

POLITECNICO DI TORINO

Master's Degree in Computer Engineering -
Cybersecurity



Master's Degree Thesis

Log Analysis for Network Anomalies Detection in Splunk

Supervisors

Prof. Alessandro SAVINO

Candidate

Alessandro ZAMPARUTTI

MONTH 2023/2024

Abstract

The rapid expansion of technology has resulted in a substantial rise in data generated by online applications, platforms, and digital services. This stream of information brings both advantages and challenges, specifically in the fields of data analysis and cybersecurity. This thesis focuses on using Splunk Enterprise and Splunk Infosec software and tools to further enhance network anomaly detection and security event analysis. Its primary objective is to develop a simple application that can be deployed within any Splunk infrastructure which allows to gain a general insights into network security as well as effective investigation of possible security threats.

Splunk Enterprise is a powerful and versatile tool known for its capabilities in data analysis, visualization, and monitoring. It provides a platform for ingesting, searching, and analyzing diverse datasets from different sources, including log messages, network traffic data, and security event logs. Additionally, Splunk Infosec provides specialized features and configurations specifically for security applications, offering dashboards, alerts, and visualizations designed to assist security monitoring and incident response.

To properly set up the infrastructure for ingesting, parsing, and normalizing relevant network data, this research explores into the essential tools offered by Splunk. This includes advanced analytics, real-time monitoring and alerting, and interactive visualization tools. By integrating various data sources and implementing risk-based alerting mechanisms, the aim is to establish a security monitoring solution capable of identifying patterns and behaviors suggestive of security breaches.

The methodology adopted involves implementing and testing different detection techniques, including the detection of network scanning activities and command and control attacks. By simulating attack scenarios throughout an event generator tool and analyzing the effectiveness of the queries and algorithms implemented, the thesis aim to evaluate the performance and reliability of the proposed security monitoring solution. The customization and configuration options available within Splunk can also optimize the detection capabilities and enhance the usability of the solution in different environment and systems, depending on the size of the monitored infrastructure.

Acknowledgements

I would like to thank all my friends for their support throughout the journey of completing this thesis. I want to thank you for all the countless hours spent together at the university, searching for a place to relax before lectures, for all the shared drinks the nights before an exam, and for the laughter that filled our days. Your encouragement, understanding, and patience have been invaluable to me, and I am grateful for your presence in my life.

I want to thank my family for being always supportive and for believing in me. I am truly fortunate to have such wonderful people by my side.

Thank you for being there for me every step of the way.

Table of Contents

List of Figures	VI
Acronyms	IX
1 Introduction	1
1.1 Overview	2
2 Splunk Enterprise: Architecture and Security Applications	3
2.1 Exploring the Structure of Splunk Enterprise	4
2.1.1 Data Processing Tiers	5
2.2 Search Processing Language	8
2.2.1 Main SPL Commands and Functionalities	10
2.2.2 Scheduled Searches and Alerts	11
2.3 Data Models	12
2.3.1 Normalization Through Data Models	14
2.3.2 Accelerated Data Models	15
2.3.3 The Common Information Model	17
2.4 Splunk Security Essentials	20
2.4.1 Network Service Discovery	21
2.5 Splunk InfoSec	21
2.5.1 InfoSec and CIM integration	22
2.5.2 Configuration of The InfoSec App	23
2.6 Simulating Real-World Data with Splunk Eventgen	25
2.7 Splunk Fortinet FortiGate Add-on	25
3 Design of the Splunk Application	27
3.1 Data Generation	27
3.1.1 Fortigate Sample Events Format	28
3.1.2 Eventgen Configurations	30
3.1.3 Simulation of Network Attacks	33
3.2 Model and Data Exploration	36

3.2.1	Configuration of CIM Network Traffic Data Model	36
3.2.2	Exploring the Generated Events	38
3.3	Experimental Searches and Analysis	40
3.3.1	Detect outliers in Network Traffic	41
3.3.2	Detect Network Scanning Activities	43
3.3.3	Detect Command And Control Attacks	44
3.3.4	Correlation of Traffic Data with Geo-Spatial Information . .	45
3.3.5	Network Communication Map	47
3.4	Dashboard Development	51
3.4.1	Dashboard Source Code	52
4	Experimental Results	54
4.1	Monitoring and Alerts Detection	54
4.1.1	Alerts Definition	57
4.1.2	Alert Actions for Incident Response	58
4.1.3	Alert Detection Results	58
4.2	Dashboard visualization as Continuous Monitoring system	60
4.2.1	Network Anomalies Visualizations	61
4.2.2	Network Communication Visualizations	62
4.2.3	Attacks Detection Within the Dashboard	63
4.3	Possible Detection Improvements	63
4.3.1	Splunk Enterprise Security	64
5	Conclusions	66
	Bibliography	68

List of Figures

2.1	Splunk graphical user interface including extracted fields and searched logs.	4
2.2	Scenarios and environments in which Splunk can be used to perform analysis, alerting and monitoring of the collected data	6
2.3	Example scheme of a clustered Splunk infrastructure	8
2.4	Example of a SPL query executed on the Splunk web interface to extract and visualize logs	9
2.5	Example of log visualization within Splunk graphical interface. . . .	10
2.6	Alert editing page with possible customization and action to perform at the moment of the triggered event	13
2.7	Pivot dashboard of the Network Traffic data model with configurable options to define expected visualization.	14
2.8	Edit page of the Authentication Data Model. Is it possible to see the search and tags that are used to identify authentication logs from different sources and the computed standardized fields	15
2.9	CIM Network Traffic Data Model with the corresponding search used to identify network logs and normalize them	18
2.10	Official table of the CIM Network Traffic Data Model mapping of fields.	19
2.11	MITRE ATT&CK matrix available in the Splunk Security Essentials dashboard with clickable items.	20
2.12	Network Service Discovery drilldown page with security content and possible attack detection	21
2.13	Security Content Basic Scanning page with mapping of the attack to MITRE ATT&CK tactics and techniques.	22
2.14	Example of a Security Posture page of the InfoSec application	23
2.15	InfoSec Health Dashboard with the status of the event ingested for each data models and the built percentage of the acceleration. . . .	24
2.16	InfoSec Health Dashboard with the status of the application required installed and their version.	24

3.1	Edit of the acceleration settings of the Network Traffic Data Model	39
3.2	Job inspector result of the query that search the events with source-type fortigate_traffic directly from the index. The search required 9,97 seconds to scan 148795 events.	40
3.3	Job inspector result of the query that search the events with source-type fortigate_traffic exploiting the Network Traffic data model. The search required 7,698 seconds to scan 148795 events.	40
3.4	Table representation of the result of the search to detects outliers	43
3.5	Cluster map visual representation of the blocked traffic by destination port. The panel is part of the application created with this thesis research.	48
3.6	Table visualization for the Application Panel within the Network Anomalies Dashboard.	49
3.7	Force directed graph visualization of the communication of different hosts. The graph was filtered to show the traffic communications and paths of hosts with the IP address 8.8.8.8 (a well known DNS address).	51
3.8	Dashboard edit configuration interface to add panel with statistical table.	52
3.9	Dashboard source xml code while in edit mode. The code describe the configuration of a cluster map.	53
4.1	Alert action email configuration with all possible options.	56
4.2	The timechart displays the count of traffic events, categorized into regular traffic and attacker traffic. The graph illustrates that the attacker generates a notably higher volume of traffic events within a short time window.	59
4.3	The timechart displays the count of traffic events, categorized into regular traffic and attacker traffic.	60
4.4	The timechart shows the highlighted count of traffic events of the attacker. A pattern of regular communication is evident, characterized by low traffic over an extended time window.	60
4.5	Possible filtering options included in the Network Communication Map section of the dashboards.	61
4.6	KPIs panels along with the correspondent tables that carries information about the outliers and possible threats detected.	62
4.7	Geographical maps with blocked incoming traffic by destination port and incoming traffic by application, and statistical panels.	62
4.8	Incident Review dashboard of Splunk Enterprise Security with notable events used to start investigation inside the application.	65

Acronyms

CIM

Common Information Model

CSV

Comma Separated Values

ES

Enterprise Security

GUI

Graphical User Interface

HEC

HTTP Event Collector

HTTP

Hypertext Transfer Protocol

IDS

Intrusion Detection Systems

IP

Internet Protocol

IPS

Intrusion Prevention Systems

JSON

JavaScript Object Notation

KPI

Key Performance Indicators

LDAP

Lightweight Directory Access Protocol

OS

Operating System

SH

Search Head

SPL

Search Processing Language

TCP

Transmission Control Protocol

UDP

User Datagram Protocol

UF

Universal Forwarder

UTM

Unified Threat Management

XML

eXtensible Markup Language

Chapter 1

Introduction

In the past decade, the landscape of personal computing has witnessed significant changes, accompanied by the rapid expansion of networking capabilities. This technological evolution has led to the development of increasingly complex and compute-intensive applications that serve companies and their clients with faster and more efficient devices. However, the rapid growth of technology has resulted in a massive stream of data generated by the communication and interconnection of those software and applications, presenting challenges in data analysis and security implementation.

For businesses operating in this dynamic environment, the effective detection of network anomalies has become crucial to maintaining a safe and secure work environment within their systems. As data complexity grows, it becomes necessary to adopt advanced methodologies that facilitate easy and efficient anomaly detection.

This thesis aims to address the need for effective network anomaly detection using Splunk, a powerful and versatile tool for data analysis and security applications, and to assist organisations in identifying and mitigating potential threats within their network infrastructure by offering an easily comprehensible approach, and reducing the need for substantial effort or a high level of expertise. Through the exploration of Splunk's capabilities and methodologies, is it possible to provide valuable insights into the process of detection of network anomalies, ultimately contributing to enhanced security and productivity within companies.

Splunk's capability to handle multiple data formats, including logs, authentication information, and web responses, makes it a powerful software for enhanced security monitoring and investigations. This flexibility not only accommodates the large amount of data generated in today's environment but also enables the definition of complex queries and algorithms.

1.1 Overview

This thesis work involves an initial phase of studying various applications and infrastructure within the Splunk platform. The study begins with an in-depth exploration of the Splunk Infrastructure, focusing on understanding the diverse resources and their functionalities within the system. Subsequently, the thesis examines Splunk Fundamentals, describing the various methods by which Splunk Enterprise acquires data from different sources.

To further enrich the research, this work emphasizes the role of two integral components of Splunk's security and information management capabilities: Splunk InfoSec and the Common Information Model (CIM). These components are fundamental in facilitating robust security event analysis, threat detection, and incident response.

The Security Essentials application offers useful guidelines on the discovery of security content within the system, and allows to explore use cases and map data to well known cybersecurity frameworks.

The Splunk InfoSec application provides predefined configurations, dashboards and alerts which extend the potentiality of the underlying Splunk infrastructure, and allows organizations to gain visibility into security events across their network and applications. It enables security teams to develop and build real-time monitoring, detect anomalies, and conduct effective incident investigations.

Complementing the capabilities of Splunk InfoSec, the Common Information Model serves as a standardized framework for data normalization and standardization. By employing CIM, the thesis demonstrates how security data from various sources can be consistently translated into a common format, enabling seamless integration, correlation, and analysis of security events.

To ensure the availability of relevant and realistic data for experimentation, the project employs Splunk Eventgen for data generation, simulating various network anomalies and security events. The use of Splunk Eventgen facilitates the creation of data-sets representative of actual security scenarios, allowing for comprehensive testing and evaluation of the implemented security techniques.

Throughout the thesis, special attention will be given on defining the visualization of identified anomalies and exploring communication paths within a dashboard. Additionally, the performance of the provided queries will be evaluated to ensure that the infrastructure can effectively operate without delays or overloads.

Chapter 2

Splunk Enterprise: Architecture and Security Applications

This chapter provides an in-depth introduction to the Splunk software and its underlying architecture, highlighting how various applications can enhance the potentiality and flexibility of data analysis.

It begins by exploring the structure of the Splunk Enterprise tool and its data analysis capabilities, powered by the Splunk Processing Language (SPL). Subsequently, the focus shifts to the Common Information Model (Splunk CIM), a standardized framework designed to seamlessly integrate and normalize a wide range of diverse data sources. The chapter then examines into the sophisticated security features offered by the Splunk InfoSec app. This app empowers organizations to fortify their cybersecurity posture by proactively detecting potential threats and responding to security incidents. Moreover, the chapter explores the significance of event detection and response in an automated environment. Splunk's robust capabilities enable the identification of critical events and the automation of predefined response actions, ensuring timely and efficient incident management. In conclusion, it is provided an overview of two distinct Splunk applications employed to generate and manage logs for the experimental section of the thesis: Splunk Eventgen and the Splunk Fortinet FortiGate Add-On. The former is responsible for generating test events using sample data and customizing log creation, while the latter serves to parse and standardize logs originating from Fortinet FortiGate firewalls.

2.1 Exploring the Structure of Splunk Enterprise

Splunk is a well-known data analytics tool with strong capabilities for collecting, indexing, and analyzing immense quantities of machine-generated data. It offers a scalable and effective method for processing numerous data sources, making it a valuable tool for a wide range of businesses and use cases. The Splunk interface enables users to analyze log messages originating from diverse systems or devices. Its primary objective is to identify and investigate potential failures or anomalies efficiently. Furthermore the software excels in security applications since it enables businesses to effectively recognize and take action against security threats in real-time.

The graphical user interface (GUI) visible in the Figure 2.1, comprehends extracted fields from the data, the actual log itself, and additional information provided directly by Splunk, enabling efficient searching of desired logs.

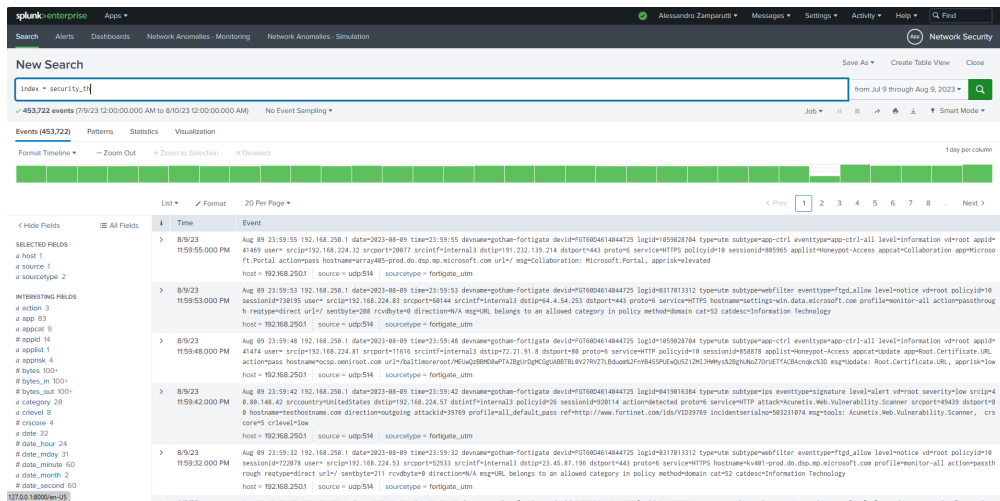


Figure 2.1: Splunk graphical user interface including extracted fields and searched logs.

Some of the key features and characteristics of Splunk Enterprise are:

1. **Data Input:** Splunk Enterprise can ingest data from a wide range of sources, including log files, metrics from applications, network devices, cloud services, and more. It supports various input methods such as file monitoring, network inputs, scripted inputs, and HTTP Event Collector (HEC).
2. **Data Indexing:** Once the data is received, Splunk performs various operations, such as parsing, breaking, and timestamping of the events, to correctly index the data. This indexing process ensures fast and efficient data retrieval. The

indexed data is then stored in compressed and optimized data structures known as buckets.

3. **Search and Analysis:** Splunk provides a powerful search language called the Splunk Processing Language (SPL), which allows users to run searches and create complex queries to analyze the indexed data. Users can apply filters, aggregations, and transformations to gain meaningful insights from their data.
4. **Dashboards and Data Visualization:** Splunk offers a variety of visualization tools, including charts, graphs, tables, and dashboards. These tools empower users to craft interactive and customizable visual representations, effectively showing complex data insights in a clear and insightful manner. Dashboards play a central role in consolidating and presenting metrics and statistics, allowing users to monitor real-time data trends, identify anomalies, and make data-driven decisions.
5. **Alerting and Monitoring:** Splunk allows users to set up alerts based on scheduled searches and predefined conditions, ensuring prompt notifications of critical events or anomalies. These alerts can trigger automated actions or notifications, ensuring timely and appropriate responses to the identified events.
6. **Data Models:** provide a semantic layer that facilitates the organization and correlation of data from diverse sources. These models define the relationships and field extractions necessary to create unified and coherent data representations, enabling users to analyze complex datasets more effectively.
7. **Apps and Add-ons:** Splunk's modular architecture allows users to enhance its capabilities through apps and add-ons. These pre-built extensions provide specific functionality, such as security analytics, network monitoring, and application performance monitoring.
8. **User Access and Roles:** Splunk offers robust user access controls and role-based permissions to ensure that the right users have appropriate access to data and functionalities. It is also possible to configure access to the system to adopt various authentication scheme such as LDAP.

