72	744.649	744.649
0,00	command.search.fieldalias	72	744.649	744.649
0,00	command.search.index.usec_1_8	8	-	-
0,00	command.search.index.usec_8_64	180	-	-
31,35	command.search.kv	72	-	-
1,11	command.search.rawdata	72	-	-
0,00	command.search.lookups	72	744.649	744.649
0,00	command.search.parse_directives	2	-	-
0,00	command.search.summary	74	-	-
0,00	command.search.tags	72	744.649	744.649
0,00	command.search.typer	72	744.649	744.649
1,28	command.simpleresultcombiner	75	744.649	744.649

Figure 27: Process Inspection on Indexer Large Virtual Machine

10.2. Costs of query execution

Therefore, the query which lasted 101.36 seconds on the original virtual machine, has dropped to 55.052 sec. on the doubled power virtual machine, and then to 33.448 sec. on the quadruplicated power one. The law is almost linear so it can be assumed that with 32 GB of RAM and 16 CPUs the time spent in the search query will be about 7 seconds, or 3 seconds if a 64 GB and 32 CPUs will be used. This is in line with the HW requirements recommended by Splunk for Enterprise platforms.

From the process inspection above you can also see that the components that contribute most to the overall time of the queries are related to "command.search", in particular its component 'command.search.kv'.

In the case of execution with the major HW configuration, what happens is that after Splunk software identifies the events containing the indexed fields that meet the search queries, the events are further analyzed to see which of them also meet the other search criteria of the query.

The subprocess 'command.search.kv' (32.11 sec.) tells us how long it took to extract the fields from the events.

In this case it is more than 90% of the total time of the search and much more relevant than the component 'command.search.rawdata' (1.18 sec.) which indicates how long it takes to extract events from the file raw data.

This aspect will be further analyzed in detail to understand why this happens, as behind these services may be hidden inefficiencies caused by the data.

10.3. Organization of Indexes in Splunk

Splunk index data are organized in **buckets**, which are directories.

There are 5 types of buckets: **hot, warm, cold, frozen** and **thawed**. Therefore the data in Splunk have a life cycle moving from hot, to warm, to cold and finally to frozen, making an automatic rolling as you reach the saturation dimension of the previous level. The hot type of bucket is the only one that is not necessarily optimized. Each bucket contains events related to time periods. The size of each of them can be decided by the user when creating and configured the index.

The search for a certain value, a field, always starts from the hot index, and when finished the search proceeds to the next "coldest" field.

When you search for a string and you find it in a bucket you must then access the rawdata to verify the event that contains the string you are looking for, but to do this must decompress the rawdata, which are stored in compressed format.

The more events that satisfy the search, the more decompressions will be necessary to do, and therefore the greater will be the commitment of the CPU.

If the decompressions are few, the CPU used to access the rawdata is a little amount of time. Therefore, this is our case in Figure 24, as the evidence of the process shows that since there are only 28 results (this is the value at the time of execution) out of a total of 744,649 scanned events, the search in question can be considered to belong to the category "scatter", according to the following classification:

- "**Dense"** search 1 result for every 1-1,000 indexed events
- "**Scatter**" search 1 result per 1,000-1,000,000 indexed events
- "**Rare**" search 1 result per 1,000,000-1,000,000,000
- "**Needle in a haystack**" search 1 result per 1,000,000,000-finished

10.4. Reasons for being cautious in the search

In the case of the search query under consideration, we want to further investigate the time "command.search.kv" used to extract the fields of the events, as in the search query there is no field, except the host metadata, and therefore it is inexplicable the high cost of time spent in this process.

The explanation, however, is that the default configuration of Splunk is such that all fields are always extracted. For this reason, we set KV_MODE = none in the props.conf configuration field to disable the automatic extraction of key value pairs. Repeating the query execution will result in a 10% shorter time.

A further reason for attention is the high value of time spent in the subprocess "dispatch.fetch.rcp.phase_0" which is the time spent waiting for the fetching of events from

the search.

The fetch is mostly waiting for events to be extracted from the disk. It may be caused by I/O operations. In fact, since it has been said that the process is not CPU bound, consequently the I/O component prevails over the processing component. A concourse in this case could be a bad structuring of the data in the buckets, but in our case this hypothesis is to be discarded since the hot type bucket has not yet been saturated. Another cause could be the I/O disputes, in which case the reason could be attributed to the supplier of the VPS (Aruba provider), and therefore out of our control.

10.5. Other variables that impact execution times

Finally, let's consider other elements on which the execution costs could depend.

The size of the buckets is important, although it is usually self-contained and depends on the machine cores. Another element may be the "Bloom Filter". These are data structures that categorically establish that an element is not contained in a data set. In this way you can say, without opening a data set within which we are looking for a string, that a certain data is not contained in it.

This increases performance considerably. Bloom filters are created in the indexing phase and are active by default in Splunk.

The speed of writing and reading to disk also has an impact on search performance. The indexing phase depends on the write performance of the disks; while the search phase is partially dependent on the read performance. The SSD disk type performs best for searching and is estimated a logarithmic distance with the performance of the disks of mechanical type.

11. MACHINE LEARNING AND STATISTICAL APPROACH

Machine Learning to determine the anomalies

According to Gartner's PPDR (Predict, Prevent, Detect, Respond) model, all security tasks can be divided into five categories: **prediction; prevention; detection; response and monitoring**. We are focusing on the **detection** task of cybersecurity.

The methodology followed so far by using the Splunk tool was based on the identification of events made through regular expressions, by which the individual important fields where in-identified. On these fields I the queries were built and the results to be displayed in the dashboards were extracted. The dashboards were successively made dynamic and further explorable thanks to drill down.

Let's suppose that something could escape this process of log mining. The process we are going to describe is called 'Data Mining'. The process involves the extraction of patterns from the data and the fitting of the model built on it. The concept behind the fitting of the model is handle the information which can be inferred from processing such a model. That's why we evaluated the use of **Machine Learning** techniques, which allow us to discover those events that may have escaped our investigation or that were not thought to exist. Machine Learning is the machine's ability to learn automatically without being programmed by a man. It is a method to generalize from the data contained in the logs, and build models. The activities to be carried out in the Machine Learning activity are the following:

1. Collect data
2. Clean and transform data
3. Explore and view data
4. Modelling data
5. Evaluate and refine the results of the model
6. Release the model (for future predictions and uses)

The activity carried out in the Machine Learning and statistical approach of this project work has consisted in the installation of the tool **MLTK version 3.2.** This is an app downloadable

from Splunk base and executable in the infrastructure, with which, thanks to the libraries of Python's scientific computing, the SPL commands can be extended to the field of Machine Learning. The MLTK toolkit allows to perform various activities including:

1. Predict numeric fields
2. Predict categorical fields
3. Identify numerical outliers
4. Identify categorical outliers
5. Provide time series
6. Cluster numerical events

Within this paper some experimentations applied to the application logs will be reported, and thus producing the results related to the activities 2. (prediction categorical fields) and 4. (identify categorical outliers). An anomaly is defined as the unusual behavior or pattern of the data. It particularly may indicate the presence of the error in the system and itdescribes that the actual result is different from the obtained result thus concluding that the applied model does not fit into the given assumptions. In other words, the model drawn in the data mining phase indicates that the actual result is different from the obtained result from the model. In a more general meaning the **anomaly does not necessarily mean an error** but **it indicates a highly probability of it.**

Both experiments play an important role in the determination of anomalies for the reasons mentioned at the beginning. In the present section, the results of Machine Learning activities will prove to be important both for the determination of malfunctions and also for the identification of events potentially subject to further investigation. By analyzing the log data anomalies, the security of the system could be improved and the potential intrusions could be detected.

In the first part of this paper, we have presented the reader some empirical ways of detecting web attacks. The Machine Learning (ML) for Application Security is an area in which the use of Machine Learning can improve app security. There are web applications, or web services, or micro services, in which ML can solve some of tasks relevant to cybersecurity.

Classification and Regression

In order to apply the concepts of Machine Learning for application security there are two main possible ways:

- classification to detect known types of attacks like injections˘
- regression to detect anomalies in HTTP requests

These two techniques are applicable in different kind of problems. The former when we are looking for a classifications output (the output variable is over or below a given threshold? is or is not? etc.). The latter is characteristic of a numerical and precise value of the answer (for example the output variable is 5?). In the case under examination we will take into consideration a classification task, namely a prediction of categorical fields (activity 2).

But we can also distinguish regression from classification according a different aspect.

Regression (or prediction) is a task of predicting the next value based on the previous values.

Classification is a task of separating things into different categories.

Supervised and Unsupervised Learning

The tool at our disposal (MLTK) allows us to use both ’supervised’ or ’unsupervised’ learning.

That is, the construction of the model that is obtained is in the first case guided by examples (supervised) in which the system is made aware of the correspondence, in a set of training data, between the output (the field to be predicted) and the input (the so-called ’predictor’ fields). In the second case it is assumed that the correspondence between the data of the input training set and those of the output is not known, and the same are treated indistinctly at the end of the construction of the model.

Logistic Regression, SVM, PCA are examples of supervised learning.

Linear Regression, KBMeans, DBSCAN, Spectral Clustering, are examples of unsupervised learning.

In both cases input data are provided to the Machine Learning algorithm, which will proceed to a training and testing phase of the model.

Data pre-processing and data transformation

In the work done in this project work the data are provided as a result of a search, coming from a CSV data source, and they have been previously 'cleaned' and then 'transformed'. The cleaning process consists in making the data as structured as possible and this takes place also by eliminating unstructured information that may occur within the logs.