2.1.1 Data Processing Tiers

In the Splunk Enterprise architecture, each instance serves a specific purpose and is categorized into one of the three processing tiers, each corresponding to a primary processing function:

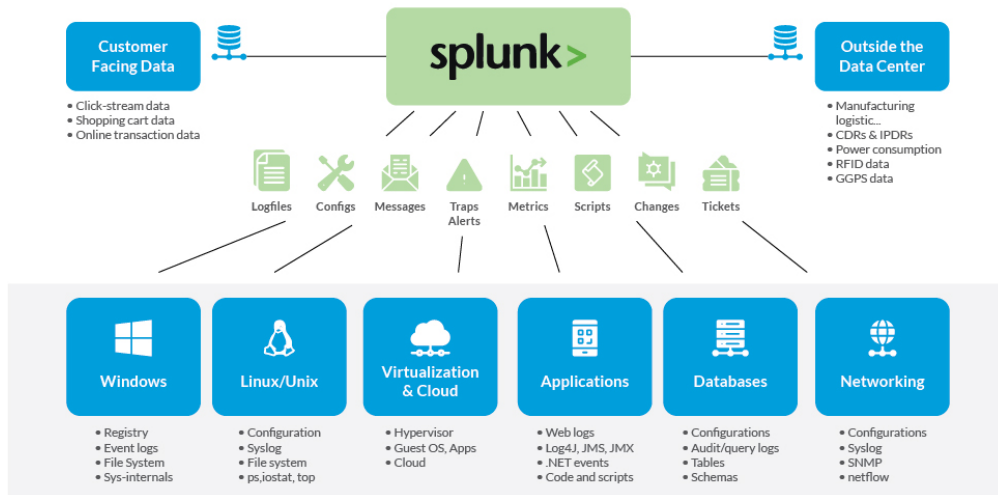


Figure 2.2: Scenarios and environments in which Splunk can be used to perform analysis, alerting and monitoring of the collected data

- Data input tier
- Indexing tier
- Search processing tier

Each instance within these tiers has a unique functionality and name, determined by its specific role in the data processing journey.

Data Input Tier: Forwarders

The components known as Forwarders belong to the data input tier and are solely responsible for collecting data from various sources and forwarding it to the indexing tier. Splunk offers a specialized agent called Splunk Universal Forwarder, a lightweight software component capable of monitoring files and directories on machines. Its primary function is to send the collected data to the indexing tier while providing the option to include additional metadata to enhance the events context. Notably, a forwarder can be any third party agent with the capability to transmit data via HTTP, TCP, UDP, or other protocols.

Indexing Tier: Indexers

The indexer plays a crucial role in a Splunk infrastructure as it is responsible for receiving, processing, and storing the data ingested from various sources. When

data is forwarded from the different collectors, it arrives at the indexer, which is in charge of parsing and breaking the events into key-value pairs and extracting relevant fields and metadata from the raw data. After parsing, the component indexes the data in a high-performance database optimized for searching called index. The Indexed data is subsequently stored in buckets, which are compressed and optimized data structures used to efficiently store large volumes of data over time.

In a distributed Splunk environment, multiple indexers may be deployed for redundancy and high availability, where the indexers handle data replication, ensuring that events are securely and redundantly stored across multiple nodes to prevent data loss in case of failures. The indexer also manages the retention and lifecycle of data, allowing administrators to define how long data should be preserved and when it should be archived or deleted. This capability is essential for compliance, storage optimization, and data governance. Indexers also assume the responsibility of extracting results from user-submitted queries, performing the search, and leveraging the index to efficiently retrieve relevant data. The efficient indexing and search capabilities of the indexer contribute to Splunk's real-time data analysis and visualization capabilities.

Search Processing Tier: Search Heads

Splunk Enterprise offers a web page endpoint that serves as the primary interface connecting the powerful capabilities of the Splunk platform with end-users. This web interface and its functionalities are typically managed by a component known as the Search Head. The Search Head acts as a central hub, providing users with an intuitive and user-friendly way to interact with the Splunk system and access its features and insights.

As the primary component handling user interactions, the Search Head facilitates the execution of search queries submitted by users. It is essential for converting users' search requests into efficient and optimized queries that leverage the indexing and data processing capabilities of the underlying indexers. By orchestrating these searches and interacting with the indexers, the SH ensures that users receive relevant and timely results for their queries. Moreover, the SH offers a range of features that allow users to create, save, and manage their search queries, visualizations, and dashboards. Through the graphical interface, users can customize and adapt search results to meet their specific requirements, enabling them to obtain valuable information and discover meaningful patterns in the data. Search Head provides users with the possibility to configure access controls, define roles, and manage permissions, ensuring secure and controlled access to sensitive data and functionality in Splunk environments. This capability is especially important in large organizations, where data access and security are crucial aspects of data

governance and compliance.

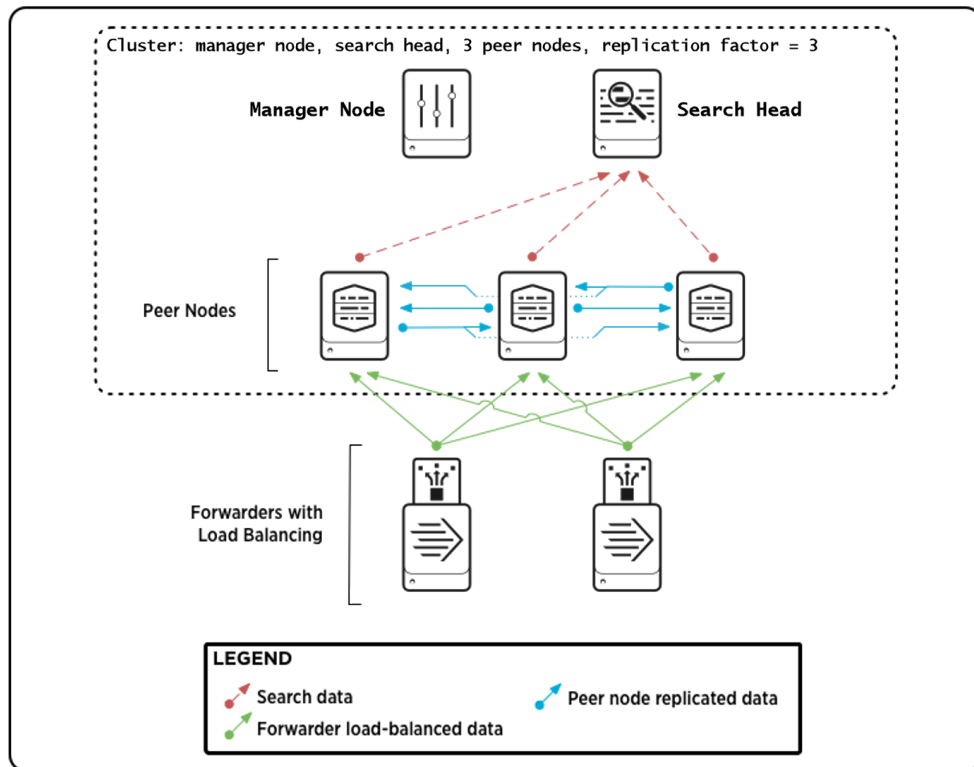


Figure 2.3: Example scheme of a clustered Splunk infrastructure

2.2 Search Processing Language

The Splunk Processing Language (SPL) is a powerful query language designed specifically for data analysis and retrieval within the Splunk platform. As the core of Splunk’s search functionality, SPL enables users to efficiently search, analyze, and visualize large volumes of machine-generated data in real-time. The indexed data of the Splunk platform is used by SPL, which enables users to utilize a variety of search and filtering methods to identify specific events or patterns in the data. The language offers an extensive set of commands and functions that enable data manipulation, aggregate and extraction, giving users the ability to uncover useful information from raw log data.

The SPL offers a comprehensive set of functionality that ensures efficiency and simplicity in the data searching process. Some of its key features include:

1. Search and Filtering: SPL supports a rich set of search commands that allow

users to define precise queries and filter data based on specific criteria. With intuitive syntax, users can combine multiple search conditions, enabling the retrieval of targeted information quickly.

2. Transformations and Aggregations: SPL provides commands for transforming and aggregating data, allowing users to group events, calculate statistics, and generate summary metrics. These transformations allow to simplify complex data sets and focus on critical information.
3. Field Extractions: SPL allows users to extract additional fields from the raw data, converting unstructured logs into structured events. This feature enhances the analysis process, as extracted fields become attributes that can be used for searching, filtering, and visualizing data. Splunk offers built-in automation for the extraction of specific fields from ingested logs, which is based on the source type of the data. This automated process streamlines the extraction of relevant information from raw logs, making it more convenient and efficient for users to work with the data.
4. Visualization and Dashboards: SPL integrates the search results with Splunk's visualization tools, enabling users to create interactive and customizable charts, graphs, and dashboards.
5. Advanced Data Analysis: SPL offers advanced functions and techniques for time series analysis, predictive analytic, and data modeling. With the ability to perform complex calculations and predictive tasks, users can gain deeper insights and make data-driven decisions.
6. Real-time Analysis: One of the core strengths of SPL is its real-time data analysis capabilities. As data is ingested into Splunk, SPL is able to processes and indexes it, allowing users to perform searches and visualize results on-the-fly.

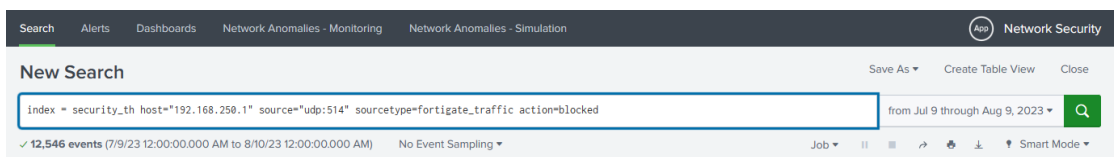


Figure 2.4: Example of a SPL query executed on the Splunk web interface to extract and visualize logs

As it is possible to notice in the Figure 2.5 there are significant information provided by splunk for each log: the host, the source and the sourcetype. The host attribute

designates the origin of the log, signifying the machine or device responsible for generating the data. The source field identify the specific file name, input source or data stream from which the logs was collected. Additionally, the sourcetype field categorizes the log data based on its format, facilitating the interpretation process or the extraction of data with the same format.

SPL is an essential tool for data analysts, security experts, and IT operations teams due to its adaptability and versatility. It permits to transform raw logs into valuable information creating new fields starting from the data present in the original logs and compute statistics or troubleshoot anomalies. This is made possible by its notable characteristics, along with a simple syntax, which enables users to promptly explore and make complex data sets more meaningful.



Figure 2.5: Example of log visualization within Splunk graphical interface.

2.2.1 Main SPL Commands and Functionalities

The following list represent the most common used SPL commands to efficiently extract the needed information from the raw events, to compute additional values starting from the fields already present and to perform statistics and aggregate information. All commands are listed sequentially and preceded by the pipe character "|" to denote that the output of the preceding command is passed as input to the subsequent command in the pipeline.

Some of the most common SPL commands are:

- **search:** used to start a new search and generate events from the specified sources. The command can be omitted if it is used in the first line of the query, and can be added later to start a subsearch to include events retrieved from different sources or indexes with respect to the initial ones.

- **eval:** used to calculate an expression and assign the result to a variable that becomes a result field. Within this command other useful operation can be performed to evaluates mathematical, string, and boolean expressions, such as if and case instruction, string and numerical comparison or generation of multi-value fields [1].
- **stats:** used to compute aggregated statistics such as average, count, and sum, over the results set. In the output result a row is generated for each distinct value specified in the by clause [2].
- **timechart:** used to perform statistical aggregation applied to a field to produce a chart, with time used as the X-axis. The operation that can be applied are similar to the the one of the stats command but the output includes always the time filed. Mainly used to display the information in a timeline chart.
- **where:** used to compare a field with a specific value or with a different field to filter the events.
- **lookup:** used to extract fields from a KV store or CSV lookup table, to include information to the previously searched events. It is possible to assign to each event a specific value from the lookup table by matching the events with the fields included as argument to the command.
- **rex:** used to extract new fields using regular expression named groups starting from the raw logs of from an already extracted field.

2.2.2 Scheduled Searches and Alerts

Scheduled searches and alerts are integral components of Splunk’s functionality, playing an important role in proactive monitoring, incident detection, and timely response within an organization’s data environment. These features promote rapid awareness of important events and trends, enabling immediate intervention that strengthens security and performance optimization.

Scheduling of alerts and reports in Splunk involves the execution of underlying scripts, primarily written in the Python language, and are scheduled through the utilization of the 'crontab' scheduler. These scripts are responsible for orchestrating various tasks within Splunk, including the acceleration of data models and generation of lookup files. The crontab scheduler operates within the operating system environment and enables the periodic execution of these Python scripts, ensuring timely and automated scheduling of alerts and reports as required by the Splunk configuration.

Scheduled Searches

Scheduled searches enable splunk users to automate the execution of specific searches on a predefined schedule. These queries can be configured to run at precise intervals, ranging from daily to weekly or with more a detailed cron schedule. The outcomes of these searches can be stored, indexed, or transformed into alerts, reports, or visual representations.

This automation capability simplify the data query process and ensure consistent generation of critical information without requiring manual involvement. Additionally, these results can be seamlessly integrated into dashboards, reducing the need to repeatedly execute searches when visualizing the different panels. This integration enhances efficiency and usability, promoting the optimal utilization of search results for informed decision-making.

Alerts

Alerts in Splunk are automated notifications triggered by predefined conditions or thresholds in the data. Alerts are used to promptly notify relevant stakeholders or execute an automated response when specific events or patterns occur within the data. This proactive approach to monitoring ensures that potential issues or security threats are detected in real-time, allowing for a rapid response and mitigation.

Alerts provide continuous monitoring of incoming data to detect specific conditions. They operate through scheduled executions of searches that users define. These searches can be customized to identify specified conditions and thresholds, leveraging various data attributes and patterns. Alerts can be configured to send notifications via email, SMS, or integration with third-party tools, such as incident response platforms. Additionally, it is possible to execute custom scripts that perform specific actions when an alert is triggered, enhancing the automation of error response processes.

2.3 Data Models

Data Models in Splunk are mostly used to facilitate the process of normalization by enabling the integration and organization of data from diverse sources and types into a consistent and structured format.

Data Models provide a logical framework for structuring and categorizing data, without considering the data source. They offer a standardized schema that defines fields, relationships, and semantics. This schema is crucial for achieving data normalization, as it transforms different data into a unified format, enabling seamless analysis and correlation. Data Models are designed to accommodate a

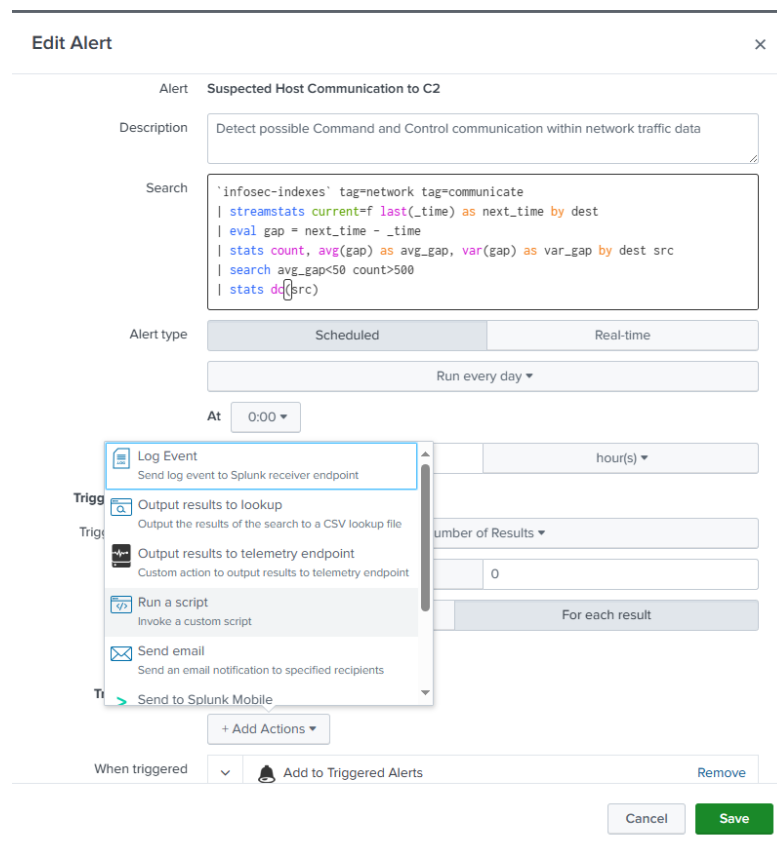


Figure 2.6: Alert editing page with possible customization and action to perform at the moment of the triggered event

wide array of data, including logs, events, and transactions, thus enhancing the efficiency and accuracy of data analysis processes. Within the security landscape, standardization of data models is necessary to facilitate communication among diverse software and tools. This standardization ensures uniformity in naming conventions and information structure, allowing Splunk to effectively exchange and interpret data without discrepancies or interoperability issues.

Data models are knowledge objects generated by initially searching for specific logs, such as Authentication events, and labeling all extracted data with event types and tags. By Utilizing these designated tags, it is possible to extract all related events without the need to search for a specific index or sourcetype, enabling immediate access to data with shared attributes. In the model creation interface, users can then link all necessary tags to the data model, establishing a naming convention for the extracted fields.

Pivot tool

All data models in Splunk offer a valuable feature for creating a Pivot directly from the model. A Pivot serves as a Splunk knowledge object, enabling users to search for logs and data within the data model without needing to write specific queries using the Search Processing Language to extract events. Instead, it is possible to utilize predefined structures available in the interface, such as column and field selection, filtering options, visualization tools, and time range definition to create the desired output. These features prove especially beneficial when non-specialized users need to create and search for specific events efficiently.

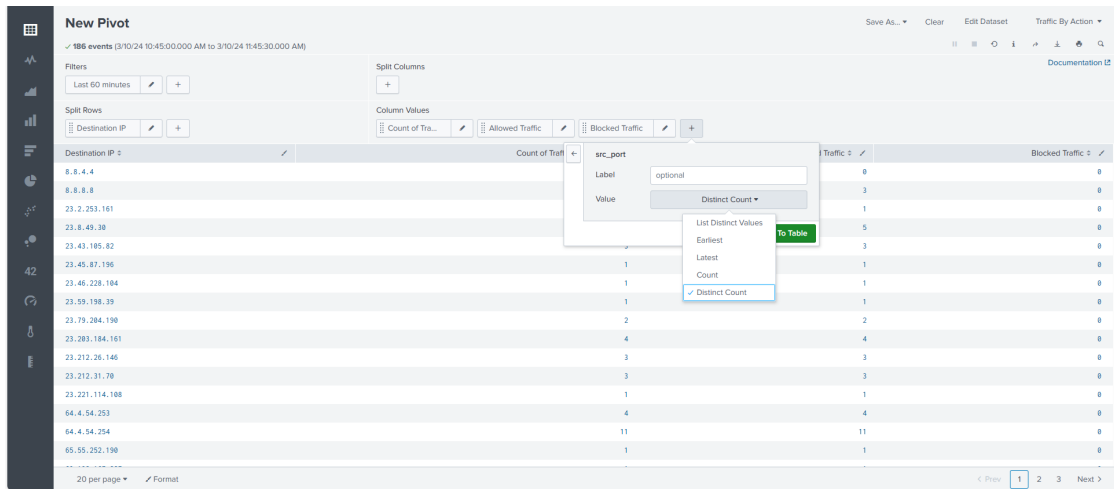


Figure 2.7: Pivot dashboard of the Network Traffic data model with configurable options to define expected visualization.

2.3.1 Normalization Through Data Models

Data Models enable normalization by offering predefined data structures that categorize similar data elements into common fields. This process involves mapping disparate field names, data types, and units of measurement into standardized attributes. For instance, different log sources might use variations of terms like "user," "username," or "account" to represent the same entity. Data Models unify these variants under a single attribute, ensuring uniformity in data representation.

By performing the normalization of different logs it is possible to ensure consistent field naming conventions and units of measurement across different data sources, eliminating ambiguities and simplifying the analysis. Furthermore, normalized data becomes highly correlated, enabling effective cross-referencing and correlation

between information from various sources. This correlation enhances the ability to detect patterns, trends, and anomalies.

In the field of data analysis, the normalization procedure introduces a certain degree of simplicity. The need for complex queries to decode different formats is unnecessary, thus accelerating and simplifying the analysis process. Additionally, the normalization process offers the invaluable potential to enrich the events extracted from the logs with metadata and contextual information, which improves the understanding of the underlying patterns, relationships, and insights hidden within the data landscape.

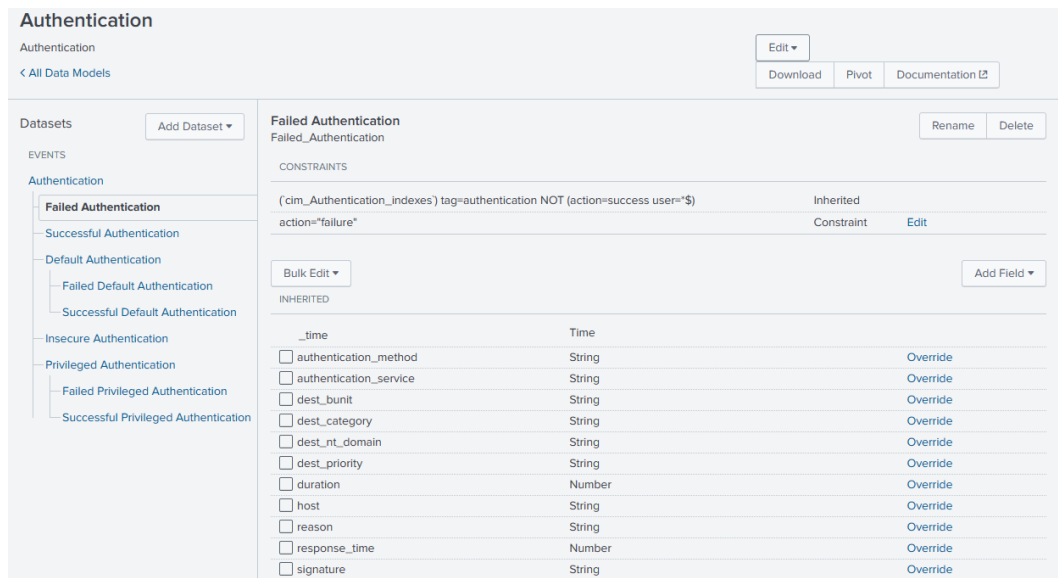


Figure 2.8: Edit page of the Authentication Data Model. Is it possible to see the search and tags that are used to identify authentication logs from different sources and the computed standardized fields

2.3.2 Accelerated Data Models

When managing large sets of information the efficiency and speed of the search and analysis processes are fundamental. Splunk’s accelerated data models can be adopted to achieve these goals by providing pre-built, optimized structures for data analysis. Accelerated data models are designed to expedite the analysis process, particularly for complex and high-volume data sources. They combine the power of Splunk’s data processing capabilities with the efficiency of accelerated search techniques, resulting in significantly faster query performance and enhanced user experience.

The pre-processed nature of these models eliminates the need for manual data transformations and complex query formulations. Furthermore, accelerated data models take advantage of Splunk's underlying architecture, optimizing the storage and retrieval of data. This means that queries and searches conducted on accelerated data models yield results with remarkable efficiency, providing real-time insights.

The Process of Data Model Acceleration

The process of accelerating data models in Splunk involves optimizing the structure and indexing of data for faster and more efficient analysis. This acceleration is achieved through a combination of pre-computation, data summarization, and indexing techniques:

1. **Pre-built Structures:** Accelerated data models are pre-built templates that include field extractions, knowledge objects, and predefined relationships. These templates are designed to capture domain-specific information and insights, ensuring that the analysis process is improved.
2. **Data Summarization:** One of the core techniques in accelerating data models is data summarization. This involves pre-computing and summarizing key statistics and metrics within the data. These summaries are stored alongside the raw data, enabling faster retrieval during analysis. By summarizing data, the system reduces the need to perform complex calculations during query execution, which significantly speeds up the analysis process.
3. **Data Aggregation:** Accelerated data models leverage data aggregation to consolidate information across different dimensions. Aggregation involves grouping data by specific attributes, such as time, source, or event type. This process reduces the volume of data that needs to be processed during queries, enhancing query performance and responsiveness.
4. **Indexing Techniques:** Splunk's underlying architecture relies on indexing to facilitate quick data retrieval. Accelerated data models utilize optimized indexing techniques that align with the specific structure of the data model. These indexes allow the system to pinpoint relevant data more efficiently, minimizing the time taken to retrieve information.
5. **Metadata and Lookups:** Accelerated data models often incorporate metadata and lookup tables. These elements enrich the data by adding context and additional information. During acceleration, metadata and lookup data are computed during the build phase and indexed, allowing for rapid access in the analysis.

6. **Caching:** Splunk employs a caching mechanism to store intermediate results of queries. This caching optimizes the query execution process by reusing previously calculated results when similar queries are executed. Caching reduces redundant calculations and improves overall performance.

2.3.3 The Common Information Model

The Common Information Model is a standardized framework within Splunk that facilitates the integration, normalization, and correlation of diverse data sources. CIM acts as a common language that allows different data formats and sources to be understood and correlated, simplifying the process of data analysis and making it easier to derive meaningful insights. The primary objective of CIM is to provide a consistent and structured way to interpret data, regardless of its origin or format.

A significant majority of the applications available within Splunkbase, which serves as the official repository for a wide range of add-ons that can be seamlessly integrated as applications within a Splunk installation, have been designed to be compliant to the Common Information Model standards. These CIM-compliant applications are crafted with the aim of ensuring compatibility and consistency in terms of data interpretation, formatting, and correlation within the Splunk ecosystem, allowing for flawless integration and interoperability between various applications and data sources.

Some of the key features provided are:

1. **Normalization:** One of the fundamental functions of CIM is data normalization. It defines a standardized set of field names and definitions for various types of data, such as network traffic, security events, and system logs. This ensures that data from different sources adheres to a common naming convention, reducing ambiguity and enhancing interoperability.
2. **Unified Data Model:** CIM offers a unified data model that presents a comprehensive view of data attributes across various domains. This model includes essential fields and attributes for different data categories, making it easier to correlate and analyze data from different sources.
3. **Field Mapping:** CIM maps fields from different data sources to standardized CIM field names. This mapping allows data analysts to compare and combine data from various sources without needing to understand the specific terminology used by each source.
4. **Data Enrichment:** CIM provides a framework for enriching data by adding context and metadata to events. This context helps in better understanding the significance of events and enables more accurate analysis.

5. Normalization and Tagging Add-ons: Splunk offers CIM-compatible add-ons that can be installed to normalize and tag data. These add-ons translate data from specific sources into the standardized CIM format, ensuring consistent and coherent data interpretation.

Common Information Model: Network Traffic

The CIM Data Model for Network Traffic is designed to enhance the understanding and analysis of network data, which is crucial in the context of cybersecurity, network monitoring, and threat detection. It provides a common language and structure for describing network-related events, allowing to efficiently manage and respond to security incidents, performance issues, and anomalies. The CIM Data Model for Network Traffic encompasses a range of fields and event types related to network interactions. These include details about source and destination IP addresses, ports, protocols, bytes transferred, and timestamps. By organizing these attributes into a structured format, the Data Model facilitates consistent interpretation and correlation of network data across various data sources, devices, and applications.

Network Traffic
 Network_Traffic
 < All Data Models

Edit
 Download Pivot Documentation

⚠ This Data Model cannot be edited because it is accelerated. Disable acceleration in order to edit the Data Model.

Datasets
 EVENTS
 All Traffic
 Traffic By Action
 Allowed Traffic
 Blocked Traffic

Traffic By Action
 Traffic_By_Action

CONSTRAINTS

(cim_Network_Traffic_indexes) tag=network tag=communicate	Inherited
action=*	Constraint

INHERITED

_time	Time
app	String
channel	Number
dest_bunit	String
dest_category	String
dest_interface	String
dest_ip	String
dest_mac	String
dest_priority	String
dest_translated_ip	String
dest_translated_port	Number
dest_zone	String
direction	String
duration	Number

Figure 2.9: CIM Network Traffic Data Model with the corresponding search used to identify network logs and normalize them

The CIM Data Model for Network Traffic finds application in a variety of scenarios, including:

- **Cybersecurity:** Identifying and responding to network anomalies, intrusions, and potential security breaches.
- **Network Monitoring:** Monitoring network performance, identifying bottlenecks, and optimizing network resources.
- **Incident Response:** Investigating security incidents by analyzing network communication patterns and identifying potential attack vectors.

To properly configure and map each field from the ingested log to its corresponding field in the data model, it is necessary to refer to the official documentation provided by Splunk. In the Figure is it possible to see some of the fields mapping with a description and list of example values. For the complete table of fields, refer to the official web page.

Dataset name	Field name	Data type	Description	Abbreviated list of example values
All_Traffic	<code>action</code>	string	The action taken by the network device.	<ul style="list-style-type: none"> • recommended • required for pytest-splunk-addon • prescribed values: <code>allowed</code>, <code>blocked</code>, <code>teardown</code>
All_Traffic	<code>app</code>	string	The application protocol of the traffic.	required for pytest-splunk-addon
All_Traffic	<code>bytes</code>	number	Total count of bytes handled by this device/interface (<code>bytes_in</code> + <code>bytes_out</code>).	recommended
All_Traffic	<code>bytes_in</code>	number	How many bytes this device/interface received.	recommended
All_Traffic	<code>bytes_out</code>	number	How many bytes this device/interface transmitted.	recommended
All_Traffic	<code>channel</code>	number	The 802.11 channel used by a wireless network.	
All_Traffic	<code>dest</code>	string	The destination of the network traffic (the remote host). You can alias this from more specific fields, such as <code>dest_host</code> , <code>dest_ip</code> , or <code>dest_name</code> .	<ul style="list-style-type: none"> • recommended • required for pytest-splunk-addon
All_Traffic	<code>dest_bunit</code>	string	colspan="2" rowspan="2">These fields are automatically provided by asset and identity correlation features of applications like Splunk Enterprise Security. Do not define extractions for these fields when writing add-ons.	
All_Traffic	<code>dest_category</code>	string		
All_Traffic	<code>dest_interface</code>	string	The interface that is listening remotely or receiving packets locally. Can also be referred to as the "egress interface."	
All_Traffic	<code>dest_ip</code>	string	The IP address of the destination.	
All_Traffic	<code>dest_mac</code>	string	The destination TCP/IP layer 2 Media Access Control (MAC) address of a packet's destination, such as <code>06:10:9f:eb:8f:14</code> . Note: Always force lower case on this field. Note: Always use colons instead of dashes, soaces, or no separator.	

Figure 2.10: Official table of the CIM Network Traffic Data Model mapping of fields.

2.4 Splunk Security Essentials

Splunk Security Essentials serves as a centralized platform for the understanding of security monitoring, threat detection, and incident response. It features a variety of dashboards designed to guide users through the process of collecting necessary data and identifying specific threats effectively. One of the main features provided by the application is the ability to explore various attacks and phases of the cyber kill chain, and navigate through different attack scenarios. By selecting a specific attack it is possible to see information about its behaviour along with detailed description and mapping to the MITRE ATT&CK framework, providing valuable links to corresponding pages for further insight. In most of the cases the application also provides some example of queries in the SPL language that can be used as a baseline for the implementation of the detection of the specific threat. Additionally, Splunk Security Essentials includes compliance monitoring capabilities to help organizations adhere to regulatory requirements and security standards, by offering predefined compliance checks, reports, and alerts to identify gaps in compliance and mitigate risks.

MITRE ATT&CK Matrix

Content (Total)

ATT&CK version:14.1

Reconnaissance	Resource Development	Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Active Scanning	Acquire Access	Content Injection	Cloud Administration Command	Account Manipulation	Abuse Elevation Control Mechanism	Abuse Elevation Control Mechanism	Adversary-in-the-Middle	Account Discovery	Exploitation of Remote Services	Adversary-in-the-Middle	Application Layer Protocol	Automated Exfiltration	Account Access Removal
Gather Victim Host Information	Acquire Infrastructure	Drive-by Compromise	Command and Control Infrastructure	BITS Jobs	Access Token Manipulation	Access Token Manipulation	Brute Force	Application Window Discovery	Internal Spearphishing	Archived Collected Data	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
Gather Victim Identity Information	Compromise Accounts	Exploit Public-Facing Application	Container Administration Command	Boot or Logon Assistant Elevation	Account Manipulation	BITS Jobs	Credentials from Password Store	Browser Information Discovery	Lateral Tool Transfer	Audio Capture	Content Injection	Exfiltration Over Alternative Protocol	Data Encrypted for Impact
Gather Victim Network Information	Compromise Infrastructure	External Remote Services	Deploy Container	Boot or Logon Initialization Scripts	Boot or Logon Assistant Elevation	Build Image on Host	Exploitation for Credentials Access	Cloud Infrastructure Discovery	Remote Session Hijacking	Automated Collection	Data Encoding	Exfiltration Over Cloud	Data Manipulation
Gather Victim Org Information	Develop Capabilities	Hardware Additions	Exploitation for Eject Elevation	Browser Extensions	Boot or Logon Initialization Scripts	Debugger Evasion	Forced Authentication	Cloud Service Dashboard	Remote Services	Browser Session Hijacking	Data Obfuscation	Exfiltration Over Other Network Medium	Defacement
Phishing for Information	Establish Accounts	Phishing	Inter-Process Communication	Compression Client Software Binary	Create or Modify System Process	DevFunctify/Decode Files or Information	Forge Web Credentials	Cloud Service Discovery	Replication Through Removable Media	Clipboard Data	Dynamic Resolution	Exfiltration Over Physical Medium	Disk Wipe
Search Closed Sources	Obtain Capabilities	Replication Through Removable Media	Native API	Create Account	Domain Policy Modification	Deploy Container	Input Capture	Cloud Storage Object Discovery	Software Deployment Tools	Data Staged	Encrypted Channel	Exfiltration Over Web Service	Endpoint Denial of Service
Search Open Technical Capabilities	Stage Capabilities	Supply Chain Compromise	Scheduled Task/Job	Create or Modify System Process	Escape to Host	Direct Volume Access	Modify Authentication Process	Container and Resource Discovery	Taint Shared Context	Data from Cloud Storage	Fallback Channels	Scheduled Transfer	Financial Theft
Search Open Hosts/OS/Containers	Trusted Relationship	Serverless Execution	Event Triggered Execution	Event Triggered Execution	Domain Policy Modification	Multi-Factor Authentication Request Generation	Debugger Evasion	Use Alternate Authentication Mechanism	Data from Configuration Inventory	Ingress Tool Transfer	Transfer Data to Cloud Account	Firmware Corruption	
Search Victim-Owned Hosts	Valid Accounts	Shared Modules	External Remote Services	Exploitation for Privilege Escalation	Execution Guardrails	Multi-Factor Authentication Request Generation	Device Driver Discovery	Data from Information Resources	Multi-Stage Channels			Insight System Recovery	
	Software Deployment Tools	Software Deployment Tools	Hijack Execution Flow	Hijack Execution Flow	Exploitation for Defense Evasion	Network Sniffing	Domain Trust Discovery	Data from Local System	Non-Application Layer Protocol			Network Denial of Service	
	System Services	Implant Internal Image	Process Injection	File and Directory Permissions Modification	OS Credential Dumping	File and Directory Discovery		Data from Network Shared Drive	Non-Standard Port			Resource Hijacking	
	User Elevation	Modify Authentication Process	Scheduled Task/Job	Hide Artifacts	Steal Application Access Token	Group Policy Discovery		Data from Removable Media	Protocol Tunneling			Service Stop	
	Windows Management Instrumentation	Office Application Startup	Valid Accounts	Hijack Execution Flow	Steal Web Session Cookies	Log Enumeration		Email Collection	Proxy			System Shutdown/Reboot	
		Power Settings			Impair Defenses	Steal or Forge Kerberos Tickets	Network Service Discovery	Input Capture	Remote Access Software				
		Pre-OS Boot			Impersonation	Steal or Forge Kerberos Tickets	Network Share Discovery	Screen Capture	Traffic Signaling				
		Scheduled Task/Job			Indicator Removal	Unsecured Credentials	Network Sniffing	Video Capture	Web Service				
		Server Software Component			Indirect Command Execution		Password Policy Discovery						
		Traffic Signaling			Manipulating		Peripherals Device Discovery						
		Valid Accounts			Notify Authentication Process		Permission Groups Discovery						

Figure 2.11: MITRE ATT&CK matrix available in the Splunk Security Essentials dashboard with clickable items.

2.4.1 Network Service Discovery

By selecting an item from the MITRE ATT&CK matrix (Figure 2.11) provided in the analytics advisor dashboard, it is possible to access information about attacks, listed along with the mapping to data sources, tactics and techniques. Additionally, a drilldown option is available from the same page, leading to a separate dashboard displaying a list of all attacks within the specific security content, as shown in the Figure 2.12. Before searching for a detailed information about a specific attack, it is possible to review all the stages of operations required for proper detection configuration, including data collection and normalization using the Common information Model, asset enrichment, and automation and orchestration for advanced detection.