Subsequently, the transformation can intervene, optionally, in the 'pre-processing' part of the data. In this phase the values can be 'standardized' with respect to the average and standard deviation, in the case of large numerical values that otherwise could complicate the calculations.

In addition, it may be necessary to bring the data back into tabular form, because often the scientific calculation algorithms, by which the data are processed, require matrices.

Pre-processing uses algorithms such as Fieldselector (which selects the fields that are the best predictors), PCA (which reduces the number of fields), KernelPCA (which reduces the number of sizes), StandardScaler (which standardizes the fields).

In this work reported in this paper we have carried out pre-processing, using the KernelPCA algorithm, in activity 2. (prediction of categorical fields), in order to reduce the number of fields, i.e. reduce the size, and make more efficient the computation. In some cases, KernelPCA or PCA can be used, to make it easier to view the final data, reducing the size to two.

Category Outlier Prediction

Category Outlier Prediction

Let's see below the phase of the pre-processing (Figure 28):

Figure 28: Category Outlier Prediction Preprocessing

After setting the method , the individual fields to be pre-processed and other optional fields, it must be pressed the ‘Apply’ button to start the pre-processing.

At the end of the operation the data will be visible in a preview with the pre-processed format, including fields PC_1, PC_2, related to the resizing.

It is also possible to put in sequence other pre-processing phases, adding a step, with the ’Add a step’ key.

Once the pre-processing is finished, we will have the data in the desired format, and we will be able to pass to the real ‘learning’ phase.

Learning phase: ‘fit’ and ‘apply’

The MLTK tool allows the user to extend the SPL language, which is thus enriched with new search commands, among which the most used are :

- **’fit’** is a command that serves to ’learn’ a model from the data, search results (or lookup)
- **’apply’** is a command that applies a learned model (with the fit command) to a new data set

The "fit" command requires you to specify the algorithm to be used.
Some of these algorithms are: LogisticRegression, LinearRegression, OneclassSVM , BernoulliNB, GaussianNB, GradientBoostingClassifier,˛ RandomForecastClassier,˛ SGDClassifier,˛ SVM.

Will not go into the details of the algorithms, which belong to the open source libraries of Python (more than 300 open source programs), of the libraries sci-kit learn, pandas, statsmodel, numpy and scipy, available thanks to the add-on Splunk Python for scientific computing.

The result of the ’fit’ command will be the model, which will be given a name (see in the Figure 26, the name given is ‘example_prova’) , and which will be an object of the Splunk platform, and therefore reusable in a later moment, with different data sets.

The ’fit’ phase of the model for the previously pre-processed data will appear as shown in the following Figure 29:

Figure 29: Model Fit Phase

The Figure 26 shows that the Logistic Regression algorithm will be used as a supervised learning method, in which the data for training and testing are 50% each.

Model validation consists in training the model with a portion of the data (training set) and testing the model with another portion (test set).

Moreover, the field to be predicted, called response field, is 'service'. The 'service' field is a field of the log, which has been appropriately pre-processed, and which indicates the application service used.

The remaining fields are predictor or explanatory fields (i.e. info, timestamp, FlexId, code). Each activity of the learning can be implemented using different algorithms such as linear regression, logistic regression, statistical distribution, probability measurements, Kalman Filtering states space method, K-Means, spectral clustering, Birch, DBSCAN.

The model, including the pre-processing phases, is stored with its identifier and can then be used to 'apply' it to another set of data.

The commands of the SPL language to realize the 'fit' phase of the model are the following:

"| inputlookup indexB-learning.csv | FIT KernelPCA "FlexId", "codice", "info", "service", "timestamp", "username" k=2 INTO modello_previsione_KernelPCA_0 | FIT LogisticRegression fit_intercept=true "service" FROM "FlexId" "codice" "info" "timestamp" "username" "PC_1" "PC_2" INTO "modello_previsione" "

The model so far created can subsequently be stored among Splunk's objects, with a view to its possible reuse. The next action is to apply the model to the remaining part of the data. The following SPL command will be issued to splunk :

"| inputlookup indexB-learning-c.csv | apply modello_previsione_KernelPCA_0 | apply "modello_previsione" | classificationstatistics("service", "predicted(service)" "

These commands apply pre-processing models and then apply the prediction model to the entire data set to predict the "service" field. The macro ’classificationstatistics’ is also used to calculate the precision, recall, accuracy, and F1, which are quantities that indicate the goodness of the model obtained. The closer these values are to 1, the best the model is considered.At the end of the execution of the apply commands the following table will be displayed Figure30):

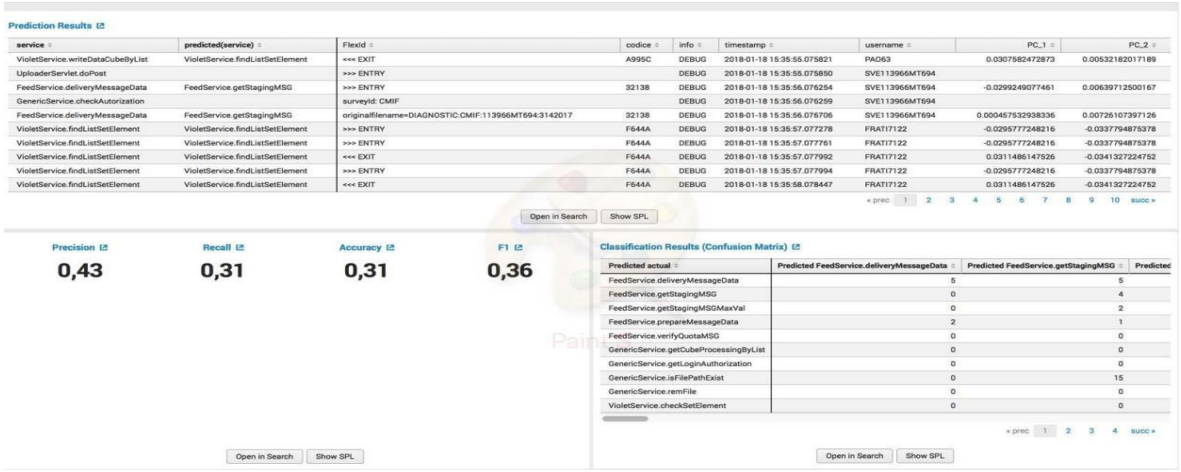


Figure 30: Result of applying the data model

A Confusion Matrix will also be created that will indicate the result of differences between the predictive model compared to real data. The more the data will be distributed along the diagonal, the more accurate the model will be (Figure 31).

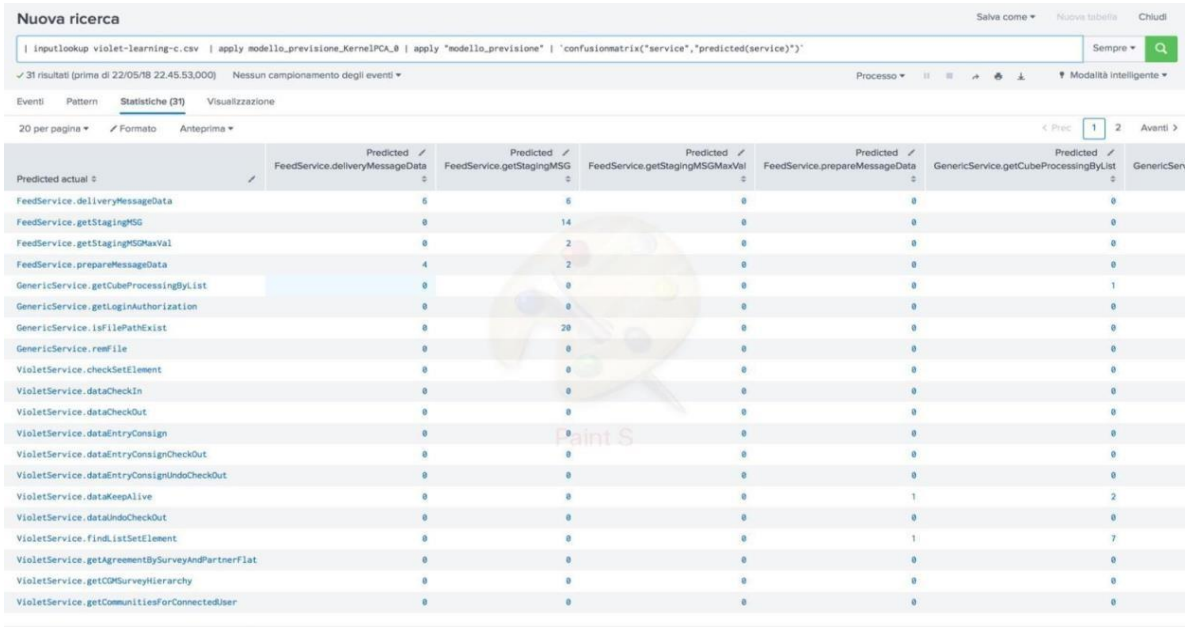


Figure 31: Confusion Matrix

Final result of model processing: refining of the model

The final result that will be obtained is the result of successive attempts in which all the parameters have adjusted in order to obtain a result as reliable and accurate as possible. The process of adjusting the parameters is called ‘refining of the model’. This is the hardest and most expensive part of the work, also because it requires resources and time.A model can be refined by removing distracting fields or even adding more fields to increase accuracy.Once the model has been validated and refined can be considered valid and therefore usable in all those cases of log data having the same structure.The meaning that can be drawn from the experimentation reported so far is that the anomaly becomes evident when there is a deviation between the current result and that predicted, with the premise that the model is considered a **good model**. The anomaly must therefore be investigated and further elaborated as a sign of a ’potential’ problem.