Stage 1: Collection [🔗](#)
 You have the data onboarded, what do you do first?

Security Content	Description	Searches Included	Discovery
Basic Scanning	Looks for hosts that reach out to more than 500 hosts, or more than 500 ports in a short period of time, indicating scanning.	None	Discovery, Network Service Discovery
Hosts Sending To More Destinations Than Normal	This will typically detect scanning activity, along with lateral movement activity.	Searches Included	Discovery, Network Service Discovery, Remote System Discovery
Unauthorized Connection Through Firewall	Any communication through the firewall not explicitly granted by policy could indicate either a misconfiguration or even malicious actions, putting your security and compliance at risk.	Searches Included	Exfiltration, Discovery, Command and Control

Figure 2.12: Network Service Discovery drilldown page with security content and possible attack detection

2.5 Splunk InfoSec

The Splunk InfoSec App is a powerful solution designed to support security intelligence and streamline the identification, analysis, and response to potential security threats and incidents within an organization. As an integral part of the Splunk system, this app extends the platform’s capabilities by offering specialized features and tools customized for security professionals and teams.

The primary purpose of the application is to offer a comprehensive platform for the effective management and mitigation of security risks [3]. By aggregating data from diverse sources and by displaying relevant information on the provided

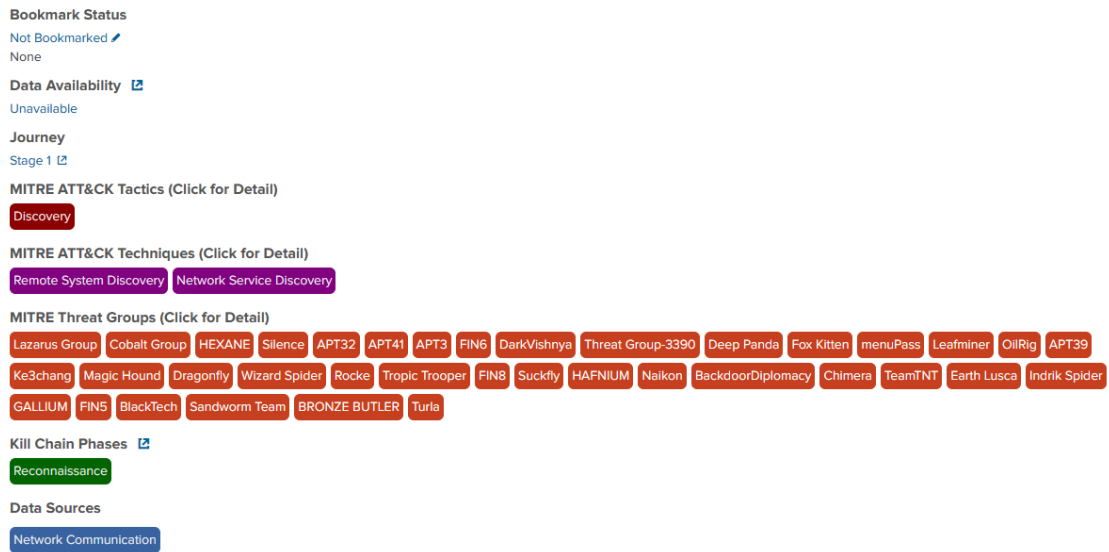


Figure 2.13: Security Content Basic Scanning page with mapping of the attack to MITRE ATT&CK tactics and techniques.

dashboards, the application allows to continuously monitor the general overview of the security of the infrastructure, and to effortlessly identify anomalies, thoroughly investigate incidents, and promptly respond to threats in real-time, all while providing an intuitive and user-friendly interface that sets it apart from more complex solutions. Beyond its core functions, the InfoSec app is designed for additional customization by offering the necessary security features while allowing users to expand the platform according to specific security requirements. This adaptability can be further enhanced by integrating additional apps and add-ons from the Splunkbase repository.

2.5.1 InfoSec and CIM integration

The Splunk InfoSec app's effective operation relies on the seamless integration of the CIM data models, which serve as the backbone that ensures a consistent, normalized, and contextualized view of the event data brought into the Splunk environment [4]. The CIM data models, in combination with accelerated data models, permit the swift correlation of data across various sources, enabling the uncovering of hidden patterns and potential security threats that might have otherwise gone unnoticed. Furthermore, the accelerated data models enhance the performance of searches and analyses by optimizing the execution of complex queries and making it possible to extract useful information from vast amounts of data in real-time.

To maximize the utility of the InfoSec app, it is crucial to ensure that both the CIM data models and the accelerated data models are configured accurately. It is possible to configure only the relevant data models based on specific requirements, but it is necessary to have a solid knowledge of the information gathered within the system. Mapping each type of log to the appropriate model is essential, since it ensures that the InfoSec app accurately interprets the data, enabling precise determination of information and facilitating the visualization of results.



Figure 2.14: Example of a Security Posture page of the InfoSec application

2.5.2 Configuration of The InfoSec App

The InfoSec documentation offers a helpful guide for installing and configuring the application accurately. Additionally, the app includes a dashboard displaying the current configuration status, showcasing the required accelerated data models and installed applications necessary for its proper functionality.

As mentioned earlier, apart from the CIM application, it is essential to install the following add-ons downloadable from the Splunkbase repository:

- Punchcard - Custom Visualization: used to provide new visualization options to display metric aggregated over two dimensions [5].
- Force Directed App For Splunk: used to visualize networks, attack paths inside an environment and connections between objects [6].
- Splunk App for Lookup File Editing: used to edit lookup files directly from the splunk web page and to maintain history and propagate modification also in a clustered environment.

By accessing the Health dashboard within the InfoSec application, is it possible to review the status of all configurations and installed add-ons to determine the readiness of the app for use and verify which data is properly configured for system monitoring. The first panel of Figure 2.15 illustrates the number of events received in the last 24 hours for each CIM data model utilized by InfoSec, while the second panel provides an overview of the data model acceleration status including a percentage of the build.

Data Models Used by InfoSec App: Events in 24 Hours		Data Model Acceleration Status	
If count = 0, check prerequisites in Help menu		All accelerated data models and their status	
data_model ↕	events ↕	data_model ↕	complete ↕
CIM_Authentication	0	CIM_Network_Traffic	100 %
CIM_Change	0		
CIM_Endpoint.Processes (optional)	0		
CIM_Intrusion_Detection	0		
CIM_Malware	0		
CIM_Network_Sessions	0		
CIM_Network_Traffic	932		
CIM_Web (optional)	0		
Network_Sessions.All_Sessions.VPN (optional)	0		

Figure 2.15: InfoSec Health Dashboard with the status of the event ingested for each data models and the built percentage of the acceleration.

On the same page, there is a dedicated panel displaying the currently installed add-ons that are necessary for the InfoSec application, along with their respective versions, as shown in the Figure 2.16.

Installed Add-ons Used by InfoSec App	
Required Add-ons: Common Information Model (CIM), Force Directed Viz, Punchcard, Lookup Editor (Optional), Sankey Diagram (Optional)	
add-on ↕	version ↕
Force Directed Visualisation App for Splunk	3.1.0
Splunk App For Lookup File Editing	4.0.2
Punchcard	1.5.0
Splunk Common Information Model	5.2.0

Figure 2.16: InfoSec Health Dashboard with the status of the application required installed and their version.

2.6 Simulating Real-World Data with Splunk Eventgen

Splunk Eventgen is a powerful application utility that enables the generation of synthetic event data within the Splunk environment. The primary purpose of Splunk Eventgen is to mimic real-world data, allowing users to simulate various scenarios and conditions without affecting actual operational systems and without configuring real external apparatus that logs information. This proves invaluable for testing the efficiency of search queries, evaluating dashboard performance, validating the accuracy of alerts and visualizations, and conducting test scenarios for developing new applications [7].

Splunk Eventgen offers a range of features that facilitate flexible and customized data generation. Users can specify attributes such as timestamps, sources, source-types, hostnames, and more, enabling the creation of data that mirrors their actual environment. Additionally, the tool supports the use of predefined sample datasets, which can be customized to match specific use cases. Eventgen offers the capability to initiate event generation using existing samples as a starting point. Moreover, it provides the flexibility to extract and replace tokens from the original sample, resulting in the creation of customized events. The randomization feature incorporated within the application introduces an element of randomness into log ingestion, providing a useful tool for simulating real-world logging scenarios.

One of the notable aspects of the application is its capability to generate data in varying volumes and at adjustable frequencies. This feature proves beneficial for load testing, capacity planning, and evaluating system behavior under different data loads. Furthermore, Splunk Eventgen's compatibility with the Splunk Common Information Model ensures that generated data aligns with standardized data models, enhancing the accuracy and usefulness of analysis.

2.7 Splunk Fortinet FortiGate Add-on

The data-set utilized as a testing environment for this thesis is based on logs generated by Fortinet FortiGate firewalls. Consequently, it becomes essential to introduce an application within the Splunk infrastructure that can effectively parse, extract, and manage the format of these specific log types.

The Splunk Fortinet FortiGate Add-on serves as a bridge between various Fortinet devices, including firewalls, IDS and IPS systems, with the Splunk platform. It allows to efficiently collect, index, and analyze security-related data from these devices, allowing for comprehensive visibility into network activities, threats, and vulnerabilities [8]. The application provides different functionalities:

- **Log Data Collection:** the application is designed to gather log data from

FortiGate firewalls, ensuring that every relevant event is captured for further analysis. This includes a wide range of security-related logs, traffic data, system events, and more.

- **Data Parsing and Normalization:** upon collection, the add-on parses the raw log data, extracting valuable information and fields. It then normalizes these fields according to the CIM, ensuring consistent data structure and enabling efficient querying and correlation with other data sources.
- **Field Extractions:** the add-on automatically extracts relevant fields from the log data enriching the events with context and metadata, enhancing the subsequent analysis.
- **Dashboard and Visualizations:** the FortiGate Add-On includes pre-built dashboards and visualizations that are adapted to each type of log data. These visual representations provide visibility into network traffic, security incidents and system health, allowing the identification of anomalies and potential threats.

Chapter 3

Design of the Splunk Application

In this chapter it is going to be described the process and experiments conducted to develop an application centered on the detection of network anomalies. Based on the extensive tools and knowledge presented in the previous chapter, the focus will be on the practical application of these concepts.

The experimental analyses carried out in this chapter cover a wide variety of topics, beginning with the preparation of data and going through the process of selecting algorithms and implementing the application. All these configuration and testing processes will lead to the creation of a simple yet comprehensive tool for the detection and investigation of network anomalies. The queries presented will be incorporated and customized as part of the final application, serving as alerts and dashboard visualizations. The final solution also includes the configuration files to continuously generate traffic events, the alerts created to detect the different network anomalies, and a dashboard with various information related to the communications paths, the geographical areas and the attacks detected.

3.1 Data Generation

In the initial part of the experimentation, the focus is on the generation of synthetic network events. Understanding the format of Fortigate sample events is crucial for creating meaningful test data. It is also explored the configuration settings of Eventgen, a tool that aids in the generation of log data which are essential for simulating normal traffic and various network attacks. Two distinct log types generated by Fortinet Fortigate devices will be created, described, and later normalized to prove how the utilization of data models enables the detection of anomalies across diverse data sources.

3.1.1 Fortigate Sample Events Format

Fortigate Fortinet firewalls are able to detect and log various information about the network traffic data that come across these devices. These events are crucial for monitoring network activities, identifying threats, and ensuring the overall security of an organization's digital infrastructure. Within these events, FortiGate sample data conform to a structured format, that must be followed to generate real and meaningful information. The Fortinet devices have the capability to produce logs that include both traffic events, which describe communication between source and destination addresses and traffic paths, and events related to Unified Threat Management (UTM) and Intrusion Detection System (IDS) functionalities.

Fortigate traffic logs format

The following is an overview of the key components typically found in Fortigate traffic sample events:

- **Timestamp:** each event includes a timestamp indicating when the event occurred. This timestamp is essential for tracking the timing of network activities and security-related events.
- **Device Information:** Fortigate sample events contain device-specific information, such as the device name (e.g., Fortigate firewall), a unique device identifier, and relevant device properties. This information helps in associating events with specific devices in a network.
- **Event Type and Subtype:** Events are categorized into various types and subtypes, allowing for easy classification and analysis. For example, events can be classified as "traffic" with subtypes like "forward," "deny," or "session."
- **Network Details:** Fortigate events provide detailed network information, including source and destination IP addresses, source and destination ports, and the protocol used (e.g. TCP, UDP).
- **Policy and Session Information:** Events often reference security policies and sessions associated with the network traffic. This information assists in determining whether network activities conform to established security policies or require further investigation.
- **Action Taken:** Fortigate events specify the action taken in response to network traffic. Common actions include "allow," "deny," or "reset." This information is crucial for evaluating the security posture of the network.

- **Additional Details:** Depending on the specific event, Fortigate sample events may include additional details related to the event, such as the number of bytes transmitted, service details, and any custom attributes configured in the firewall.
- **Event Message:** A descriptive event message or log message provides context for the event. It often includes information about the event trigger, potential security risks, or other relevant details.

Fortigate UTM logs format

An UTM event example shares common fields relevant to Fortinet devices such as source and destination information, timestamp and device name. However, it includes additional details such as the severity of events, unique identifiers assigned to incidents, and correlation score and level. These additions help define the event's severity and its correlation with known patterns or rules within the security system.

The following are some example of additional fields present in a Fortigate UTM log:

- **attack info:** name of the attack detected by the UTM and the corresponding id.
- **profile:** in most of the cases it is present a profile of the attack detected and a reference to the official Fortinet web page, where a detailed description of the detection is showed.
- **incident number:** usually a detection of a security event can triggers the creation of an incident in a dedicated service. In this field it is included the number of the incident related to the event logged.
- **threat score and level:** to the IPS signatures, web categories, malware, and applications are all assigned a severity that is associated with a threat weight, which can be calculated as the value of the category of the attack times the number of incident detected. The severity of the detected incident increases proportionally with the threat score value.

Understanding the format and structure of Fortigate sample events is essential for effective log analysis and security monitoring within the Splunk environment. These events serve as the foundation for detecting network anomalies, identifying security incidents, and responding to potential threats. In the subsequent sections, are going to be explored how the configuration of Splunk Eventgen to create these events starting from some sample data.

3.1.2 Eventgen Configurations

An essential task of the present research is the creation of network traffic behavior that closely resembles authentic patterns, which begins with a collection of extracted Fortigate events. This task necessitates to correctly configured the Eventgen application, to derive original logs that closely emulate authentic network traffic. To ensure the precision and fidelity of these synthetic events, a prerequisite step involves the creation of a structured CSV or TXT file containing a subset of sample events. This file is subsequently positioned within the designated 'sample' directory within the application.

The selection of this sample dataset has been informed by an authoritative source: the official Splunk BOTS GitHub repository [9], which features a repository of exemplary events including both real-world cyber-attacks and normal network traffic. The acquisition of this dataset involved its retrieval from the repository, followed by its ingestion into a Splunk environment. Subsequently, an extraction of sample events was performed to serve as the foundation for the event generation process.

The configuration of Eventgen revolves around the specification of temporal parameters that dictate the temporal aspects of the event generation process.

The Parameters must be defined in a dedicated text file called eventgen.conf included in the application default folder and automatically recognized by the Eventgen Add-on.

```
[fgt_traffic_sample.csv]
disabled = false
mode = sample
interval = 30
earliest = -30s
latest = now
count = 35
randomizeCount = 0.2
randomizeEvents = true
```

Listing 3.1: eventgen.conf file with the configuration for traffic sample events generation.

stanza definition The initial line specifies the filename, situated within the 'sample' directory. This file serves as the foundational dataset guiding subsequent configurations.

disabled The 'disabled' parameter is set to 'false,' enabling the event generation process specifically for this dataset.

mode Eventgen offers two distinct modes: 'sample' designates the utilization of a sampled dataset, where random samples are extracted from the file. Alternatively, 'replay' mode replays events from the file in their original order.

interval The 'interval' configuration determines the frequency of event generation. For example it can be set at thirty seconds intervals.

earliest and latest The 'earliest' and 'latest' configurations define a temporal interval for the timestamp of the events. An example can be a time range from the past thirty seconds up to the current moment.

count The 'count' parameter specifies the total quantity of synthetic events generated during each execution of the generation process. At each iteration defined by the interval parameters a certain quantity of events are generated together. Changing this configuration allows to increment or decrement the volume traffic and to simulate different kind of scenarios.

hourOfDayRate This configuration defines the likelihood of event generation for each hour of the day. The values specified for each hour, ranging from 0 to 23, represent the probabilities (ranging from 0 to 1) that an event will be generated during that specific hour. For instance, during the early morning hours (0-5), the probabilities are relatively low, reflecting a reduced likelihood of network activity. In contrast, the probabilities increase during business hours (8-17), signifying heightened network traffic during this period. This parameter allows for the replication of daily network patterns and the simulation of event occurrence at various times of the day, enhancing the authenticity of the generated events.

dayOfWeekRate This parameter controls the distribution of events across different days of the week. It specifies the likelihood of generating events on each day, from Sunday (0) to Saturday (6), using probability values ranging from 0 to 1. This configuration enables the emulation of variations in network traffic patterns over the course of a week. For instance, a higher probability for weekdays (1-4) suggests increased network activity during business days, while lower probabilities for weekends (5-6) indicate reduced traffic during the weekends.

dayOfMonthRate The parameter extends the temporal control by facilitating the distribution of events throughout a month. Similar to the previous parameters, it assigns probability values to each day of the month, ranging from 0 to 1. This configuration permits the recreation of monthly patterns in network activity. In particular, higher probabilities on specific days (e.g., the 8th and 15th) may

correspond to regular network maintenance or billing cycles, while lower probabilities on others (e.g., 3rd and 30th) indicate periods of reduced activity.

randomizeCount The following parameter introduces randomness into the number of events generated. It accepts a float value that represents a percentage. Setting `randomizeCount` to 0.2 indicates a 20% randomization in either direction (an increase or decrease) in the number of events generated. This means that, on average, there is a 20% chance that the actual number of generated events may deviate from the expected count. This randomness adds variability to the event generation process, simulating scenarios in which the frequency of network events may fluctuate unpredictably.

randomizeEvents This parameter, when configured as 'true', activates the randomization of event generation within the provided sample file. When working with sample files containing a considerable variety of distinct events, this setting ensures that events are selected randomly for generation during each execution.

outputMode This setting defines the output mode, indicating that Splunk will utilize a modular input mechanism to ingest and manage the generated events. Modular inputs are versatile components in Splunk that enable data collection from various sources. This choice of output mode ensures seamless integration of the generated events into the Splunk environment.

index Specifies the target index where the generated events will be stored within the Splunk instance.

host This configuration identifies the host or source of the generated events. In this context, it indicates that the events originate from the specified IP address.

source Specifies the source of the data. The User Datagram Protocol (UDP) port 514, is the one typically used by the Fortigate devices for syslog data.

sourcetype This parameter assigns a categorization tag to the incoming data. Categorizing data using sourcetypes is essential for applying appropriate parsing rules and data transformation processes. It allows Splunk to understand the nature and format of the data, making it easier to extract meaningful information during indexing.

```
token.0.token=^[A-Za-z]{3}\s\d{2}\s\d{2}:\d{2}:\d{2}
token.0.replacementType = timestamp
token.0.replacement = %b %d %H:%M:%S

token.1.token=date=(\d{4}-\d{2}-\d{2})
token.1.replacementType = timestamp
token.1.replacement = %Y-%m-%d

token.2.token=time=(\d{2}:\d{2}:\d{2})
token.2.replacementType = timestamp
token.2.replacement = %H:%M:%S
```

Listing 3.2: eventgen.conf file with token configuration to replace the timestamp when generating new events.

The provided configuration settings are responsible for substituting the timestamps within sample logs with new timestamps falling within the interval defined by the "earliest" and "latest" parameters. These settings consist of tokens, each including a regular expression pattern to capture the timestamp section in the logs and a replacement configuration specifying the format for the newly generated timestamps.

With tokens it is also possible to replace internal IP addresses found within the source and destination fields of the original events. These settings enable the substitution of the addresses captured by the regular expression with randomly selected IP addresses sourced from a designated file. The path of the file must be correctly configured depending on the corresponding version of the operating system.

3.1.3 Simulation of Network Attacks

In the preceding section, it has been established the configuration necessary for replicating generic network traffic that closely mirrors real-world scenarios. However, for a correct evaluation of the application, it is required to introduce simulated network attacks that can be concealed within a regular network traffic. Achieving this objective requires the accurate configuration of Eventgen to generate these attacks based on predefined templates. Within the scope of the study, the aim is to simulate two distinct types of attacks: network scanning attacks and Command and Control attacks. While hidden in normal network traffic, those attacks present specific patterns that necessitate precise detection by the application.

Network Scanning Attack

The first type of attack, the network scanning attack, involves malicious actors systematically probing network resources, seeking vulnerabilities, and identifying potential targets for exploitation.

This simulation will involve the generation of traffic patterns that resemble a scanning operation, making it appear as a routine network activity. To simulate network scanning attacks effectively, Eventgen must be configured to generate traffic patterns and event logs that can be compared to the behavior of real attackers. Within the network scanning attacks it is possible to recreate two different type of attack: the host discovery where the attacker is trying to detect all the active host inside the network and their addresses, and the port scanning attack where the intruder tries to discover open and closed ports and the services listening to it.

This configuration includes:

1. **Traffic Patterns:** define traffic patterns that mimic the sequential probing of IP addresses, ports, and services, commonly associated with network scanning. These patterns should emulate the gradual exploration of the network, making it appear as if a legitimate user is engaging in routine network activity.
2. **Randomization:** incorporate randomization within the scanning patterns to introduce variability, mirroring the unpredictability of actual scanning activities. This ensures that the generated traffic does not follow a predictable pattern that could be easily detected.

Host Discovery

The configuration showed in the Listing 3.3 generates events where the attacker's static IP address remains constant, while a variety of potential addresses are utilized as destinations. This setup simulates an exploration of the network and its hosts, to gain knowledge of the active devices.

```
interval = 20
earliest = -10s
latest = now
count = 1

token.3.token=srcip=(\d+.\d+.\d+.\d+)
token.3.replacementType = static
token.3.replacement = 203.0.113.100

token.4.token=dstip=\d+.\d+.\d+(\d+)
token.4.replacementType = random
token.4.replacement = integer[1:254]

token.5.token=dstport=(\d+)
token.5.replacementType = static
token.5.replacement = 80
```

Listing 3.3: eventgen.conf file with token configuration of the host discovery attack events

Generating events with a specific time interval between each occurrence is important to mimic the attack's intensity accurately. A longer time interval between events reflects a slower-paced attack, making it more challenging to detect and distinguish from typical network traffic.

By using an relatively short interval in the Eventgen settings is possible to recreate a low intensity host discovery attack. Scanning the simulated local network 192.168.224.0/24 with 255 distinct possible hosts indicates that the attack will likely require approximately 1 hour and 20 minutes to complete, considering an interval of 20 seconds between each scanning attempt. Reducing the interval will lead to the generation of an high intensive scanning attack that can be more easily detected, since it will generate a large volume of traffic data in a small period of time.

Port Scan Attack

To configure Eventgen to generate events that resemble a port scan attack it is sufficient to utilize similar settings as the ones provided for the host discovery attack, with a reduced interval between the generation of the logs and a dynamic generation of the port number as well as the IP address of the destination. This settings recreate an exploration of the network and its services, probing the ports that are generated to identify active services and their availability. The number of the event generated is increased since the port scanning attack usually require the

testing of numerous port and destination addresses at the same time to effectively discover useful information about the targeted system.

Command and Control Attack

The Command and Control attack, is designed to replicate the covert communication channels established by malicious actors to control compromised systems within a network [10]. These communications often follow a predetermined pattern that might go unnoticed by conventional monitoring systems. The application's effectiveness depends on its ability to recognize and flag such patterns as network anomalies. By simulating Command and control attacks within the normal traffic, is it possible to assess the application's capability to distinguish between legitimate network activity and illicit command and control communications.

To achieve effective simulation of command and control attacks, Eventgen has been configured to facilitate communication between two devices: one acting as the attacker and the other as the victim. The simulated attack is characterized by its high intensity, involving a rapid influx of commands directed towards the victim.

The attack is simulated by the generation of one event each 30 seconds, which indicates a almost regular pattern of communication. Additional randomization is introduced in the generation of the timestamp of the event to replicate a more realistic behaviour. In the Eventgen configuration, it has been designated a fixed source IP address to represent the attacker and a static destination IP address for the victim, ensuring the consistency and controlled nature of the simulated C2 attack.

3.2 Model and Data Exploration

Before examine the experimental analysis, it is necessary to ensure that the data is appropriately structured and normalized. In this section, it is explained how the configuration of the Common Information Model and the acceleration are performed to increase the performance. It is also explored the data set to gain insights into its content and characteristics.

3.2.1 Configuration of CIM Network Traffic Data Model

As mentioned in the previous chapter, the Common Information Model serves as a well-established framework designed to bring standardization to the realm of information security and network management data. The requirement in the context of this research is the standardization and structuring of Fortigate events created through the use of the Eventgen application. This normalization process

is fundamental as it establish the basis for enabling seamless integration with the InfoSec application, ensuring its effective and accurate functioning.

Fortigate, as a versatile security solution, generates a wide variety of event data, often with varying formats and structures. The challenge lies in normalizing this diverse array of events into a unified format that adheres to CIM standards. This process involves mapping Fortigate-specific data fields to their corresponding CIM counterparts and ensuring the correct data types and naming conventions are maintained.

Thankfully, a significant portion of the normalization process is facilitated by the Fortigate Fortinet Add-On, which can be easily installed as an application within the Splunk Enterprise environment utilized for the event generation. This application not only provides the parsing and extraction of diverse fields within a typical event, but also adheres to the Common Information Model standards by renaming and evaluating certain fields to align with the requirements of a CIM data model.

To establish the correct connections between Fortigate data sources and the Network Traffic Data Models, it is required to make adjustments to the configuration of the CIM application integrated within the Splunk environment. Within the application's settings, there is an option to specify, for each data model, the indexes that should be queried to locate CIM-compliant data. Additionally, it provides the flexibility to customize certain parameters to optimize the acceleration of the data model.

Acceleration of Network Traffic Data Model

The acceleration of the data model discussed in the preceding section is essential for both the efficiency of data retrieval from the index and the correct integration with the Splunk InfoSec application. It is also necessary to create alerts the can frequently search for a large volume of data and monitor constantly the infrastructure.

Data model acceleration involves pre-computing and storing summary data that accelerates search and reporting processes. For the Network Traffic Data Model, this means aggregating and indexing relevant data to provide faster query responses when investigating network traffic patterns and anomalies.

To accelerate the Network Traffic Data Model effectively, it's crucial to identify and select the appropriate summary fields. These are the data attributes that will be pre-computed and indexed for faster retrieval. Common summary fields for network traffic data include source IP, destination IP, port numbers, protocols, and event counts.

Configuring Acceleration Settings

Splunk provides configuration options for accelerating data models through the user interface. To configure acceleration for the Network Traffic Data Model it is necessary to navigate to the Data Model Editor within Splunk, which allows any user with enough privileges, to manage data models and their acceleration settings. From this page it is possible to enable the acceleration by selecting the relevant data sources and summary fields that should be accelerated. It is important to specify the frequency at which the acceleration summary should be built or updated, which depends on the data rate of change and the need for up-to-date results. Additionally it is necessary to define the time range for which accelerated data should be generated. It is important to balance the range between historical and real-time data depending on the use case, since a larger summary range value can enhance performance for older data searches, but it comes at the cost of increased disk space utilization and longer build times.

For the purposes of this research, querying data dating back up to one month is sufficient. Therefore, an appropriate summary range value for this scenario is exactly one month.

3.2.2 Exploring the Generated Events

Exploring and analyzing data is a critical aspect of any data-driven endeavor, especially in the context of network traffic data generated and collected for security and operational purposes. After completing the configuration of the Data Model, there is the possibility to explore the ingested data in two different ways: common searches and leveraging the acceleration of the Network Traffic Data Model.

Common Search

Common searches in Splunk allow for ad-hoc queries and exploration of data without the need for predefined data models. This approach is flexible and useful for quick investigations. To search for a particular data set is sufficient to know the index where the logs are stored and usually the sourcetype or directly the source file name.

By using a simple search command in a determined time range is it possible to access all the raw logs contained in the chosen index, along with all the fields extracted at index and search time. Starting from the search command it is then possible to filter and analyze the various logs, as well as compute statistics based on the individual fields contained within these events.

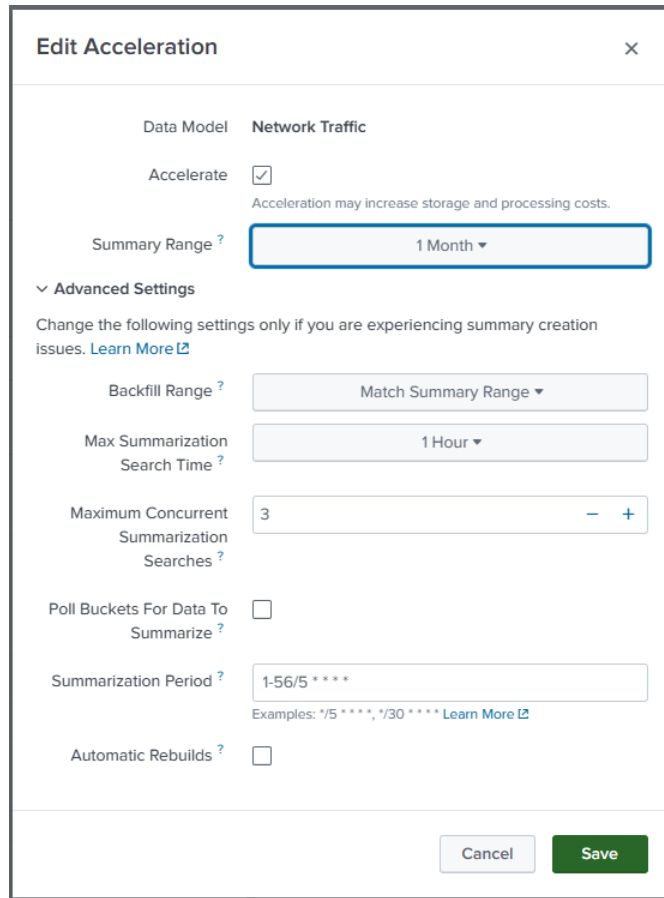


Figure 3.1: Edit of the acceleration settings of the Network Traffic Data Model

Search with the Data Model

The configured Network Traffic Data Model offers a structured and optimized approach for exploring network traffic data. It simplifies and accelerates the process of uncovering valuable insights. To use the data model to extract the logs contained within, it is possible to use the generative command `| from datamodel` that is able to create events starting from the information kept inside the specified model.

Exploiting the data model within the search query notably improves the extraction efficiency of logs. However, it is important to note that only the fields included in the data model are extracted and made accessible for analysis, while any other fields are not displayed and cannot be used for further event processing.

Additionally, with accelerated data models it is commonly used the `tstast` command which is able to perform statistics in a similar way to the normal `stats` command, but with the ability to perform the operations on indexed fields in `tsidx` files. Since it searches only on indexed fields instead of the raw events, it is faster to

execute and it is able to exploit the acceleration functionalities of the data models.

Search Performance

To see the performance of the query is it possible to select the job inspector view provided in the Splunk web interface after a search is completed. In the Figures 3.2 and 3.3 are displayed the timing performance of the two query previously described. The command that exploited the functionality of the data model is more than two second faster while extracting the same amount of events. The job inspector computes and shows all the timing of each command within the search query, allowing to detect the part of the SPL searches that required more time to execute, and to improving the performance by indicating the optimized search.

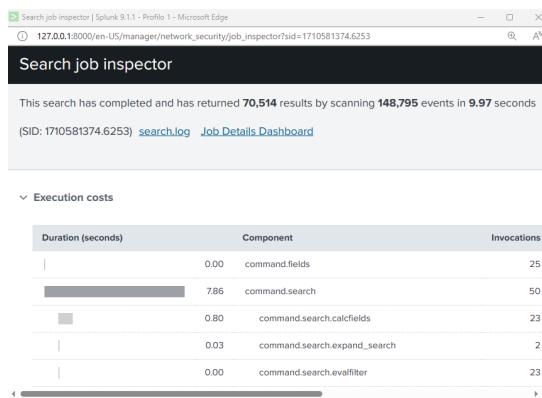


Figure 3.2: Job inspector result of the query that search the events with source-type fortigate_traffic directly from the index. The search required 9,97 seconds to scan 148795 events.

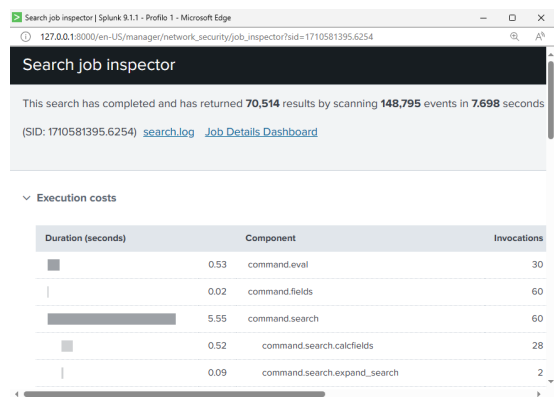


Figure 3.3: Job inspector result of the query that search the events with source-type fortigate_traffic exploiting the Network Traffic data model. The search required 7,698 seconds to scan 148795 events.

3.3 Experimental Searches and Analysis

In this section are going to be presented the analytical techniques adopted to detect anomalies and security threats within the network traffic data. Various aspects have been addressed, including outlier detection, the identification of network scanning activities, and the detection of command and control attacks. Additionally, it has been correlated traffic data with geo-spatial information and created a query able

to display data in a Network Communication Map using the tools provided by the InfoSec application.

3.3.1 Detect outliers in Network Traffic

In the realm of network security and performance monitoring, the ability to detect outliers within network traffic data can ensure the safeguarding of the networks and their optimal operation. Outliers, in this context, represent data points or network behaviors that deviate significantly from the norm. These anomalies can range from sudden surges in data traffic to unusual patterns of communication, and they often signify potential security threats, operational issues, or even opportunities for optimization.

The techniques used for detecting outliers in network traffic encompass statistical analyses, pattern recognition, and data modeling. By meticulously examining the statistical distribution of network parameters and evaluating historical trends, it is possible to detect anomalies that might otherwise evade human observation.

The algorithm 1, which is pseudo code for the SPL queries adopted in the solution, is specifically designed to identify unusual spikes in the count of destination IP addresses observed within the past 24 hours. This analysis is valuable in monitoring network traffic and identify potentially anomalous behavior.

The first step is to summarize and retrieve statistics from the Network Traffic Data Model extracting events in the last thirty days, and then calculate the distinct count of destination IP addresses within one day time spans for each source IP address.

After the extraction of the events, some statistics and variable are computed:

- **maxtime:** contains the value of the current time, at the moment the searches is run. Used as a baseline to distinguish all the events that are older than 24 hours.
- **number of data sample:** counts the number of data samples in the searched range for each source IP to exclude events that have few samples and for which the outliers detection can cause the generation of false positive.
- **maximum count of destination:** finds the maximum count of destination IP addresses within the last day.
- **average count of destination:** calculates the average count of destination IP addresses before the last day.
- **stdev:** computes the standard deviation of the count of destination IP addresses before the last day, providing a measure of data variability.