We tried to repeat the experiment carried out with respect to the field to be predicted ’service’, also with respect to the field ’username’, obviously with other predictors, and we obtained an even more reliable model, with a precision = 0.99.

The result of this elaboration is shown below (Figure 32):

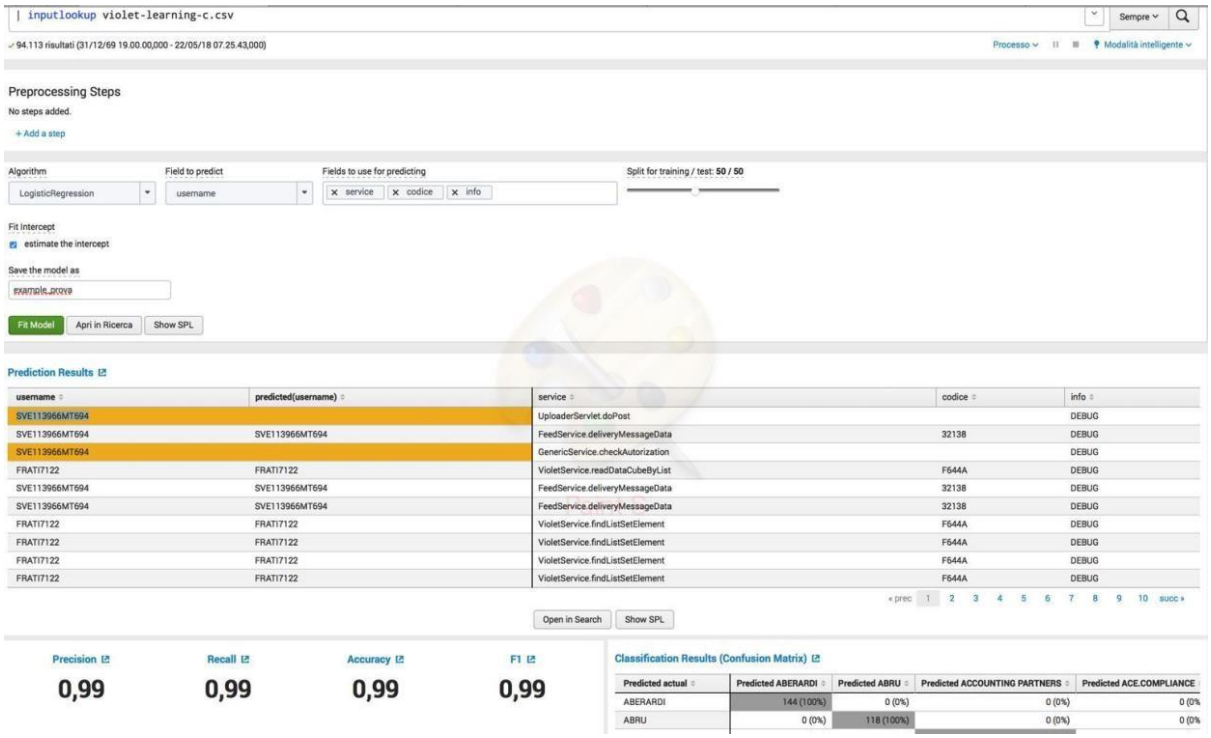


Figure 32: Output of the prediction model

The suggestion for further investigations

The approach outlined here is to be understood as a starting point for subsequent investigations and certainly not a deterministic and exhaustive approach. For this reason, it is a useful tool in the early stages of forensic investigations related to logs, when you do not yet know where to direct attention or when, even after analyzing everything, you still want further evidence that you have not missed anything. What must be said is that predicting categorical fields is a type of learning known as classification. This type of algorithm learns the tendency of data to belong to one category or another, based on the related data. Therefore what emerges from the visualization of the above tables is the actual state of the username field, versus the expected state of the same field. It should be noticed, in yellow, the incorrect prediction from the model, and therefore the anomaly. Although it has not been treated in the elaboration here reproduced, a further possible application of this technique, could be the forecast of the application load, according to other applicative variables. This experiment, which presupposes a complex preparation of the data, would allow the provisioning of the resources in cases of high load forecasting, and therefore would have repercussions on the management costs.

Categorical Outlier Indentification

In this paragraph the experimentation related to the identification of the categorical outliers (activity 4.) is reported. This activity it is not to be considered a modeling of data. The input file is, as in the previous task, a CSV data source, and it must be cleaned through a process analogous to the previous section by making the data as structured as possible, made by eliminating unstructured information that occurs within the logs. After using the SPL command 'anomalydetection' with which the outliers could be determined.

"| inputlookup indexB-learning-c.csv | anomalydetection "service" "timestamp" "username" action=annotate | eval isOutlier = if(probable_cause ¯"", "1", "0") | table "service" "timestamp" "username", probable_cause, isOutlier | sort 100000 probable_cause"

This command is not really a 'Machine Learning' command but applies probabilistic techniques to events by identifying anomalous events and calculating the probability of each event, and then identifying the smallest probabilities. Probability is the result of the frequencies of each value in the individual event fields. In the process of identifying anomalies of the data

in the logs, we can proceed with a ’univariate’ or with a ’multivariate’ approach. That is; in order to identify anomalies, we can determine the individual anomalies for each field, taken individually; otherwise we can execute the command to identify the anomalies by specifying at the same time several fields in relation to which we want to determine the anomalies. It is obvious that the results obtained with the ‘univariate’ approach do not coincide with the ‘multivariate’ one, because the method of calculating the probability is different. In the former case it is a probability of a simple event, in the latter case it is a compound probability. The determination of the outlier values can be used as the trigger of an alert or a subsequent action. Below is reported the experimentation on the determination of the categorical outliers (anomalies) made on the single field ’service’ (Figure 33):

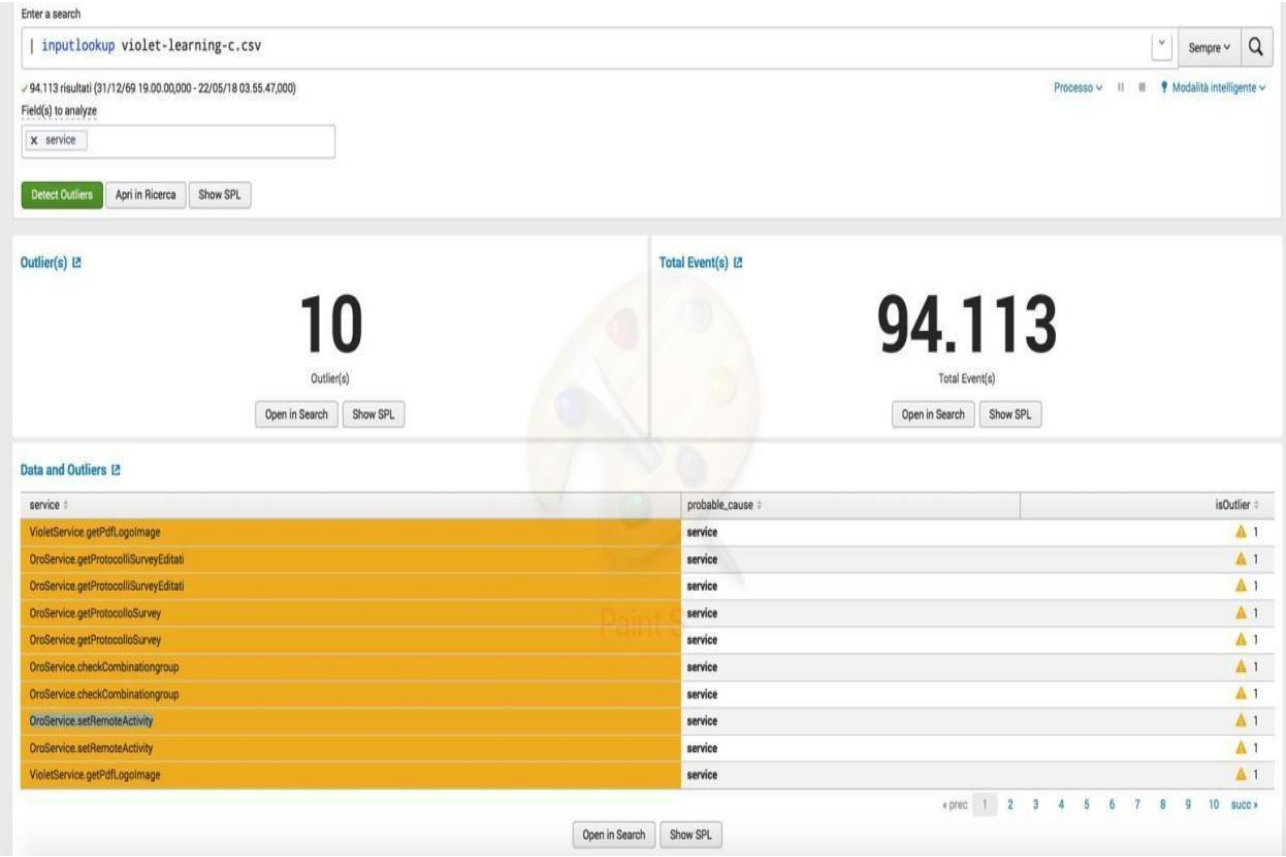


Figure 33: Unique categorical output on ‘service’ field

In the following Figure 34, it is reported the determination of the outliers related to the single field ‘username’:

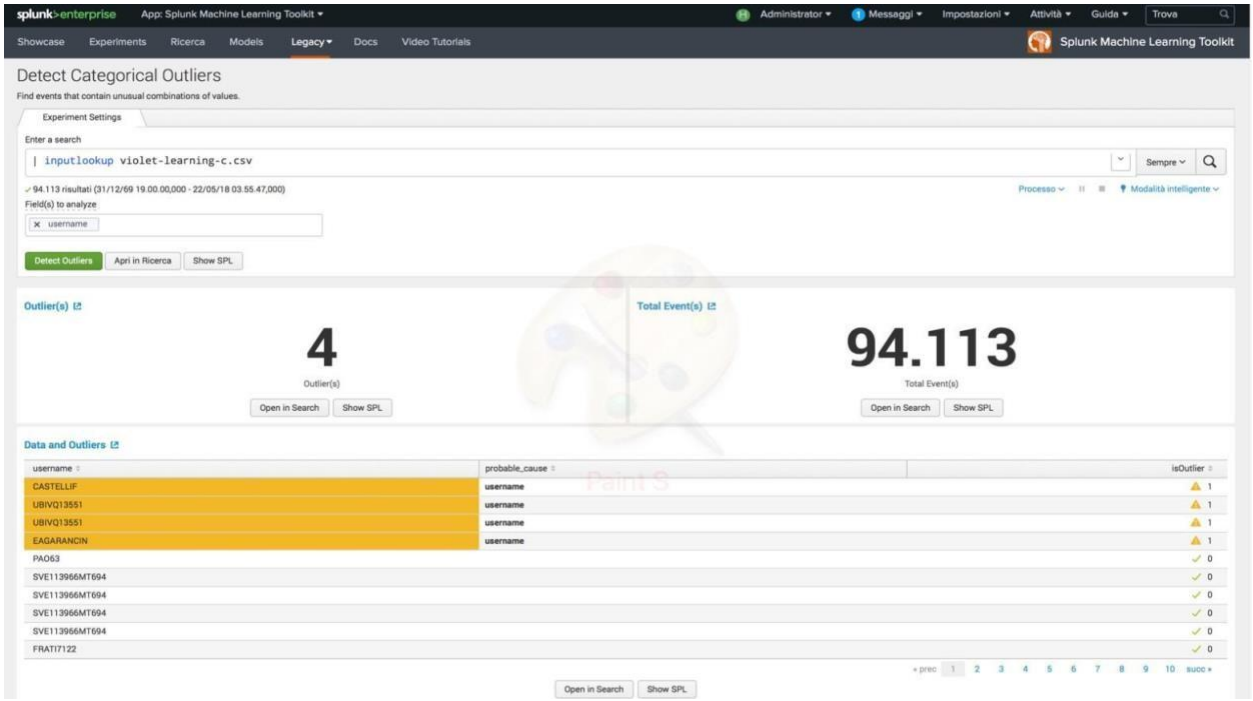


Figure 34: Unique categorical output on field ‘username’ field

In the following Figure 35, the reader can observe that, if they are added multiple fields, i.e. ‘username’, ‘service’ and ‘timestamp’, the result obtained as output table is a different one, thus indicates different outliers from the previous ones.

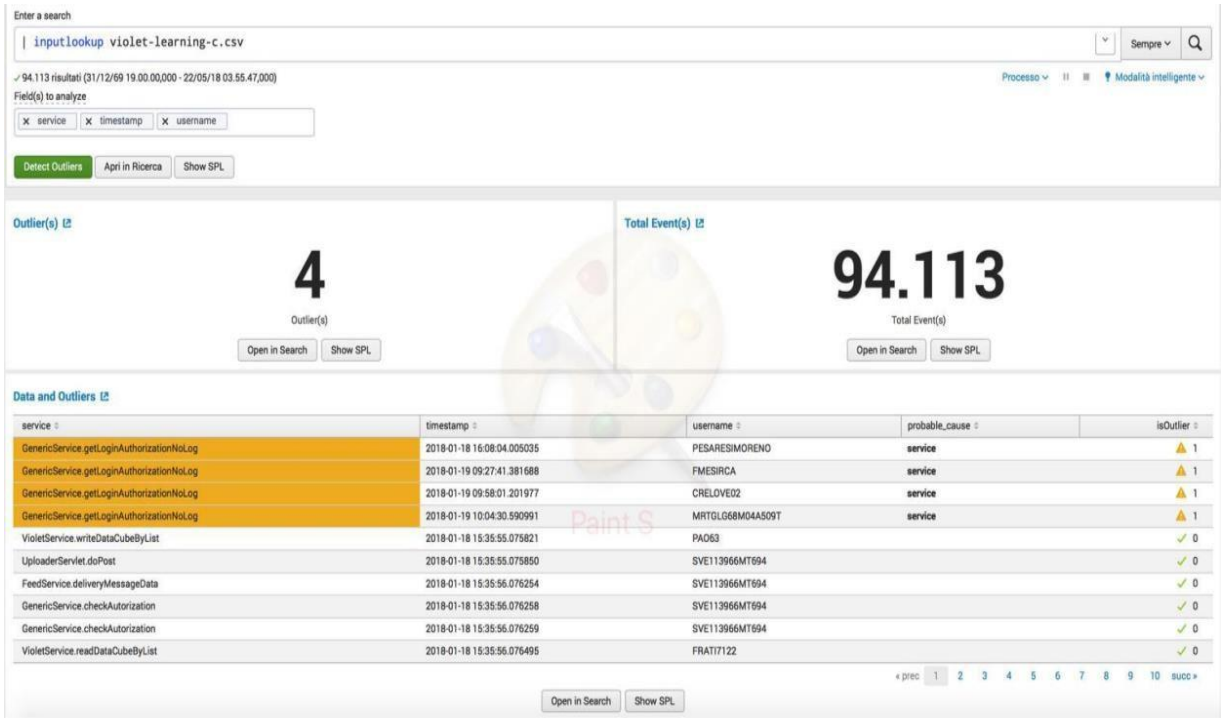


Figure 35: Multivariate categorical output

Note that the results showing the outliers are accompanied by visual information on their number, on the overall events, on the reason why those events are considered outliers. This report can be a starting point to investigate furtherly the anomalies, and to verify if there are any fraudulent behaviors behind them. In the image highlighted in yellow the data indicating unusual events.

12. CONCLUSION

My work begun with the aim of enhancing the application log files, thus making them company assets. An asset implies adequate protection mechanisms, professional skills that deal with it, adequate training, standards of writing and compliance. In this work can be assessed how much value can be extracted at an informative level from the analysis of the logs, exclusively considering those logs of applicative nature. Backwards, the work retraced the reverse path of the synthesis of the data analyzed, to make them visible and usable from the perspective of Business Analytics. It was observed if it is more useful in the process of log analysis to “look at the tree or the forest”, and thanks to a tool as powerful as versatile and scalable as Splunk it was concluded that both could be done. Forensic analysis is no longer an investigative activity at Sherlock Holmes way of doing, that is, using a magnifying glass to go deeper and deeper into the details that can trace the anomalous event or fraudulent compromise, but it must be considered an overview. It also has been examined the performance aspects of the architecture, capacity planning and dimensioning from an enterprise point of view, since the future lies in Big Data, that is, in the processing of large amounts of data, in real time, and with data variable formats and structures. Finally, it was seen how intelligence can be added to the discovery process thanks to Machine Learning and to the probabilistic approach. The next steps will consist in increasing the ability to correlate logs of different origins, aggregating events that originate from the same action, reasoning on time, synchronizing events that may present apparent misalignments, and involving multiple devices. This aspect is all the more important since APT (Advanced Persistence Threats) attacks are going to spread. These attacks are characterized by being prolonged, persistent, and wide-ranging on multiple targets of the information system. A limit of the present work at the moment is that the analysis is based only on the information contained in the logs and not on all other sources such as online traffic data, scripts execution, API calls, commands, which would provide a certain vision even more complete, but at the moment outside the range of action of the presented project.

13. REFERENCES

1. Towards structured log analysis, by Dileepa Jayathilake, from International Joint Conference on Computer Science and Software Engineering (JCSSE), Publication Year: 2012, Page(s): 259 – 264
2. A. Krishna, Splunk Admin & Architect: Complete Tutorials + 30 Days Lab, Udemy, online.
3. Exploring Splunk search processing language (SPL) primer and cookbook, by David Carasso, Splunk's Chief Mind, CITO Research New York, NY, 2012
4. R. Meyer, Detecting Attacks on Web Applications from Log Files, SANS Institute, 2008, Roger Meyer
5. Splunk and the SANS Top 20 Critical Security Controls, Mapping Splunk Software to the SANS Top 20 CSC Version 4.1, 2014 by Splunk Inc.
6. Splunk Inc, Splunk Machine Learning Toolkit User Guide 3.1.0. 2018

14. GLOSSARY

I. Core Splunk Concepts:

- **Splunk:** A software platform for searching, monitoring, and analyzing machine-generated big data, through a web-style interface.
- **Indexer:** The Splunk component that processes and stores machine data. It parses the data, creates indexes, and stores it in time-based buckets.
- **Search Head:** The Splunk component that provides the user interface for searching and analysing data. It sends search queries to the indexers and presents the results.
- **Forwarder:** A lightweight agent that collects data from various sources and sends it to the indexers.
    - **Universal Forwarder:** A lightweight forwarder that sends data with minimal processing.
    - **Heavy Forwarder:** A forwarder that can parse and route data before sending it to the indexers.
- **Search Processing Language (SPL):** The search language used in Splunk to retrieve, manipulate, and analyse data.
- **Events:** Individual pieces of data, typically a line of text, that Splunk indexes.
- **Fields:** Key-value pairs extracted from events, used to filter and analyze data.
- **Index:** A repository of indexed data in Splunk.
- **Source type:** A field that identifies the data format of an event.
- **Source:** The origin of the data (e.g., a file, a network port).
- **App:** A pre-built collection of dashboards, reports, and configurations for specific use cases.
- **Add-on:** A smaller, focused package that provides specific functionality, often for data ingestion or field extraction.
- **Data Model:** A hierarchical representation of data that simplifies searching and reporting.
- **Knowledge Object:** Saved searches, reports, dashboards, alerts, and other configurations that enhance Splunk's functionality.