Algorithm 1 Pseudo code of the SPL query to detect outliers in network traffic data.

```

1: ▷ Retrieve summarized statistics of all traffic, counting unique destination IP,
   grouped by source IP with a time span of 1 day
2: get events from data model
3: minsample ← 8 ▷ Set the minimum number of events needed to consider it an
   outliers
4: _time ← event timestamp ▷ Timestamp of each event
5: Group By source IP, _time
6: for each source IP do
7:   dc = distinct count destination IP
8: end for
9: maxtime ← now() ▷ get current timestamp
10: lastday = maxtime − 24 hours ▷ timestamp of the last day
11: Group By source IP
12: for each source IP do
13:   count = tot number of events
14:   max = maximum dc where _time ≥ lastday
15:   average = average dc where _time < lastday
16:   stdev = standard deviation of dc where time < lastday
17: end for
18: lowerBound = (average − 2 × stdev) ▷ calculate lower bound
19: upperBound = (average + 2 × stdev) ▷ calculate upper bound
20: ▷ Identify outliers
21: if max > upperBound AND count > minsample then
22:   isOutlier = True
23: else
24:   isOutlier = False
25: end if
26: where isOutlier = True ▷ Filter out records which are outliers

```

To identify values that deviate significantly from the norm an upper and lower bounds are calculated. Those values are retrieved from the previously calculated average and standard deviation. Then, it is possible to evaluate whether the current count exceeds the upper bound and whether there are a sufficient number of data samples to consider it an outlier. If both conditions are met, a variable is set and later filtered by, in order to extract only the relevant events.

By executing the previous search, it is possible to retrieve only the outliers that are present in the events of the last 24 hours, and create a table to show those values. By categorizing the search results based on the source IP address, it is possible to observe whether the outliers with the highest count originate from an internal server or an external one, and possibly continue the investigation searching other information or events related to that particular source.

source IP	count	avg	Upper Bound
192.168.250.160	242	59.784	138.273
192.168.225.111	17	7.926	14.875
192.168.225.96	14	4.423	10.494
192.168.229.283	11	4.042	9.462
192.168.224.71	10	2.043	5.627
192.168.229.237	9	2.000	5.240
192.168.230.27	9	2.258	4.727
192.168.224.8	8	1.667	3.201
192.168.224.96	8	1.667	3.200
192.168.225.46	8	2.286	5.053
192.168.225.6	8	1.857	3.886
192.168.229.253	8	2.007	4.709
192.168.230.17	8	1.812	3.775
192.168.224.3	7	2.000	4.275
192.168.224.38	7	1.786	3.545
192.168.224.44	7	1.667	3.618
192.168.224.65	7	1.786	3.249
192.168.224.7	7	1.611	3.795
192.168.229.252	7	1.684	3.183
192.168.230.18	7	1.933	3.701

Figure 3.4: Table representation of the result of the search to detects outliers

3.3.2 Detect Network Scanning Activities

Network scanning activities are commonly used in the initial reconnaissance step that cyber attackers often employ to map out vulnerabilities, identify weak points, and gain information that can be later used to launch more sophisticated attacks towards the target [11]. Detecting these probing activities can avoid potential security breaches and safeguarding network integrity in the future.

Similar to the previous search, this analysis requires the extraction of information directly from the Network Traffic Data Model. The command computes two essential statistics for each source IP address: the unique count of destination ports and the number of IP addresses contacted by that source IP. Based on this data, the subsequent calculations reflect those used to identify outliers in the count of destination, with the addition of evaluating the average, count, and standard deviation for destination ports.

To detect a possible scanning activities, we filter the results to identify source IP addresses that are outliers and have contacted more than 100 distinct destination

ports or more than 100 distinct destination IP addresses. These thresholds are set to capture behavior indicative of network scanning, as a high number of destination ports or IPs suggests a systematic exploration of network assets. It is always possible to adjust those threshold, depending on the size of the infrastructure monitored and the volume of traffic data, in order to customize the control and reduce the generation of false positive and false negative.

By running the search using the simulated data generated through Eventgen's configuration, it was successfully identified the attacker's IP address as a potential indication of network scanning activity. This finding opens the door to further investigation by analyzing the logs in more detail.

3.3.3 Detect Command And Control Attacks

Command and Control attacks are often more challenging to detect compared to other types of attacks due to their complex and stealthy nature. Unlike some straightforward attacks that may exhibit clear patterns or signatures, these attacks are designed to mimic legitimate network traffic, making them harder to identify. With access to network traffic data alone, it remains feasible to uncover potential malicious activities through the analysis of communication patterns, all without directly inspecting the actual content or data exchanged between the communicating parties [12].

The search designed to detect those types of attacks uses the suggested configurations and thresholds specified in the InfoSec and Security Essential applications. Its primary objective is to identify simple Command and Control attacks characterized by an almost regular pattern of communication with their targeted victims. The approach is based on detecting some anomalies and traits commonly linked to Command and Control traffic.

Some of them are:

- **Low volume of data:** C2 attacks often generate a relatively low volume of traffic over an extended period. This method helps them hide within regular network activity, making it difficult to distinguish malicious behavior from normal patterns.
- **Short Time Gaps:** there are minimal time gaps between communication events in C2 traffic. This can manifest as very short intervals between data exchanges, reflecting the urgency and responsiveness of the C2 communication.
- **Consistent Communication Patterns:** C2 traffic tends to follow consistent communication patterns over time. The commands and responses may exhibit regular intervals or predictable sequences, which is distinct from the variability seen in legitimate network traffic.

The algorithm 2 describe the behaviour of the search adopted to detect command and controls attacks. By starting from the network data present in the Network CIM Data Model, the search calculates the time gap between the current event and the last event for each destination IP address. This gap represents the time elapsed between two successive communications.

The algorithm then performs statistical calculations on the data grouped by source (attacker) and destination (server) IP addresses. It computes metrics relative to the number of events between a source and destination IP, the average time gap between communications for each source and destination pair and the variance of time gaps, providing insight into the variability of communication timing.

To identify potential C2 activities, the search filters the results based on the previously computed values. When the average time gap between communications falls below a certain value, it could denote the presence of C2 communication. Additionally, the requirement for a high event count ensures that a significant volume of communications has taken place between the source and destination IP addresses, further suggesting the possibility of C2 activity.

To potentially exclude hosts that periodically and frequently communicate with each other, which might be wrongly detected as potential attacks, a lookup table is utilized. In the table all the known addresses that frequently communicates between each other can be included to exclude all the events which refer to those communications. By checking if both the destination and the source are found in the provided lookup, it becomes possible to exclude these events and prevent the triggering of unwanted alerts.

When running this search with the generated data, it was possible to successfully detected the high-intensity simulated attack. Depending on the normal traffic volume monitored by the Splunk infrastructure, lowering the event count threshold for faster detection of potential attacks or increasing the average gap between events can lead to the identification of false positives that resemble normal traffic patterns. However, these adjustments also provide greater flexibility to the recognition of low-intensity attacks which may present an higher value of time gap between subsequent communications to possibly elude the search detection.

3.3.4 Correlation of Traffic Data with Geo-Spatial Information

Splunk provides a powerful capability to correlate network traffic data with geo-spatial information, allowing to create graphical maps with the geographic origins and destinations of network activities.

The search described in the algorithm 3 begins by retrieving summarized network traffic data from the Network Traffic data model. It focuses on events where the action is categorized as either "block" or "drop," which typically represent network

Algorithm 2 Pseudo code of the SPL query to detect possible command and control attacks in network traffic data.

```
1: ▷ Retrieve events of all traffic in the last month
2: get events from data model
3: _time ← event timestamp                                ▷ Timestamp of each event
4: for each destination IP do
5:   next_time = time next event with same destination
6: end for
7: gap = _time – next_time    ▷ time difference between the current event and
   the next event
8: Group By source IP, destination IP
9: for each unique combination of source IP and destination IP do
10:  count = tot number of events
11:  avg_gap = average time gap
12:  var_gap = variance of time gaps
13: end for
14: ▷ Filters records where the average time gap is less than 50 and the count is
   greater than 300
15: where avg_gap < 50 AND count > 300
16: ▷ add information from lookup table to understand if addresses are known or
   not
17: src_is_known ← lookup value [True, False]
18: dest_is_known ← lookup value [True, False]
19: ▷ Discard records where both addresses are known
20: where NOT (src_is_known AND dest_is_known)
```

security events where data packets are blocked or dropped. To enrich the data with geo-spatial information, the "iplocation" command can be used to determine the geographic location (latitude and longitude) associated with the source IP addresses. Splunk extracts location information from IP addresses by using 3rd-party databases [13].

Finally, the "geostats" command is employed to perform geo-spatial statistics based on the destination ports: it calculates the sum of counts for each destination port, providing an overview of the total network traffic activity associated with each port and allows to display values in a cluster map.

Algorithm 3 Pseudo code of the SPL query to extract geographical information of blocked traffic by destination port in network traffic data.

```
1: ▷ Retrieve summarized statistics of all traffic, counting events grouped by
   source IP and destination port
2: get events from data model
3: _time ← event timestamp                                ▷ Timestamp of each event
4: ▷ Filters events where action is either "block" or "drop"
5: where action = block OR action = drop
6: Group By source IP, destination port
7: for each combination of source IP and destination port do
8:   count = tot number of events
9: end for
10: ▷ Get location information of source IP from Splunk internal database
11: location ← geolocation from source IP
12: Group By destination port
13: for each of destination port do
14:   sum = sum of event count
15: end for
```

By providing geographical data the platform can present this information visually on a map interface. This functionality enables users to efficiently locate the collective geographic positions of the blocked or dropped IP addresses. This visual representation assists in identifying geographical patterns, clusters, or concentrations of these restricted IP addresses, offering a powerful tool during investigations on the sources of suspicious network activities.

3.3.5 Network Communication Map

When investigating a potential network attack, it is important to gain knowledge of the various paths and connections established by an infected device. The InfoSec application offers valuable features that can be integrated into a dashboard to help

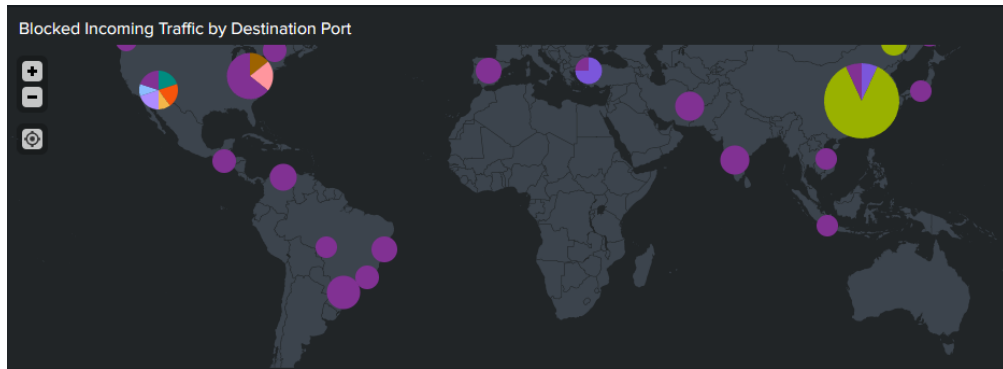


Figure 3.5: Cluster map visual representation of the blocked traffic by destination port. The panel is part of the application created with this thesis research.

IT Operations administrators and the security team in visualizing and analyzing potential attacks and their behavior through different connections within the network .

Applications Panel

The application provides the capability to filter panels and maps to focus on relevant data while excluding unwanted information. To display details about the different applications and protocols involved in filtered network events, the "eventstats" command and the field application computed by the data model can be utilized in the search query. The "eventstats" command allows to compute aggregated statistic, such as the distinct count of destination by source, without aggregating the results.

The search query is able to retrieve summarized network traffic data, focusing on events where the application field is not empty, and the action field is present. It also considers events with valid source and destination IP addresses. Then it calculates the count of unique destination IP addresses for each source IP, and it filters the results to include only source with at least one unique destination. Finally, it calculates the sum of communication counts by application, sorts the results by communication count in descending order, and presents the information in a table format with columns for the application name and the number of communications.

By examining the table of the Figure 3.6 it is possible to identify the most commonly used applications and protocols, spot outliers or unusual activities, and assess the overall network service landscape.

app	communications
Microsoft.Portal	3634
DNS	2708
SSH	454
TELNET	307
NetBIOS.Name.Service	102

Figure 3.6: Table visualization for the Application Panel within the Network Anomalies Dashboard.

Force Directed Visualization Graph

The Force Directed Visualization Graph is a graphical representation used to visualize and understand the connections and relationships between various entities, such as IP addresses, within a network. This type of graph is particularly useful for illustrating the network’s topology and the interactions between different devices or hosts. The dynamic graph uses force-directed layout algorithms to position nodes (IP addresses) and edges (connections) in a way that minimizes overlapping and provides a clear visualization of network relationships.

The provided algorithm 4, populates the Force Directed Visualization Graph within the InfoSec application.

The search begins by retrieving summarized network traffic data from the data model. The query filters out events where the source and destination IP addresses are either both "unknown" or are the same, as these are not relevant for visualization. Then it calculates the count of unique destination IP addresses for each source IP address: this count is useful for assessing the degree of communication from each source. At the end, the query aggregates the data by counting the number of events, which represent the communication occurrences, between each source and destination IP address pair. To prevent overwhelming the visualization, it limits the results to the top 250 source-destination pairs with the highest communication counts.

The graph in the Figure 3.7 visually depicts the network topology and how devices are interconnected. Analysts can identify clusters of devices communicating frequently, detect anomalous or unexpected connections, and assess the overall structure of the network.

Algorithm 4 Pseudo code of the SPL query to extract communication paths between source and destination in network traffic data.

- 1: \triangleright Retrieve summarized statistics of all traffic, counting events grouped by source IP and destination port
 - 2: *get events from data model*
 - 3: $_time \leftarrow$ *event timestamp* \triangleright Timestamp of each event
 - 4: \triangleright Filters events where source and destination IP addresses are either both "unknown" or are the same
 - 5: *where NOT(dest = src OR dest = "unknown" OR src = "unknown")*
 - 6: *Group By source*
 - 7: **for** each source **do**
 - 8: *host_count = distinct count destination IP*
 - 9: **end for**
 - 10: \triangleright Filter out any sources with zero unique destinations
 - 11: *where host_count \geq 1*
 - 12: *Group By source, destination*
 - 13: **for** each of combination of source and destination **do**
 - 14: \triangleright Compute the communication occurrences between each source and destination IP address pair
 - 15: *count = sum of event count*
 - 16: **end for**
 - 17: *top 250 events* \triangleright Limit the results to the top 250 source-destination pair
-

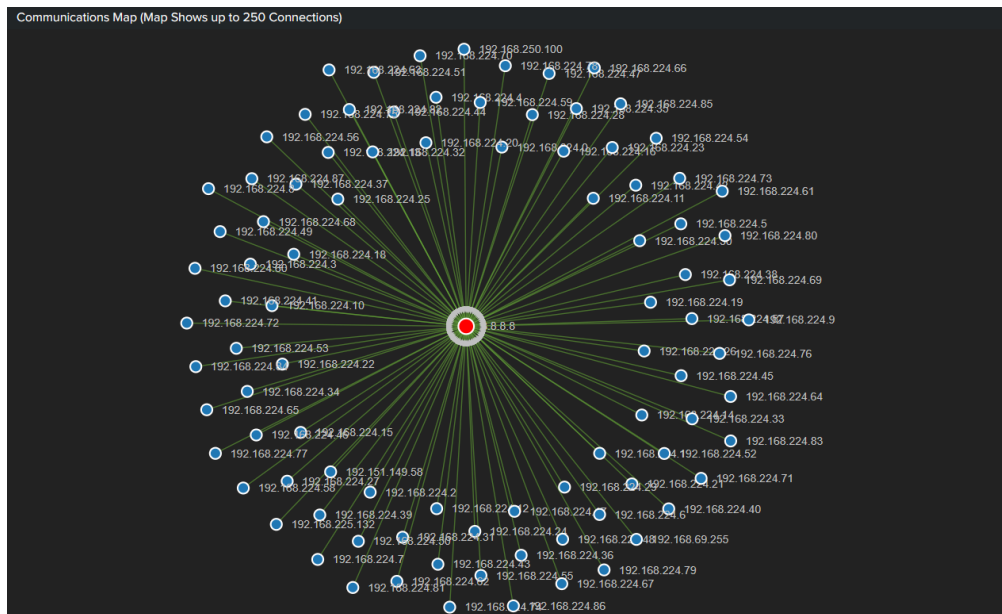


Figure 3.7: Force directed graph visualization of the communication of different hosts. The graph was filtered to show the traffic communications and paths of hosts with the IP address 8.8.8.8 (a well known DNS address).

3.4 Dashboard Development

In Splunk, dashboard visualization development involves utilizing the provided framework, which includes predefined visualization options and input objects for data filtering within the dashboard. Additionally, Splunk allows the incorporation of add-ons that expand the range of visualization options and enhance the capabilities for creating desired graphics.

Rather than exclusively relying on the graphical user interface to add charts and elements to the dashboard, it is possible to utilize the XML source code of the page allowing for further customization and the definition of more complex behavior, such as panel visibility based on predefined events. In the latest version of Splunk Enterprise, the XML code used for dashboard pages has been replaced by the new framework called Dashboard Studio which uses JSON as source code for the definition of the dashboards. This transition offers increased flexibility in positioning panels and graphs within the web page, but it is important to note that the new version is still in a development phase and has not entirely replaced the older XML version. The XML format retains the ability to perform more complex operations, ensuring compatibility with existing functionalities and workflows.