II. Real-Time Monitoring Terms:

- **Real-Time Search:** A search that continuously runs and displays results as new data arrives.

- **Continuous Monitoring:** The ongoing observation and analysis of data to detect anomalies and trends.
- **Metrics:** Numerical measurements of system performance or application behavior.
- **Logs:** Textual records of events and activities generated by systems and applications.
- **Dashboards:** Visual representations of data, typically using charts and graphs.
- **Panels:** Individual visualizations within a dashboard.
- **Key Performance Indicators (KPIs):** Critical metrics used to evaluate the performance of a system or application.
- **Anomaly Detection:** The process of identifying unusual patterns or deviations from expected behavior.
- **Threshold**: A predefined value used to trigger an alert when a metric exceeds or falls below it.
- **Latency:** The delay between an event occurring and its detection.
- **Throughput:** The amount of data processed within a given time.

III. Alerting Terms:

- **Alert:** A notification triggered by a specific event or condition.
- **Trigger Condition:** The criteria that must be met for an alert to be triggered.
- **Alert Action:** The action taken when an alert is triggered (e.g., sending an email, running a script).
- **Scheduled Alert:** An alert that runs at predefined intervals.
- **Real-time Alert:** An alert that triggers immediately when a condition is met.
- **Correlation Search:** A search that identifies patterns or relationships between events.
- **Suppression:** Preventing duplicate alerts from being triggered.
- **Escalation:** The process of notifying higher-level personnel if an alert is not acknowledged.
- **Webhook:** An automated message sent from an application when something happens. Used to integrate splunk alerts with other systems.
- **Saved Search:** A search query that has been saved for reuse, often used as the base of an alert.
- **Notable Event:** An event that is marked as significant or requiring attention, often generated by an alert.

IV. Splunk Specific Alerting terms:

- **Triggered Alerts:** A list of all alerts that have been activated.
- **Alert Actions (Splunk):** Actions that Splunk can perform when an alert is triggered, such as sending emails, running scripts, or triggering webhooks.
- **Alert Manager:** A Splunk tool for managing and configuring alerts.
- **Summary Index:** A special index that stores aggregated data, often used for performance monitoring and alerting.
- **Adaptive Thresholding:** Splunk techniques that automatically adjust alert thresholds based on historical data.

15. APPENDIX

Appendix A: Sample SPL Queries

A.1 Error Rate Alert Query

SPL:

```
{
        index=web_logs status=500
        | stats count by host
        | where count > 10
}
```

Triggers an alert if more than 10 HTTP 500 errors are detected on any host.

A.2 CPU Usage Monitoring (from metrics index)

SPL:

```
{
        index=metrics source="cpu_usage"
        | timechart avg(usage_percent) by host
}
```

Displays the average CPU usage per host over time.

A.3 Failed Login Attempts (Security Use Case)

SPL:

```
{
        index=auth_logs action="failed_login"  | stats count by user, src_ip
        | where count > 5
}
```

Detects brute-force login attempts by counting failed logins per IP.

A.4 Anomaly Detection Using Z-Score

SPL:

```
{
        index=network_logs  | timechart span=5m count
        | eventstats avg(count) as mean stdev(count) as std_dev
        | eval z_score=(count - mean) / std_dev
        | where abs(z_score) > 2
}
```

Flags outlier events based on statistical deviation from the mean.


ISSN: 2321-2152
IJMECE
International Journal of modern
electronics and communication engineering
E-Mail
editor.ijmece@gmail.com
editor@ijmece.com
www.ijmece.com

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 1, 2025

REAL TIME MONITORING AND ALERTING USING SPLUNK

N.Madhu Bhavani[1], Shivva Aishwarya[2], Adepu Shashikumar[3], Gandla Raju[4], Katroth Laxman[5]

[1] Associate Professor, Dept. of CS, Sri Indu College of Engineering and Technology, Hyderabad,
[2 3 4] Research Student, Dept. of CS Sri Indu College of Engineering and Technology, Hyderabad

Abstract: Cybersecurity is a critical concern for organizations as cyber threats grow increasingly frequent and sophisticated. Real-time detection and response to these threats are essential for safeguarding data and maintaining operational continuity. Splunk Enterprise Security (ES), a robust Security Information and Event Management (SIEM) platform, offers advanced tools for identifying and mitigating cyber threats. This paper explores the possibilities of using Splunk ES to enhance advanced cyber threat detection, focusing on its features, capabilities, and real-world applications. Splunk ES collects, indexes, and analyzes extensive machine data from diverse sources, including system logs, network traffic, and security devices. With real-time monitoring and comprehensive visibility into an organization's IT ecosystem, Splunk ES enables early detection of suspicious activities. It offers pre-configured security content, such as correlation searches, dashboards, and reports, to streamline threat identification and incident response. A notable strength of Splunk ES lies in its flexibility, allowing users to customize detection rules and dashboards to meet specific organizational needs. The platform's adaptive response features support automated actions based on predefined criteria, significantly reducing the time from threat detection to mitigation. Furthermore, the integration of machine learning enhances its ability to detect patterns and anomalies, including those that might bypass traditional signature-based detection methods. In practice, Splunk ES has demonstrated its efficacy in addressing diverse cyber threats, including advanced persistent threats (APTs), insider threats, and zero-day vulnerabilities. By offering scalable and powerful tools, Splunk ES enables organizations to detect, analyze, and respond to security risks efficiently, paving the way for more robust cybersecurity strategies. This study examines the potential of Splunk ES as a vital asset in the fight against advanced cyber threats.

Keyword(s): Cybersecurity, Cyber threat detection, Splunk, Splunk Enterprise Security (ES), SIEM (security information and event management), Real-Time monitoring

1. Introduction

The rapid evolution of cyber threats has made real-time detection and response essential for maintaining organizational security and operational continuity. Security Information and Event Management (SIEM) systems address this need by combining *Security Information Management (SIM)* for log management and compliance reporting with *Security Event Management (SEM)* for real-time monitoring and event correlation. SIEM systems collect, normalize, and analyze log data from diverse sources, such as endpoint devices, servers, firewalls, and intrusion detection and prevention systems (IDPS), to provide a centralized view of system activity. According to the Information Technology Infrastructure Library (ITIL), an event is defined as any significant occurrence within an IT environment that requires attention or action [9]. SIEM systems rely on event correlation to analyze log data in near real-time and identify patterns or anomalies indicative of potential attacks [3][7][27].

Despite their effectiveness, traditional SIEM systems face challenges such as performance bottlenecks and query prioritization issues when handling large volumes of raw log data [4]. Modern platforms like Splunk Enterprise Security (ES) overcome these limitations by integrating advanced tools for real-time threat detection, response, and analytics. Splunk ES collects and indexes machine data from multiple sources, offering comprehensive visibility into IT environments. Its pre-configured security content, customizable detection rules, and machine learning capabilities enable organizations to proactively identify and mitigate sophisticated threats, including advanced persistent threats (APTs), insider threats, and zero-day vulnerabilities.

This paper explores how Splunk ES enhances cybersecurity strategies by leveraging its advanced features. The discussion begins with an overview of SIEM systems, their evolution, and their dual SIM and SEM functionalities. It then delves into the architecture and key capabilities of Splunk ES, including its data collection, indexing, and adaptive response mechanisms. Additionally, the role of machine learning in anomaly detection is examined, focusing on its ability to uncover patterns that traditional methods may overlook. Finally, the paper presents a design for implementing Splunk ES, demonstrating its practical applications and effectiveness in mitigating diverse cyber threats.

2. Literature Review

Security Information and Event Management (SIEM) systems have evolved into indispensable tools for modern cybersecurity practices, combining log management, threat detection, and incident response. This literature review critically examines foundational and recent advancements in SIEM research, highlighting their contributions, limitations, and implications describing SIEM capabilities with Splunk Enterprise Security (ES).

Bryant Blake [1] introduced a novel threat ontology model to improve the detection and response capabilities of SIEM systems. His framework, which integrates a kill-chain model into SIEM processes, enhanced the contextual understanding of security incidents by normalizing log data and linking events across multiple stages of an attack lifecycle. This work was pivotal in reducing false positives, a persistent challenge in SIEM systems. However, while Blake's model demonstrated improved detection accuracy, its reliance on predefined ontologies left room for exploration in dynamic and adaptive threat detection, which forms a key focus of more recent studies.

Privacy and compliance have become critical considerations in SIEM deployments, as explored by Menges et al. [2]. Their research addressed the integration of General Data Protection Regulation (GDPR) principles into modern SIEM systems. By incorporating pseudonymization and granular access controls, the authors demonstrated a framework that ensures both robust security monitoring and compliance with privacy laws. This dual focus on security and compliance underscores a growing trend in SIEM research, where tools must cater to regulatory requirements without compromising operational efficiency. This study's emphasis on Splunk ES reflects a similar consideration, as Splunk provides extensive tools for regulatory reporting and data protection.

The historical progression of SIEM systems was analyzed by Inns [3], who traced their evolution from basic log management tools to sophisticated platforms incorporating real-time analytics and automated response. Inns highlighted early SIEM systems' inability to correlate multi-source events effectively, a limitation that spurred the development of advanced correlation engines. His work serves as a backdrop for understanding the advancements in modern systems like Splunk ES, which address these limitations through machine learning and customizable correlation rules.

Expanding the scope of SIEM functionality, Detken et al. [5] explored the integration of Network Access Control (NAC) with SIEM systems, demonstrating the potential for combining access management with real-time monitoring. By leveraging open-source SIEM tools, their study provided a cost-effective solution for small and medium-sized enterprises (SMEs). Although open-source tools offer flexibility, the study acknowledged their limitations in scalability and pre-built functionality, which commercial systems like Splunk ES effectively address through robust analytics and enterprise-grade scalability.

Bezas et al. [6] conducted a comparative analysis of open-source and commercial SIEM systems, identifying critical differences in performance, customization, and cost. Their findings highlighted Splunk's superiority in handling large-scale data, offering intuitive dashboards, and supporting diverse use cases. However, they also pointed to challenges such as Splunk's high cost and steep learning curve, raising questions about balancing enterprise-level functionality with accessibility, an area this study explores further by examining Splunk's usability enhancements and training resources.

Hristov et al. [8] and Kamal et al. [10] investigated the application of Splunk ES in detecting complex threat vectors, including Distributed Denial of Service (DDoS) attacks and anomalies in IoT environments. Hristov's work emphasized the importance of real-time data visualization and advanced alerting mechanisms in mitigating DDoS attacks, while Kamal's study demonstrated Splunk's adaptability in academic network monitoring. Together, these studies highlight the versatility of Splunk ES in addressing diverse cybersecurity challenges, emphasizing its role in this study as a tool for enhancing threat detection in multi-vector environments.

Bruzzese [14] extended Splunk's application by developing a custom app for synthesizing and analyzing application logs. His work illustrated the potential for tailoring SIEM platforms to meet specific organizational needs, a flexibility that underscores the adaptability of commercial systems like Splunk. Similarly, Raja and Vasudevan [9] tackled specific attack patterns by proposing rule-generation techniques for TCP SYN flood detection, highlighting the critical role of customizable correlation rules; a feature extensively leveraged in Splunk ES.

Sornalakshmi [17] explored zero-day threat detection using SIEM systems, demonstrating the effectiveness of continuous monitoring and heuristic-based alerting. This study, alongside that of Di Mauro and Di Sarno [7], who addressed encrypted traffic detection, underscores the need for innovative approaches to emerging threats.

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 1, 2025

These contributions align closely with the adaptive response mechanisms in Splunk ES, which enhance detection and response capabilities for sophisticated attack patterns.

Despite the significant advancements in SIEM systems, challenges remain. Bhatt et al. [15] identified issues such as data overload and the complexity of rule creation, which hinder effective decision-making. These limitations emphasize the importance of user-friendly interfaces and automated tools, areas where Splunk ES excels with its machine learning toolkit and pre-built content for common use cases.

These studies collectively highlight the evolution of SIEM systems and their trajectory toward greater intelligence, automation, and adaptability. However, they also reveal persistent gaps, such as the need for enhanced automation, privacy-preserving mechanisms, and advanced detection capabilities. This study builds on these insights by exploring how Splunk Enterprise Security addresses these challenges, particularly through its pre-built analytics, machine learning capabilities, and integration with diverse threat detection frameworks. By evaluating Splunk ES in the context of these advancements, this paper contributes a comprehensive analysis of its role in advancing the SIEM landscape.

3. Security Information and Event Management (SIEM) Using Splunk Enterprise Security (ES)

3.1 Foundational Principles of SIEM Systems

Security Information and Event Management (SIEM) systems form the backbone of modern cybersecurity operations, integrating various processes to detect, analyze, and respond to potential threats. Key functions of SIEM include:

- **Log Collection:** SIEM systems gather logs from a wide variety of monitoring tools and devices, including firewalls, servers, network equipment, and applications. This comprehensive log collection forms the foundation for effective threat detection [15].
- **Log Aggregation:** Once collected, logs are aggregated. This process involves grouping similar log entries to reduce redundancies and avoid overwhelming the security team with excessive data [2].
- **Log Standardization:** Since logs come from various devices and tools with different formats, standardization normalizes this data. This ensures all log data adheres to a common structure, making it easier to analyze and correlate [19].
- **Event Correlation:** Event correlation is the process of connecting different logs and events that may appear unrelated at first glance but are indicative of a potential security threat when combined. This step is crucial for identifying patterns of suspicious behavior and is often powered by rules, machine learning, or artificial intelligence [3].
- **Log Storage:** SIEM systems ensure that logs are stored securely for future reference and for compliance with regulations. Effective log storage solutions protect against tampering, ensuring the integrity of logs for audits, forensic analysis, and long-term retention [19].

Figure 1: Splunk Enterprise SIEM Overview [16]

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 1, 2025

Splunk Enterprise SIEM platform (Figure 1) is a powerful and versatile platform designed to provide advanced capabilities for log and event management. It delivers seamless log collection, normalization, and correlation, offering users a centralized and cohesive view of their critical IT services. Powered by machine learning, Splunk's analytics capabilities help detect anomalies, uncover root causes, and assess the impact of issues with remarkable efficiency [8].

This research chooses Splunk Enterprise SIEM as the preferred platform due to its numerous advantages. Its instant trial and seamless transition from proof of concept to production make it highly accessible and practical for testing and deployment [14]. Additionally, the platform ensures reliability by offering a dedicated and secure environment for each customer. One of its standout features is the ability to configure custom alert triggers, which enable users to monitor data in real time and identify potential anomalies swiftly [19].

Splunk Enterprise also excels in providing a single, unified view of all machine-generated data, streamlining data management and analysis [10]. This centralized approach not only simplifies monitoring but also enhances the flexibility and adaptability required for businesses to operate at their desired pace. These features make Splunk a reliable and effective choice for experimental and operational purposes, aligning with the dynamic needs of enterprises [16].

The ability of SIEM systems to centralize data and provide actionable insights is critical for maintaining security and operational continuity [2].

3.2 Splunk ES Architecture and Data Management

Splunk Enterprise Security (ES) is built on a robust and scalable architecture (Figure 2), designed to efficiently process, analyze, and manage vast amounts of machine data generated by IT infrastructure and applications [24]. This architecture supports real-time security monitoring and comprehensive data analysis, enabling organizations to proactively detect and respond to threats. Below is a detailed explanation of the key components in the Splunk ES architecture:

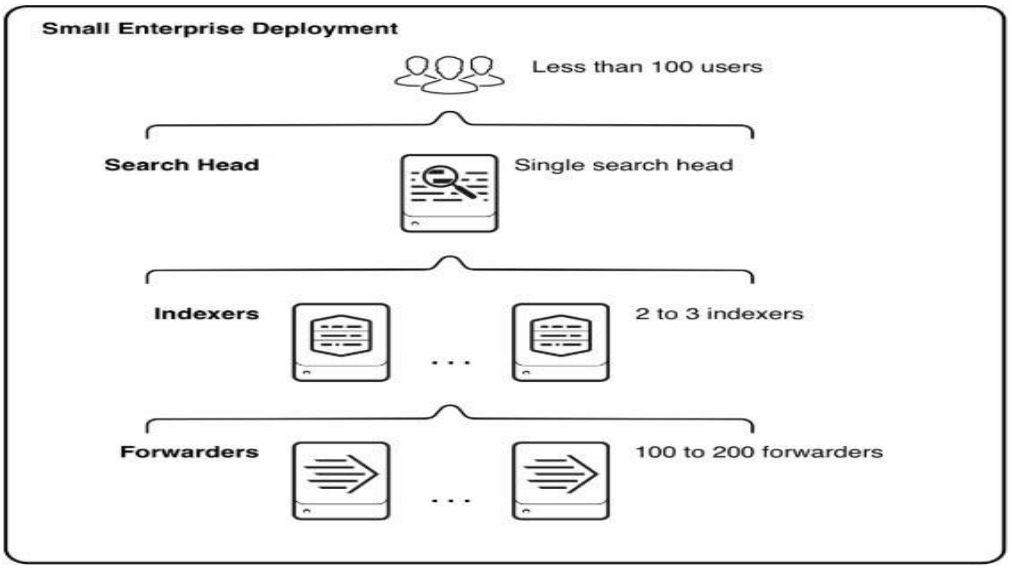


Figure 2: Small Enterprise Deployment [31]

Splunk Forwarders: Splunk Forwarders are lightweight software agents installed on endpoint devices, such as servers, workstations, and network devices, to collect log data. These agents are responsible for:

- **Log Collection:** Capturing logs, metrics, and other machine-generated data from diverse sources, including system logs, application logs, and network traffic [21].
- **Data Preprocessing:** Filtering unnecessary data and performing basic transformations to reduce the volume of data transmitted [2].
- **Data Transmission:** Securely forwarding the pre-processed logs to Indexers in real time using protocols such as TCP or HTTP [21].
- Splunk Forwarders ensure that data collection is decentralized and efficient, reducing the load on central servers [8].

Splunk Indexers: Indexers are the backbone of Splunk's architecture, responsible for:

- **Data Ingestion:** Receiving raw data from Splunk Forwarders and preparing it for indexing [20].
- **Data Parsing and Indexing:** Splitting raw data into searchable events and creating an index for fast retrieval [20].
- **Storage Management:** Storing indexed data in a highly optimized format to ensure quick access while maintaining scalability [14].
- **Search Optimization:** Enabling fast query execution by creating metadata and leveraging indexing algorithms [5].
- The distributed nature of Indexers allows Splunk to scale horizontally, accommodating growing data volumes seamlessly [20].