Developers can also incorporate custom JavaScript code to define personalized visualizations or modify the rendering of default ones, such as modifying specific

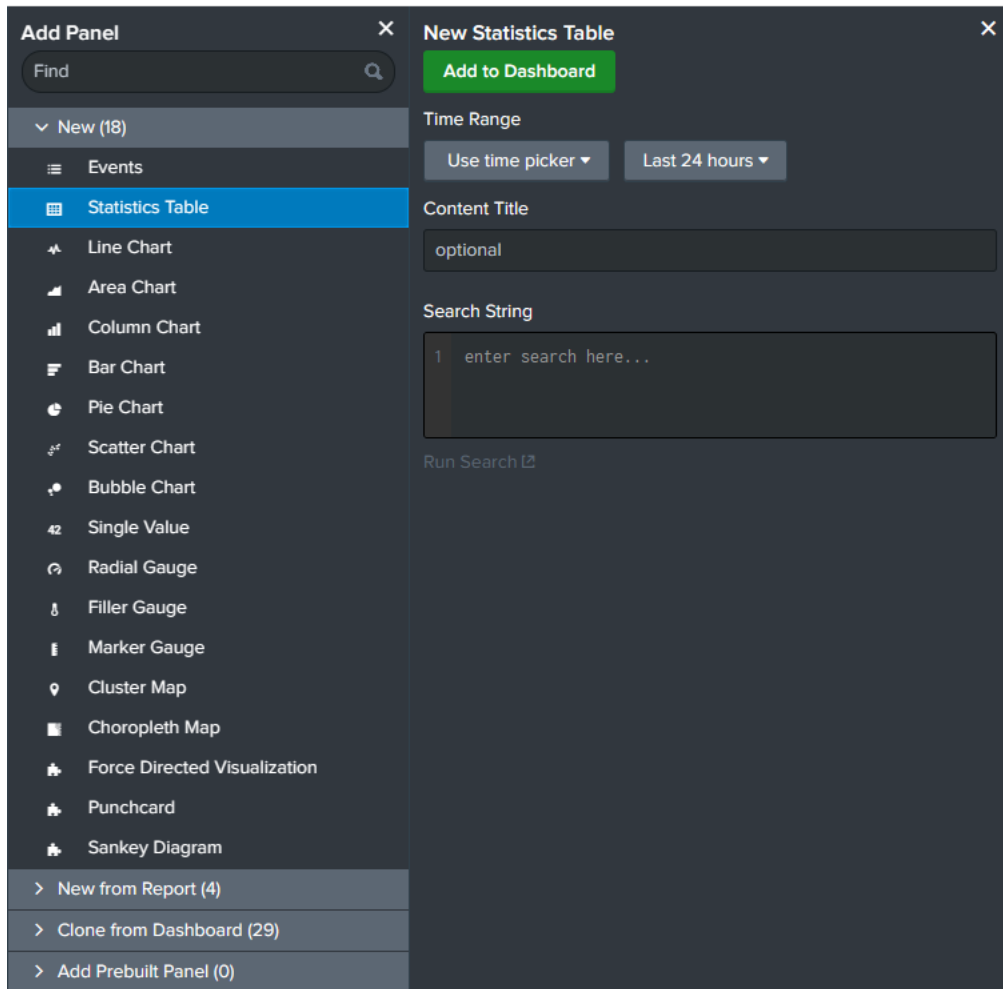


Figure 3.8: Dashboard edit configuration interface to add panel with statistical table.

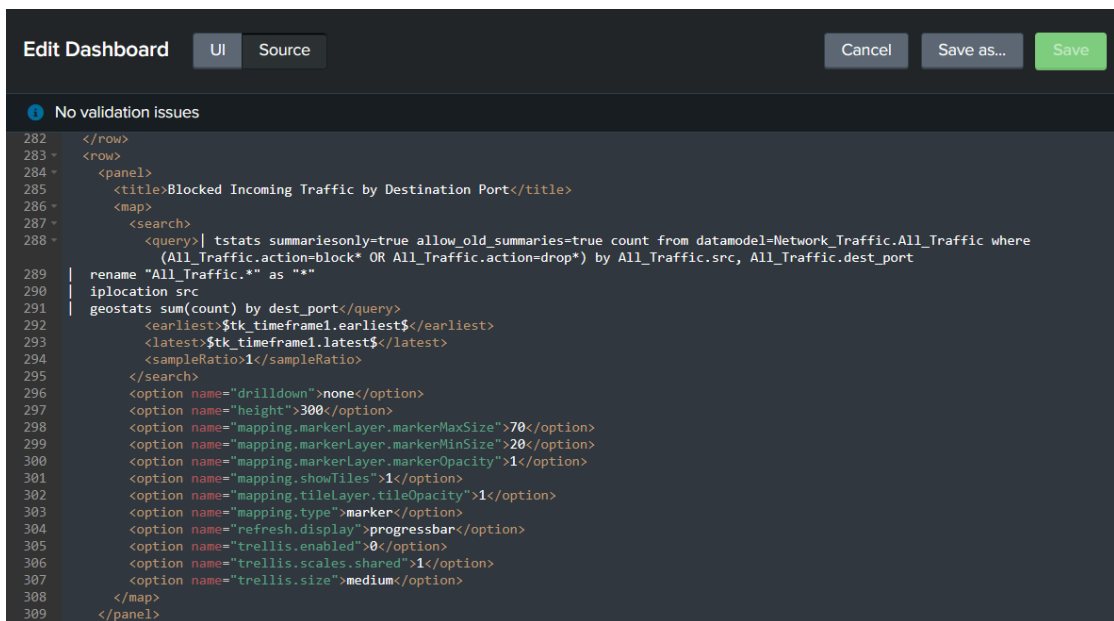
columns of tables or including buttons and other element in certain panels. Splunk offers a dedicated library known as SplunkJS, designed for creating HTML elements with the same design and configuration options as those provided by the Splunk web interface.

3.4.1 Dashboard Source Code

The XML code showed in the Figure 3.9 describe the configuration for a cluster map within the dashboard, illustrating blocked traffic by destination port. Each panel within the dashboard must contain a visualization element, for instance a map element or a table, along with a search query defining the events to be displayed.

Within the query, the time range of the search can be specified using the earliest and latest tags, which can incorporate static values or tokens referencing input elements present in the dashboard. Tokens are denoted by the dollar symbol and their values can dynamically change based on user interactions with the dashboard, such as selecting a new time frame from the dedicated input or choosing a column in a table with a configured drilldown option.

Additional customization options can be integrated to further enhance the behavior of the visualization, including parameters like the minimum and maximum marker size, trellis options, colors of the fields.



```

282 </row>
283 </row>
284 <panel>
285 <title>Blocked Incoming Traffic by Destination Port</title>
286 <map>
287 <search>
288 <query>| tstats summariesonly=true allow_old_summaries=true count from datamodel=Network_Traffic.All_Traffic where
  (All_Traffic.action=block* OR All_Traffic.action=drop*) by All_Traffic.src, All_Traffic.dest_port
289 | rename "All_Traffic.*" as "*"
290 | iplocation src
291 | geostats sum(count) by dest_port</query>
292 <earliest>${tk.timeframe1.earliest}</earliest>
293 <latest>${tk.timeframe1.latest}</latest>
294 <sampleRatio>1</sampleRatio>
295 </search>
296 <option name="drilldown">none</option>
297 <option name="height">300</option>
298 <option name="mapping.markerLayer.markerMaxSize">70</option>
299 <option name="mapping.markerLayer.markerMinSize">20</option>
300 <option name="mapping.markerLayer.markerOpacity">1</option>
301 <option name="mapping.showFiles">1</option>
302 <option name="mapping.tileLayer.tileOpacity">1</option>
303 <option name="mapping.type">marker</option>
304 <option name="refresh.display">progressbar</option>
305 <option name="trellis.enabled">0</option>
306 <option name="trellis.scales.shared">1</option>
307 <option name="trellis.size">medium</option>
308 </map>
309 </panel>

```

Figure 3.9: Dashboard source xml code while in edit mode. The code describe the configuration of a cluster map.

Furthermore, in the source code it is also possible to define a base search, which is a search identified by an unique identifier and which is not included in any visualization but created and executed at the loading of the dashboard. The results of the base searches can be used as a starting data set for additional queries present within the other visualization, enhancing the overall performance and response time by reducing the execution of similar searches that extract the same events.

Chapter 4

Experimental Results

This chapter summarizes the outcomes, findings, and observations derived from the simulations of the attacks and the detection performed by the application, given the configuration explained in the previous sections, and analyses conducted within the network environment. This experimental phase was conducted to demonstrate the effectiveness of the applied techniques, the robustness of the employed models, and the efficiency of the detection mechanisms against the simulated events.

The simulation was conducted within a laboratory environment, ensuring continuous activity of the Splunk infrastructure and real-time reception of events, to closely emulate a realistic scenario. The infrastructure was configured to notify by email and to log the events in the triggered alerts section whenever anomalies were detected within the network traffic data. Furthermore, a dashboard was created, to visually display all the information about the communication between the different hosts and the action taken by the firewall, as a tool for continuous monitoring of the system and investigation of the detected IP addresses.

4.1 Monitoring and Alerts Detection

The application includes an advanced system designed to identify and respond to the different types of network anomalies detected within the analyzed environment. This system incorporates two distinct alert mechanisms customized specifically to identify network scanning attacks and Command and Control attacks.

Both alert systems apply searches similar to those utilized in the dashboard for visualizing information. They focus on inspecting the same type of events, maintaining consistency in their analysis approach. To guarantee timely responses upon detection, the configured actions involve sending alert notifications via email and integrating the identified alert events into Splunk's triggered alerts section. This setup allows for immediate notification and visibility of identified anomalies.

within the Splunk infrastructure, allowing for a rapid intervention and resolution of the possible incident.

There are two primary methods for configuring email notifications in Splunk: via the web interface within the alerts section or by incorporating a specific command directly into the search query.

Web Interface Configuration Within the alerts section of the Splunk web interface, users can select the option for email notification as an alert action. In the page it is prompted to insert various details required for sending the email, including destination email addresses, subject line, priority, email body message, and other optional configurations (Figure 4.1). Splunk offers the flexibility to include additional information in the email, such as links to the search results, the search string used, or the results themselves presented as an inline table or attached in PDF or CSV format.

Sendmail Command in Search Query Alternatively, it is possible to incorporate the "sendmail" command directly into the search query to trigger email notifications. This method allows for additional control over the email content and formatting, directly within the search query itself. Users can customize the email content based on specific search parameters, making it possible to define detailed notifications or when specific customization is required for each row of the results: for instance it is possible to send a different email to different destination addresses depending on a certain value present in a certain field of the extracted events.

In the following example a custom message and a destination fields are created and later used as tokens by the sendmail command to correctly send the email to the specified destination. By following this approach it is possible to send multiple mail at the same time for each row returned by the search query before the sendmail command, and to customized for each result the message to send and the destinations. Additional options can be included as arguments to the command, to provide attachments in different formats or tables within the body message.

```
| makeresults
| eval customMessage = "sample sendmail message body"
| eval destination = "sample@splunk.com"
| sendmail to="$result.dest$" message="$result.customMessage$"
  sendresults=true inline=true format=raw sendpdf=true
```

Listing 4.1: Example of a SPL query to send multiple mail for each row of the result.

Create Alert
✕

When triggered

✕

✉
Send email
Remove

To

Comma separated list of email addresses.
[Show CC and BCC](#)

Priority

Subject

The email subject, recipients and message can include tokens that insert text based on the results of the search. [Learn More](#)

Message

Include

<input checked="" type="checkbox"/> Link to Alert	<input checked="" type="checkbox"/> Link to Results
<input type="checkbox"/> Search String	<input type="checkbox"/> Inline Table ▾
<input type="checkbox"/> Trigger Condition	<input type="checkbox"/> Attach CSV
<input type="checkbox"/> Trigger Time	<input type="checkbox"/> Attach PDF
<input checked="" type="checkbox"/> Allow Empty Attachment	

Type

Figure 4.1: Alert action email configuration with all possible options.

Both methods offer distinct advantages and can be chosen based on the specific requirements and preferences of the user or organization. The web interface configuration provides a user friendly approach for setting up email notifications, while direct command usage offers greater flexibility and control within the search query.

4.1.1 Alerts Definition

Configuring the alert system requires careful consideration of the search query's time range and the frequency at which scheduled searches should run to effectively detect potential attacks. Splunk offers real-time execution of searches, which can significantly facilitate attack detection. However, while this approach is optimal for fast identification, it's often discouraged. This is due to resource constraints in most real-world infrastructures, unable to sustain continuous query execution that may lead to system degradation and scheduled searches being skipped, compromising the system's overall performance. A balance between search frequency and system resources must be researched to ensure effective alerting without overloading the infrastructure.

Network Scan Alert

The search used to identify network scanning attacks is designed to examine data from the past month. To effectively configure the alert mechanism for this search, it has been observed to be efficient to schedule its execution at least every two minutes, ensuring a rapid response and detection of potential anomalies without imposing an elevate consumption of resources on the Splunk infrastructure. By running the search at regular intervals, it enhances the system's responsiveness in identifying and alerting possible network scanning activities while maintaining an optimal resource utilization level within the Splunk environment. The alert is activated when the query generates at least one result, meaning that an anomaly has been discovered by the system.

C2 Activity Alert

The search, which focuses on detecting Command and Control activity, is designed to analyze data spanning the previous 24 hours. Various scheduling time were tested during the simulation, with five minute intervals proved to be the most beneficial. This optimization is due to the relatively gradual nature of Command and Control attacks. Given their periodic communication pattern, shorter time intervals wouldn't significantly accelerate attack detection but would increase resource usage. While longer scheduling intervals are feasible, they would extend detection times, resulting in slower responses to potential attacks. Similar to the previous alert, the triggering of the response action occurs when the underlying query produces at least one result.

4.1.2 Alert Actions for Incident Response

Splunk offers various actions for automating responses to anomalies detected by alerts and scheduled searches. In addition to the default actions available within the Splunk Enterprise platform, many add-ons offer useful scripts that enhance integration between Splunk features and other software solutions. For instance, the Fortinet Fortigate app presents multiple options to transmit information about detected anomalies directly to the firewall. Subsequently, the firewall can implement rules based on predefined policies derived from this information.

The configured action of the alerts included in the application of this research are:

- **Email Notifications:** as described in the previous sections Splunk allows configuring email notifications triggered by alerts. It has been set up the recipient email addresses, subject lines, and content to transmit only the crucial information about the identified anomaly. Email notifications can be used to promptly alert designated individuals or teams, enabling them to verify a potential attack's presence and initiate appropriate responsive measures.
- **Triggered Alerts:** configure alerts to be logged within Splunk's infrastructure under the triggered alerts section and within a specific index of audit. This feature maintains a record of all detected anomalies, providing a centralized view for administrators and security teams to monitor and investigate incidents. It enables efficient tracking and management of triggered alerts for subsequent analysis and response.

To retrieve information about the active triggered alarms is it possible to query the `_audit` index, one of the index automatically generated by splunk and used to log all the interactions with the platform, such as searches, logins and logouts, capability checks, and configuration changes that generate audit events [14], and filters all the events related to the alerts fired belonging to a specific application. In the index all the alerts triggered are stored along with information about the expiration date, which can be used to distinguish the currently active alarms from the ones that are already expired.

After the extraction, all the necessary information can be displayed into a table to allows user to see the alarms in a custom dashboard instead of the activity page, where a dedicated dashboard created directly by Splunk can be used to achieve a similar result.

4.1.3 Alert Detection Results

In this section it is examined the outcomes derived from the implementation of the alert system specifically designed to detect and mitigate the attacks previously

described within the network traffic environment.

The study comprehend the analysis of various metrics, including the total volume of events, the successful identification of attack instances, the time taken for their detection, and the contextual aspects related to the attack simulations conducted concurrently with regular traffic generation. This concurrent environment enabled a comprehensive evaluation of the system’s effectiveness in distinguishing and detecting possible attack events among regular network operations.

Network Scan Alert Results

The two different network scan attacks were started with a total number of normal traffic events generated in a month of two millions events. The alert system demonstrated an acceptable response time, promptly identifying these attack events within an average duration of approximately twelve minutes from their occurrence. Furthermore the search required 317 attack events to successfully identify the attack and notify it. Simulating a highly intrusive attack with a duration of approximately one minute or less enables the search to immediately identify the anomaly.

The Figure 4.2 illustrates the frequency of attack events during the simulated scanning attack period. The graph place together the timing of the attack with the total count of normal traffic events within the corresponding time frame. Notably, the scanning attack generates a significantly higher volume of traffic compared to regular communication patterns, thus manifesting as an outlier in the observed network behavior and resulting in the detection by the configured alerts.

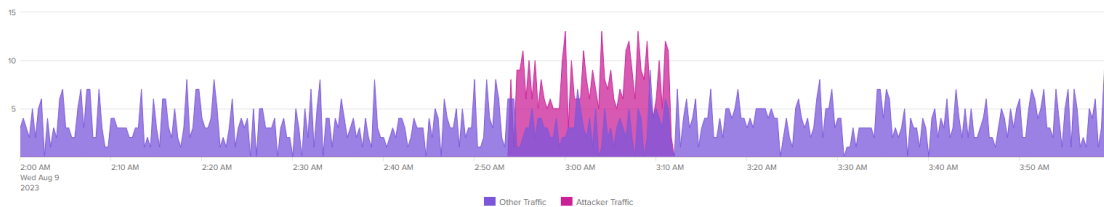


Figure 4.2: The timechart displays the count of traffic events, categorized into regular traffic and attacker traffic. The graph illustrates that the attacker generates a notably higher volume of traffic events within a short time window.

C2 activity Alert Results

The attack was started with a total number of normal traffic events created in a month of two millions events. The command and control attack generated an average count of events of 2 per minutes to simulate a relative high intrusive attack.

The alert system required two hours to correctly identifying these attack events with a total count of 300 events. Without having access to the content of the communication, a faster detection could result in a large number of false positive.

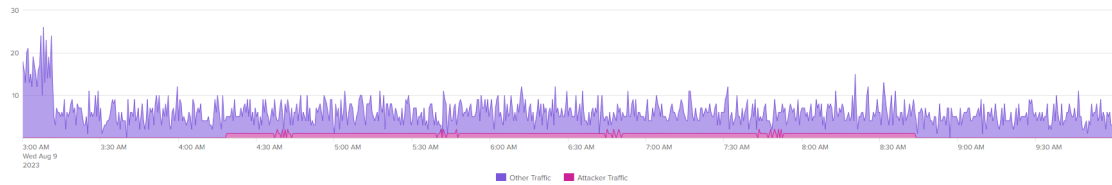


Figure 4.3: The timechart displays the count of traffic events, categorized into regular traffic and attacker traffic.