Splunk Search Head: The Search Head serves as the primary user interface for analysts, providing tools for querying, visualizing, and managing data. Its functionalities include:

- **Search Execution:** Allowing users to perform ad-hoc or scheduled searches on indexed data using Splunk's Search Processing Language (SPL) [23].
- **Data Visualization:** Generating dashboards, charts, and graphs to represent trends and anomalies in a user-friendly format [10].
- **Collaboration and Reporting:** Enabling teams to share insights and generate comprehensive reports for stakeholders [22].
- The Search Head ensures that complex data queries are executed efficiently, providing actionable insights in real-time [6].

Splunk Enterprise Security Application: The Splunk Enterprise Security (ES) Application is a specialized module designed specifically for cybersecurity use cases. It includes:

- **Security-Specific Dashboards:** Prebuilt and customizable dashboards to monitor key security metrics and threat landscapes [25].
- **Correlation Rules**: Advanced rule-based logic to correlate multiple events and identify potential security incidents or anomalies [9].
- **Incident Review and Investigation**: Tools for triaging, investigating, and responding to detected threats [16].
- **Pre-Built Content**: Including use case templates, correlation searches, and security-specific workflows to accelerate deployment and efficiency [18].
- By leveraging the capabilities of the ES application, organizations can enhance threat detection and incident response, tailoring solutions to their specific security needs [17].

4. Core Features of Splunk ES

Splunk Enterprise Security (ES) is an advanced platform designed to enhance an organization's cybersecurity posture. By offering tools for rapid detection, analysis, and mitigation of cyber threats, Splunk ES plays a critical role in modern IT security operations.

One of the core strengths of Splunk ES is its **pre-built security content**, which includes a library of resources that address common cybersecurity scenarios. This content features **correlation searches**, which are pre-configured queries that help identify patterns of behavior, such as brute-force login attempts or lateral movements within a network. These searches simplify the process of detecting potential threats. Additionally, **dashboards** in Splunk ES provide visual representations of security data, enabling users to monitor real-time threat status and trends over time. Reports are automatically generated for compliance purposes or to share insights with stakeholders, streamlining the security monitoring process [1] [2].

In terms of **real-time monitoring**, Splunk ES offers comprehensive visibility into an organization's IT infrastructure. The platform continuously aggregates and normalizes data from diverse sources, such as firewalls, endpoint devices, and cloud environments. This ensures a unified view of all operations, which is essential for early threat detection. Real-time alerts immediately notify analysts of suspicious activities, allowing for a swift response to mitigate potential risks [19].

Another powerful feature of Splunk ES is its **customization options**. Users can tailor security rules, dashboards, and alerts to meet the specific needs of their organization. Analysts have the flexibility to define custom correlation rules to detect unique threats that are relevant to their environment. Furthermore, custom

dashboards can be designed for specific teams or use cases, providing tailored insights and metrics. Alerts can also be configured to trigger notifications for critical events, ensuring the right teams are informed in real time [14] [19].

Finally, **adaptive response** in Splunk ES automates predefined actions in response to detected threats. This feature is particularly useful for reducing the time between threat detection and mitigation. Automated actions such as blocking an IP address, isolating a compromised endpoint, or generating service tickets are available. Additionally, Splunk ES integrates seamlessly with incident response systems to streamline workflows, reducing the need for manual interventions and improving response efficiency [5] [19].

5. Machine Learning Workflow in Splunk ES

The machine learning process within Splunk ES is streamlined into distinct phases, each contributing to its ability to detect and analyze threats effectively.

In the **raw data collection** phase, Splunk ES collects vast amounts of machine data, including system logs, network traffic, and application metrics, from multiple endpoints and sources. This diverse collection of data forms the foundation of Splunk ES's analytical capabilities and ensures that it has access to a comprehensive view of an organization's network and systems [26].

The next phase, **feature extraction**, involves identifying and extracting relevant features from the raw data that can aid in anomaly detection. These features highlight critical attributes such as login frequency, data transfer rates, and geographical access locations, which are indicative of normal or anomalous behavior. This step is crucial for narrowing down the large volumes of raw data into manageable, meaningful patterns [1].

Once the features have been extracted, the next step is **model training**. During this phase, machine learning models are trained using historical datasets to learn typical patterns of behavior and establish a baseline for normal activities. Both supervised and unsupervised learning techniques are applied depending on the type of threat being addressed, with supervised learning leveraging labeled datasets to predict specific outcomes, while unsupervised learning focuses on identifying patterns or anomalies without prior labelling [2]. This dual approach enhances the model's ability to detect known and unknown threats.

After the model has been trained, the **prediction** phase begins. The trained models continuously analyze incoming data in real-time, comparing it to the baseline established during training. Any anomalies or deviations from expected behavior are flagged for further investigation, ensuring that potential threats are detected promptly. This real-time analysis enables immediate responses to unusual activities, such as unauthorized access attempts or suspicious network flows [4].

In the **anomaly detection** phase, detected anomalies are categorized based on their severity. This classification helps security teams prioritize their response efforts, addressing the most critical threats first. For example, unauthorized login attempts from unusual locations or sudden spikes in network traffic can be flagged as high-severity anomalies requiring immediate attention, while less critical anomalies may be flagged for later review. This tiered approach helps optimize the incident response process [16].

Benefits of Machine Learning in Splunk ES

- Proactive Threat Identification: ML algorithms enable Splunk ES to detect threats before they escalate, minimizing potential damage [14].
- Reduced False Positives: By continuously learning and refining its models, Splunk ES reduces the number of false positives, ensuring analysts focus on genuine threats.
- Enhanced Scalability: Machine learning models can process and analyze large-scale datasets, making Splunk ES suitable for enterprises with extensive IT infrastructures [8] [9].

6. Splunk Enterprise Security Application Implementation Design:

We are ensuring high availability by creating distributed Splunk cluster architecture (as shown in Figure 3) which has Splunk servers in 2 different zones. The following are the major components that are installed and configured as part of the Splunk Security Information and Event Management (SIEM) solution:

- Indexers: An indexer is the Splunk Enterprise instance that indexes data. The indexer processes raw data by converting it into structured events, which are then systematically stored within an index for efficient retrieval and analysis. The indexer also searches the indexed data in response to the search requests.

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 1, 2025

- Search head: In a distributed search environment, the search head is the Splunk Enterprise instance that handles search management functions, directing search requests to a set of search peers and subsequently consolidates the results to present them back to the user. If this instance does only search and not indexing, it is usually referred to as a dedicated search head. It is the major front-end, generally assessed through Splunk web interface.
- Deployment Server: It manages configuration, apps and content updates across all Splunk components. Deployment server communicates with Heavy Forwarder and Universal Forwarder over port 8089.
- License Master: A License Master in Splunk is a specialized instance responsible for managing and controlling multiple license slaves. It oversees tasks such as defining license stacks, configuring pools, adding licensing capacity, and efficiently managing connected license slaves.
- Universal Forwarders: A lightweight Splunk Enterprise instance that obtains and streams data to the indexers. Currently all the Windows and UNIX servers are installed with Universal Forwarders.
- Heavy Forwarder: HF can be used to decrease parsing load to Indexers and work as log aggregators, also some of the add-ons required for parsing are better suitable on HF.
- Cluster Master Node: Regulate the functioning of an index cluster; controls and manages index replication.
- Syslog Server: Syslog provides a standardized method for network devices to transmit event messages to a centralized logging server, commonly referred to as a Syslog server. Widely supported by various devices, the Syslog protocol enables logging of diverse event types. Many networking components, including routers and switches, can generate and sending Syslog messages.
- Splunk Apps: They are pre-built collections of dashboards, panels and UI elements that are driven by pre-configured searches and packaged for a specific technology or use case.
- Splunk Web: It is an interactive graphical user interface in Splunk that remote users can connect to over a web browser for administration, problem investigation and reporting on results.