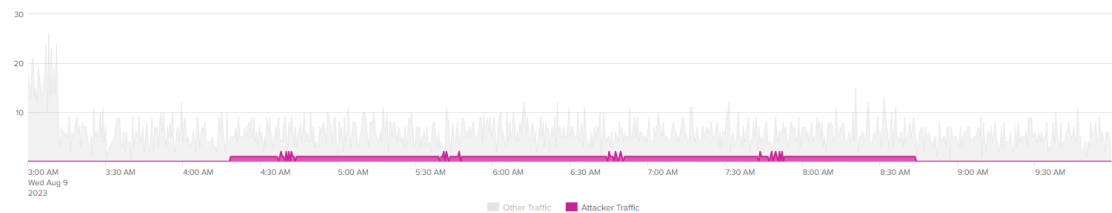


Figure 4.4: The timechart shows the highlighted count of traffic events of the attacker. A pattern of regular communication is evident, characterized by low traffic over an extended time window.

4.2 Dashboard visualization as Continuous Monitoring system

Splunk offers a robust framework for developing dashboards within its infrastructure, enabling users to visually interpret information through graphical displays that include essential data. These dashboards contain panels and inputs, where each visualization encompasses a search extracting data for presentation. Notably, dashboards can auto-refresh their searches, continuously updating with newly arrived data. This functionality proves valuable for security teams engaged in ongoing system monitoring to detect potential anomalies.

The designed application includes two distinct dashboards:

1. Network Anomalies Simulation: this dashboard showcase simulation data for a specific day and month to immediately show how the dashboard is populated

when all the configuration are completed. Since it shows only the simulated events, it requires direct data extraction from the index rather than utilizing the generated data model. This approach notably slows down query execution and panel creation due to the direct extraction process.

2. Network Anomalies Continuous Monitoring: this dashboard is designed for continuous infrastructure monitoring, providing the correct implementation of the time ranges and the utilisation of the configured data model to accelerate the searching time.

Both the dashboards enable users to restrict and filter the components of the network communication map by selecting any IP addresses within the panels, as well as filtering only the allowed or blocked traffic and the minimum number of host contacted by the specified source (Figure 4.5). This feature supports in-depth investigation of certain source or destination IP addresses, assisting in comprehending their behavior and interactions with contacted devices. In addition a time range filter can be applied to permit an historical analysis of network trends and patterns, supporting the identification of recurring issues or evolving threats, and allowing predictive analysis for future incident prevention.

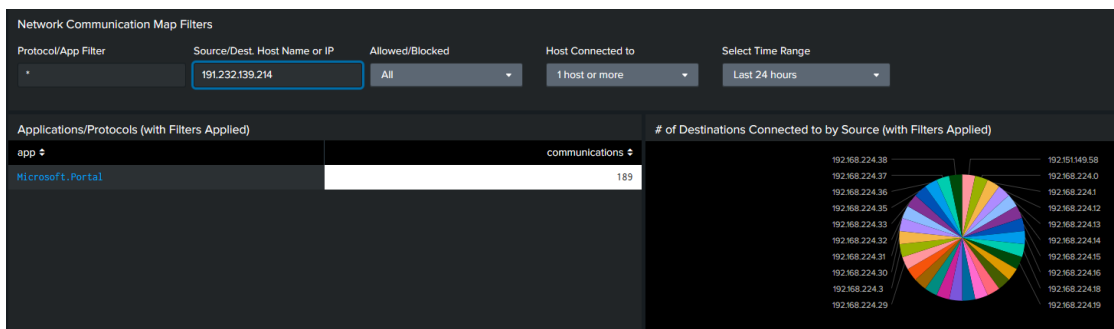


Figure 4.5: Possible filtering options included in the Network Communication Map section of the dashboards.

4.2.1 Network Anomalies Visualizations

The network anomalies section within the dashboard displays the outcomes of the searches described in the preceding chapter. It shows details concerning spikes in the number of destinations, suspected network scanning, and Command and Control activities identified by Splunk. Each information set includes two panels: one depicting the count of detected anomalies of that specific type, and the other presenting a table showcasing all relevant information (Figure 4.6). These numerical figures are commonly referred to as KPIs (Key Performance Indicators) as they offer immediate insight into the performance and health of the monitored infrastructure.

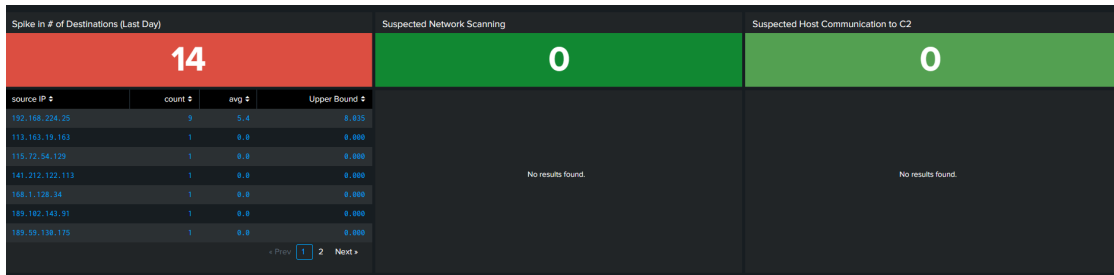


Figure 4.6: KPIs panels along with the correspondent tables that carries information about the outliers and possible threats detected.

Next to those panels, two geographical maps are available to showcase the blocked incoming traffic by destination port and the incoming traffic by application. These maps aim to visually represent the locations of IP addresses found within the indexed data, potentially offering insights into various traffic regions where the majority of the traffic originates. Furthermore, three additional panels have been included to highlight the top ten countries by blocked connections and by sources, as well as the network traffic categorized by different actions taken (Figure 4.7).

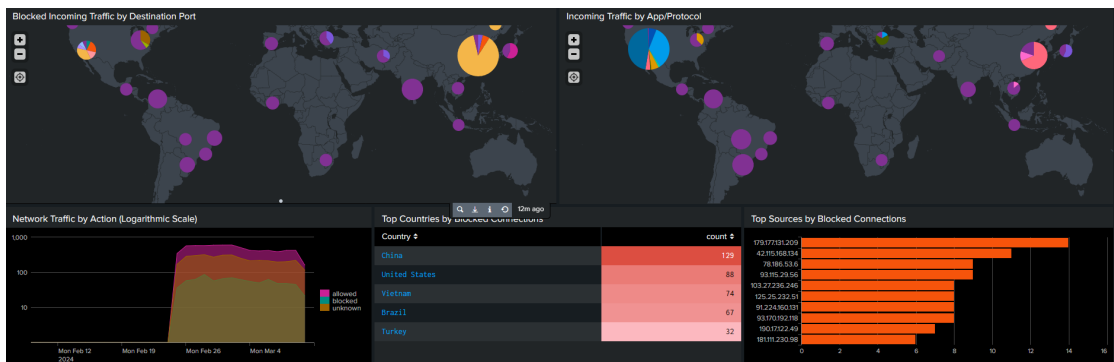


Figure 4.7: Geographical maps with blocked incoming traffic by destination port and incoming traffic by application, and statistical panels.

4.2.2 Network Communication Visualizations

The second section of the dashboard presents a distinct data set detailing communication among IP addresses within the examined network. Users can filter various panels by selecting specific IP addresses or host names, along with applications. Additionally, filtering options for allowed or blocked connections enable further investigation into specific communications. The communication map illustrates all connections among different IP addresses, and users can opt to filter hosts connected to more than a set number of devices, simplifying identification and

investigation of hosts with high traffic counts toward multiple hosts. Additionally, three panels are made available to showcase all applications found in the events, the count of destinations connected to by each source, and pairs of source-destination connections with traffic log counts within the selected time range.

4.2.3 Attacks Detection Within the Dashboard

During the simulation designed to assess the effectiveness of the alert system, various attack events were generated and subsequently detected. The dashboard, developed for comprehensive data visualization, serves as a tool to visually represent the information extracted from these detected anomalies.

The numerical panels are dynamically designed to highlight anomaly counts, providing immediate visual cues through color coding to indicate the detection of anomalies. The colour representation facilitates comprehension by allowing users to immediately determine the presence of the identified problems.

Three distinct colors denote the severity levels of detection:

- Green: represents a normal state where no significant issues have been detected within the monitored infrastructure. This color is displayed when the KPIs panel registers a count of events as 0.
- Yellow: signifies the detection of anomalies within a specified range of severity or count, indicating a cautionary or warning state.
- Red: indicates a critical situation where the count of identified attacks exceeds a specific threshold, denoting a high severity condition that necessitates immediate attention and action.

The other visualization elements within the dashboard provide a detailed overview of the detected anomalies. These graphics and tables clearly display the amount of discovered anomalies along with detailed information about the IP addresses involved in the reported communications.

4.3 Possible Detection Improvements

While the current detection mechanisms have proven effective in identifying network anomalies and potential threats within the simulated environment, there exist opportunities for further improvements:

Fine Tuning Thresholds Adjusting thresholds and criteria for anomaly detection could enhance the system's sensitivity without raising false positives. A precise calibration of count limits or time intervals may refine the detection accuracy.

The thresholds also depend on the size of the monitored infrastructure and the communication's behaviour among the internal devices.

Advanced Machine Learning Models Integrating machine learning algorithms could bolster anomaly detection capabilities. Utilizing predictive models and anomaly detection algorithms might uncover subtle irregularities not explicitly defined in rule based detection or by identifying outliers.

Expanded Data Sources integrating additional data sources beyond network traffic logs, such as server logs, endpoint data, or external threat intelligence feeds, could enrich the context for anomaly detection. By adding these information it is also possible to incorporate behavior-based analysis which could provide a deeper understanding of normal traffic patterns. Developing baselines of typical network behaviors could help in identifying deviations more accurately.

Automated Response Implementing automated responses to detected threats can augment incident response capabilities. This might involve automated isolation of affected systems or immediate deployment of predefined security measures.

4.3.1 Splunk Enterprise Security

By considering the previously described enhancements and adopting more advanced methodologies, the system's ability to detect and respond to network anomalies and security threats can be improved, ensuring a more robust and adaptive security posture.

Several of the suggested enhancements for improving detection capabilities are already incorporated within Splunk Enterprise Security (ES). Splunk ES offers a comprehensive suite of advanced features and functionalities designed to enhance security operations and threat detection:

- It integrates machine learning algorithms through the use of its Machine Learning Toolkit to create predictive models, conduct anomaly detection, and uncover complex patterns within data.
- Splunk ES can ingest and correlate data from various sources, including network logs, endpoint data, cloud services, and threat intelligence feeds. It provides also a powerful tool to investigate incidents and attacks, allowing users to correlate useful information relevant to the starting source.
- Splunk ES facilitates automated responses through Adaptive Response actions. It allows for automated actions or integration with third-party security tools to respond rapidly to identified threats or anomalies.

Splunk Enterprise Security is a separate product from the standard Splunk Enterprise, offering a specialized suite of security focused features and functionalities. It typically requires a separate license distinct from the standard Splunk Enterprise license and due to the complexity and specialized nature of security operations, it often necessitates a dedicated infrastructure deployment. This setup ensures optimal performance, scalability, and customization for security use cases.

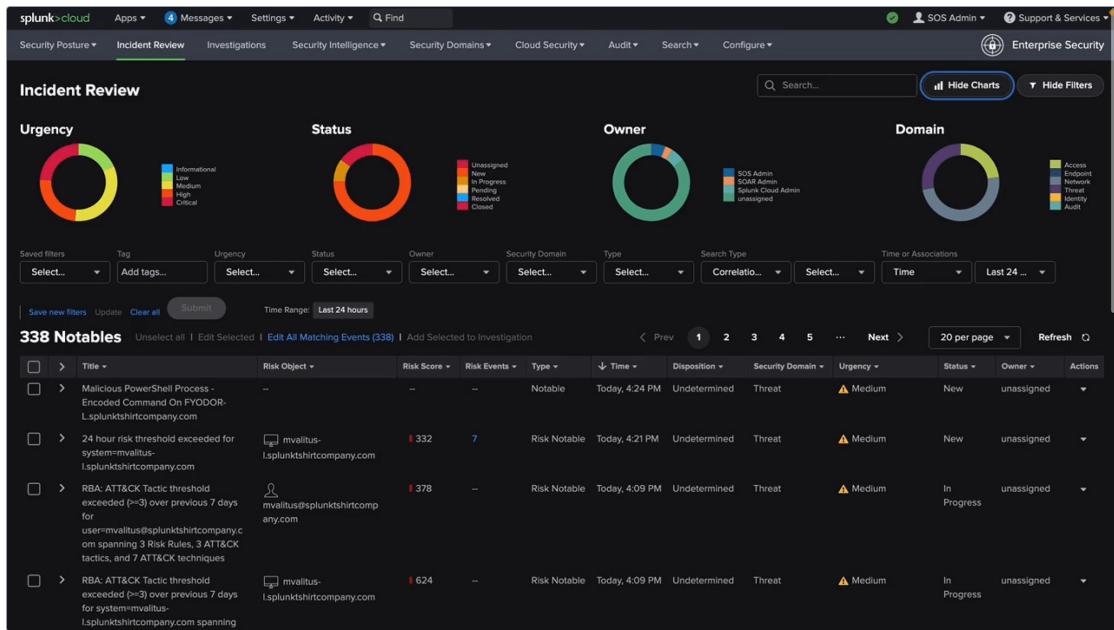


Figure 4.8: Incident Review dashboard of Splunk Enterprise Security with notable events used to start investigation inside the application.

Chapter 5

Conclusions

In this thesis research, the focus was on the detection of network anomalies utilizing Splunk Enterprise and Splunk Infosec, but the application in the security realm are more than the one described. By integrating different data types like authentication and authorization records, server logs, and web page responses, a variety of monitoring systems can be developed to uncover potential security threats within a system. The Splunk platform enables the creation of elaborate queries capable of constructing complex algorithms that aid in enhanced detection, targeting specific patterns indicative of security risks.

Using the Splunk Eventgen application, logs were generated to simulate various network attacks. These simulated attacks provided the groundwork for analyzing the effectiveness of detection mechanisms.

The alerts created successfully identified network scanning activities by analyzing spikes in destination IP counts. Similarly, Command and Control attacks were identified by discovering more regular communication gaps between IP addresses. Evaluation of the detection mechanisms highlighted their strengths and limitations: the need for thresholds and frequency adjustments to enhance sensitivity without overburdening system resources is fundamental to the correct execution and implementation of the alerts.

Splunk's dashboarding framework facilitated visualizing and comprehending network anomalies. Panels within the dashboard effectively communicate detected anomalies, providing quick insights into potential threats, and allowing investigation to understand the attacker's behaviour.

The Splunk Infosec application was designed with the intent of providing a straightforward tool, serving as an introduction to security measures for new customers and smaller companies managing their infrastructure. While adopting a more resilient solution like Splunk Enterprise Security presents advantages, it demands a higher level of expertise for proper configuration and utilization. In contrast, Infosec offers a more user-friendly interface and general applicability

within any Splunk environment.

The findings indicated potential enhancements in refining thresholds, optimizing search time ranges, and exploring the capabilities of Splunk Enterprise Security for more advanced threat detection. The continuous evolution of network security strategies remains crucial, and Splunk stands as a valuable tool in adapting to these evolving threats.

Bibliography

- [1] Splunk Inc. *Search Reference - eval*. Splunk Inc. Jan. 2024. URL: <https://docs.splunk.com/Documentation/Splunk/9.2.0/SearchReference/Eval> (cit. on p. 11).
- [2] Splunk Inc. *Search Reference - stats*. Splunk Inc. Jan. 2024. URL: <https://docs.splunk.com/Documentation/Splunk/9.2.0/SearchReference/Stats> (cit. on p. 11).
- [3] Splunk Inc. *InfoSec App for Splunk Documentation*. Splunk Inc. Oct. 2020. URL: <https://splunk-infosec-documentation.readthedocs.io/en/latest/> (cit. on p. 21).
- [4] Splunk Inc. *InfoSec App for Splunk Documentation*. Splunk Inc. Oct. 2020. URL: <https://splunk-infosec-documentation.readthedocs.io/en/latest/2%20-%20Concepts/> (cit. on p. 22).
- [5] Splunk Inc. *Punchcard - Custom Visualization*. Splunk Inc. July 2021. URL: <https://splunkbase.splunk.com/app/3129> (cit. on p. 23).
- [6] Splunk Inc. *Force Directed App For Splunk*. Splunk Inc. June 2021. URL: <https://splunkbase.splunk.com/app/3767> (cit. on p. 23).
- [7] Splunk Inc. *Splunk Event Generator: Eventgen*. Splunk Inc. Nov. 2020. URL: <http://splunk.github.io/eventgen/> (cit. on p. 25).
- [8] Fortinet Inc. *Fortinet FortiGate Add-On for Splunk*. Fortinet Inc. Nov. 2021. URL: <https://splunkbase.splunk.com/app/2846> (cit. on p. 25).
- [9] Splunk Inc. *Boss of the SOC (BOTS) Dataset Version 1*. Splunk Inc. Nov. 2020. URL: <https://github.com/splunk/botsv1> (cit. on p. 30).
- [10] The MITRE Corporation. *MITRE ATT&CK - Command and Control*. The MITRE Corporation. July 2019. URL: <https://attack.mitre.org/tactics/TA0011/> (cit. on p. 36).
- [11] The MITRE Corporation. *MITRE ATT&CK - Active Scanning*. The MITRE Corporation. Mar. 2022. URL: <https://attack.mitre.org/techniques/T1595/> (cit. on p. 43).

BIBLIOGRAPHY

- [12] Laiba Siddiqui. *Command and Control: Understanding and Defending Against C2 Attacks*. Splunk Blogs - The Key to Enterprise Resilience. Splunk Inc., Mar. 2023 (cit. on p. 44).
- [13] Splunk Inc. *Splunk® Enterprise - Search Reference iplocation*. Splunk Inc. Sept. 2023. URL: <https://docs.splunk.com/Documentation/Splunk/9.1.1/SearchReference/Iplocation> (cit. on p. 47).
- [14] Splunk Inc. *Splunk® Enterprise - Securing Splunk Enterprise*. Splunk Inc. Sept. 2023. URL: <https://docs.splunk.com/Documentation/Splunk/9.2.0/Security/AuditSplunkactivity> (cit. on p. 58).