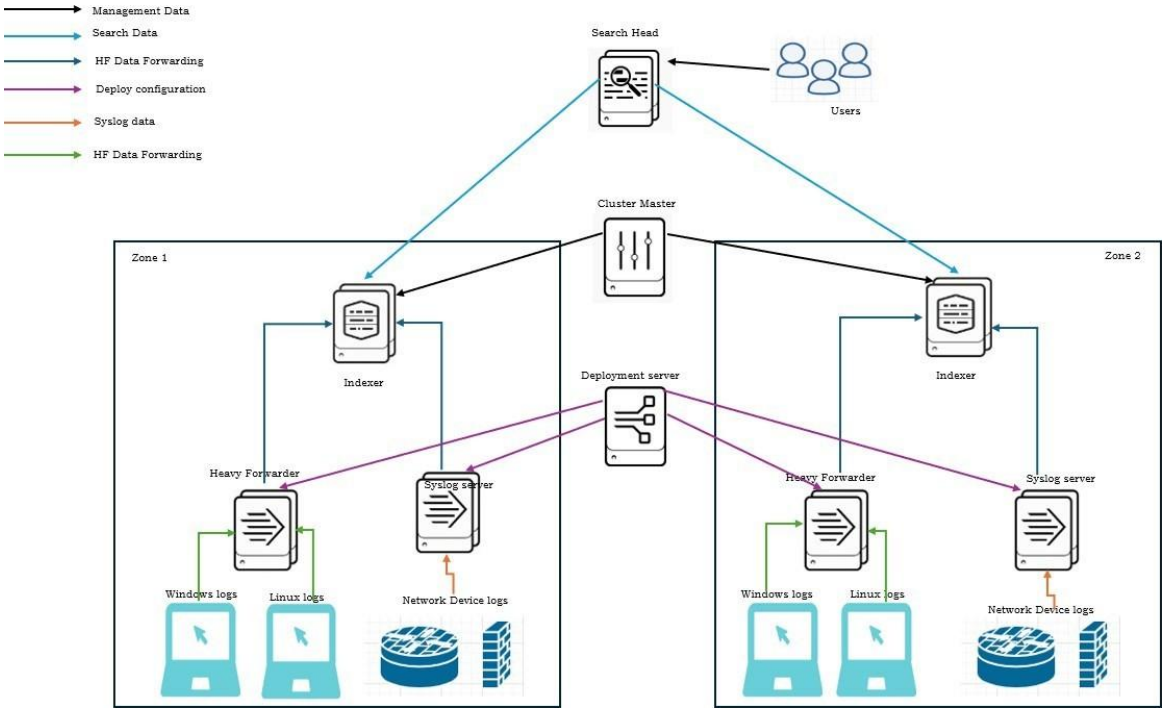


Figure 3: Splunk ES Application Architecture Design

The Universal Forwarders will be installed on Windows and Linux machines and these logs will be forwarded to the Heavy Forwarder where we can filter logs and send it further to the Indexer for data parsing and indexing. Network device logs are sent to the Syslog server which has a Splunk Universal Forwarder installed on it already, will further send these logs to the Splunk Indexer within the same zone. The Heavy Forwarder and Syslog server in both the zones is managed by the Deployment server and the Cluster Master manages the Indexers in the two zones. The Search head is used by the SOC analyst to view and analyse event data, thus working on the cyber threats detected. This Search head can view and searches data from both the indexers in our Splunk architecture

and has the Splunk ES application installed on it. In our Splunk architecture, we are considering Cluster Master to act also as the License Master (hence not shown in diagram).

Real-World Applications and Case Studies

Splunk ES has been successfully deployed across various industries to mitigate cyber threats. Some Real-world examples include:

- • Advanced Persistent Threats (APTs): Detecting and responding to prolonged, targeted attacks.
- • Insider Threats: Identifying malicious or accidental activities by employees.
- • Zero-Day Attacks: Spotting unknown vulnerabilities through anomaly detection.

7. Conclusions and Future Research

Splunk Enterprise Security has proven to be a valuable tool in the fight against modern cybersecurity challenges. Its ability to integrate advanced analytics with traditional SIEM functionalities offers organizations a scalable and effective solution for threat management. The platform's centralized data management, real-time alerting, and customizable dashboards help streamline security operations, improve threat detection, and enable rapid response. Furthermore, Splunk ES is well-suited to meet compliance requirements, making it an essential component of a robust enterprise cybersecurity strategy [20].

In this study, Splunk Enterprise Security seemed to promise its capability to provide insights that enable timely identification and mitigation of threats. The use of machine learning and adaptive response mechanisms can enhance the efficiency and precision of threat analysis, reducing response times and minimizing the potential impact of security incidents. These features make Splunk ES a versatile and reliable tool for securing enterprise environments in an ever-changing threat landscape [26].

While Splunk Enterprise Security offers significant advantages, further research could explore additional enhancements and applications to address emerging cybersecurity challenges. Some potential areas for future work include:

- **AI-Powered Threat Intelligence:** Integrating more advanced AI models into Splunk ES could improve predictive threat analysis, allowing organizations to anticipate and prepare for potential attacks before they occur [2].
- **Integration with IoT and OT Security:** As the Internet of Things (IoT) and Operational Technology (OT) environments grow, research could focus on how Splunk ES can be adapted to monitor and secure these domains effectively [8].
- **Behavioral Analysis and Insider Threat Detection:** Developing enhanced behavioral analytics capabilities within Splunk ES could improve the detection of insider threats and anomalous activity that traditional methods might overlook [9].
- **Automated Incident Response:** Future studies could explore integrating Splunk ES with more sophisticated SOAR (Security Orchestration, Automation, and Response) capabilities to automate complex incident response workflows further [20].

These research directions would not only extend the capabilities of Splunk Enterprise Security but also strengthen its role as a critical tool in the ever-evolving cybersecurity landscape. By addressing these areas, future advancements can ensure that Splunk ES remains at the forefront of enterprise security solutions [14].

Acknowledgements

The author would like to thank Business Finland (BF) within the EUREKA CELTIC-NEXT project CISSAN (www.celticnext.eu.) for supporting this work.

References

Bryant Blake (2016) "Hacking SIEMS to Catch Hackers: Decreasing the Mean Time to Respond to Security Incidents with a Novel Threat Ontology in SIEM Software", Master's Thesis, University of Kansas

Florian Menges et al (2021) "Towards GDPR-compliant data processing in modern SIEM systems", Computers & Security, Volume 103, April 2021, 102165, https://doi.org/10.1016/j.cose.2020.102165

https://www.splunk.com/content/dam/splunk2/en_us/gated/ebooks/top-5-use-cases-for-splunk-security-analytics.pdf

J. Inns, "The evolution and application of SIEM systems," Netw. Secur., vol. 2014, no. 5, pp. 16–17, May 2014

J. Kaskade, "Magic Quadrant for Security Information and Event Management," p. 32

Kai-Oliver Detken et al (2017) "Combining Network Access Control (NAC) and SIEM functionality based on open source", 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), DOI: 10.1109/IDAACS.2017.8095094

Konstantinos Bezas et al (2023) "Comparative Analysis of Open Source Security Information & Event Management Systems (SIEMs)", Indonesian Journal of Computer Science, ISSN 2549-7286, DOI: 10.33022/ijcs.v12i2.3182

M. Di Mauro and C. Di Sarno, "Improving SIEM capabilities through an enhanced probe for encrypted Skype traffic detection," J. Inf. Secur. Appl., vol. 38, pp. 85–95, Feb. 2018

Marian Hristov et al (2021) "Integration of Splunk Enterprise SIEM for DDoS Attack Detection in IoT", IEEE 20th International Symposium on Network Computing and Applications (NCA), DOI: 10.1109/NCA53618.2021.9685977

M. Siva Niranjan Raja, A.R. Vasudevan (2017) "Rule Generation for TCP SYN Flood attack in SIEM Environment", 7th International Conference on Advances in Computing & Communications, ICACC-2017, Cochin, India, https://doi.org/10.1016/j.procs.2017.09.117

Muhammad Rijal Kamal et al (2021) "Anomaly Detection with Splunk Security Information and Event Management (SIEM) on UII Network", AUTOMATA Journal, https://journal.uii.ac.id/AUTOMATA/issue/view/1482

NIST. (2022). Guidelines for Automated Threat Response in SIEM Systems. Retrieved from https://www.nist.gov

Official Manual of Splunk®, Enterprise - Alerting Manual, version 8.2.2, Copyright © 2021 Splunk Inc

Official Manual of Splunk®, Enterprise - Getting Data In, version 8.2.2, Copyright © 2021 Splunk Inc

Roberto Bruzzese, (2019) "An Analysis of Application Logs with Splunk: Developing an App for the Synthetic Analysis of Data and Security Incidents", https://doi.org/10.48550/arXiv.1912.11283

Sandeep Bhatt et al (2014) "The Operational Role of Security Information and Event Management Systems", IEEE Security & Privacy (Volume: 12, Issue: 5, Sept.-Oct. 2014), DOI: 10.1109/MSP.2014.103

Seyed Morteza Zeinali, (2016) "Analysis of Security Information and Event Management (SIEM) Evasion and Detection Methods", Master's Thesis available at: https://mendillo.info/seguridad/tesis/Morteza.pdf

Sornalakshmi.K (2017) "Detection of DoS attack and Zero Day Threat with SIEM", International Conference on Intelligent Computing and Control Systems (ICICCS) 2017

Splunk Documentation. (2024). Use Cases for Pre-Built Content in Splunk ES. Retrieved from https://docs.splunk.com

Splunk Inc. (2024). How Adaptive Response Enhances Incident Management. Retrieved from https://docs.splunk.com

Splunk Inc. (2024). How Indexing Works in Splunk Enterprise. Retrieved from https://docs.splunk.com/Documentation/Splunk/latest/Indexer/HowSplunkstoresdata

Splunk Inc. (2024). Introduction to Forwarding and Receiving. Retrieved from https://docs.splunk.com/Documentation/Splunk/latest/Forwarding/Aboutforwardingandreceiving

Splunk Inc. (2024). Real-Time Monitoring and Threat Detection in Splunk. Retrieved from https://www.splunk.com

Splunk Inc. (2024). Search Head Overview. Retrieved from https://docs.splunk.com/Documentation/Splunk/latest/DistSearch/AboutSH

Splunk Inc. (2024). Splunk Architecture: Data-to-Everything Platform. Retrieved from https://www.splunk.com/en_us/resources.html

Splunk Inc. (2024). Splunk Enterprise Security Overview. Retrieved from https://docs.splunk.com/Documentation/ES/latest/Overview/IntroductiontoES

Splunk, "Splunk® Machine Learning Toolkit User Guide 5.2.0," Splunk Inc, 2020. https://docs.splunk.com/Documentation/MLApp/5.2.0/User/WelcometoMLTK (accessed Sep. 10, 2020)

S. S. Sekharan and K. Kandasamy, "Profiling SIEM tools and correlation engines for security analytics," in 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 2017, pp. 717–721

Stallings, W., & Brown, L. (2021). Computer Security: Principles and Practice. Pearson

Wahluf Abidian et al. (2021) "Implementing Splunk in Building SIEM Based on Firewall Logs: A Case Study of UII Network", AUTOMATA Journal, https://journal.uii.ac.id/AUTOMATA/issue/view/1390

Splunk Inc. (2021). Splunk Enterprise Overview. Retrieved from