Automation of Reporting Procedures in a Security Operations Center Environment

Scheepers, Yannick J.A.

Award date:
2022

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Department of Mathematics and Computer Science
Security Research Group

Automation of Reporting Procedures in a Security Operations Center Environment

Master's Thesis

Y.J.A. Scheepers

Supervisors:
Dr. L. Allodi
Dr. E. Zambon-Mazzocato

Committee members:
Dr. L. Allodi
Dr. E. Zambon-Mazzocato
Dr. T. Özçelebi

Eindhoven, February 2022

Abstract

Security Operations Centers process and utilize large amounts of data for the purpose of network monitoring. Often times these monitoring services are outsourced and given the (somewhat) unstructured nature and scale of the data, it is difficult to provide updates to non-technical clients. Commonly, this feedback is organised and constructed manually and reported back to the client, but with a growing number of clients this approach becomes infeasible. In this thesis we propose a novel design of a system that automates this process. We describe how the system conforms to user-specified configuration to obtain, correlate and compile data automatically into reports. The implementation of this design is then documented and the system is validated using real-world data.

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

1.1 Domain

Cybersecurity has become an increasingly more talked about topic in recent years. Companies need to protect their assets and sensitive information from criminals that nowadays take their approach online. Security monitoring allows companies to keep track of this type of activity within and around their network to detect malicious action. A Security Operations Center (SOC) is a system that, among other things, is able to monitor network traffic, log files and other audit traces for intrusion detection purposes. It creates alerts based on suspicious behaviour and informs people of incidents. When considering a large company that hosts not only several servers, but also hundreds of work computers, one can imagine that the number of alerts and incidents that get reported becomes unmanageable without proper tooling. Moreover, with an increasing number of alerts, it becomes important to carry out meaningful analysis and reporting of these alerts. Given that security monitoring is growing a more and more important market, and that the customer base for these services is increasing, it is safe to say that SOC operations need to be able to scale up.

1.2 Motivation

Many companies outsource the management of security monitoring to third parties. Since these security monitoring services are outsourced, they need to offer feedback to clients in the form of reporting. Each time a security incident is detected, reports should be generated to document the findings and enable the client to respond. Besides incident reports, however, clients may be interested in other, periodic or ad hoc reports containing information regarding the data that the service is analysing. An example of such a report can be found in Appendix A.2.1.

Generating these reports is difficult to automate if one would like their content and structure to be client-specific and easy to modify. Moreover, while a lot of current SOC implementations are based on Security Onion [6] and are thus quite standardized, it is possible that the architecture will change in the future. This means that the automation of reporting needs to evolve with it as well.

1.3 Problem formulation

Automated reporting of information is a widely applicable method for optimizing work efficiency. While it has been explored for many domains, it seems that in the context of Security Operations Centers there is still much to gain. Specifically, interacting with database-like systems of SOCs and correlating data from different sources are sub-problems that are not delved into much. We can thus define the problem formally as the following question: How can reporting procedures in a

multi-client Security Operations Center environment be automated? In order to further formulate the problem, it can be broken up in separate dimensions.

Multi-client nature One of these dimensions is the multi-client nature of outsourced SOCs. A company that specializes in network monitoring and analysis will naturally want to offer their service to multiple clients. Each client might be interested in different kinds of statistics/information regarding their network. Manually obtaining this information then becomes a time-consuming task for SOC employees that will only grow worse with an increasing number of clients.

Semi-structured large data Another dimension of this problem regards the data stored in SOCs. Although the idea behind some of the systems within a SOC is to organise and structure the data that flows in, the data is still difficult if not impossible to interpret for an inexperienced individual. Couple this with the fact that the amount of data can be huge, and similarly to the first dimension, manually composing the needed information in a way that is understandable for clients becomes very tedious.

Multi-system architecture The next dimension to consider is again related to data, but also to the architecture of SOCs. Typically, a SOC consists of multiple systems that work together to provide the analytical and monitoring functionality. Given that some of these can individually store different data and that data from different systems can be related it again adds to the complexity of manually retrieving and composing information from it.

Client feedback The last dimension to look at is client oriented in the context of outsourced SOCs. Given that clients pay for a service that monitors and analyses their network, they are interested in updates on what happens. As noted earlier, it cannot be assumed that clients of these outsourced SOC services are experts in the security domain. Therefore, it is not feasible to simply let them have a look at the internal systems of a SOC or the raw data within.

1.4 System design overview

The goal of this thesis is to design a system that achieves a solution to the aforementioned problem. On a high-level the design will describe a hierarchical configuration system that allows for the creation of complex data retrieval definitions. It will also describe the design of the correlation of data from different data sources by means of an abstract intermediate data representation. And lastly, the design describes how to adapt retrieved data into a format that can be used to generate figures to put in reports.

1.5 Outline

The outline for the rest of this thesis is as follows: Chapter 2 gives an overview of the background for this project consisting of related work and existing solutions. Chapter 3 then contains an in-depth description and discussion of the design of the system. Design choices and considerations are described each in the subsections corresponding to the different parts of the system. Chapter 4 contains a description of the implementation for the proposed architecture. Chapter 5 showcases the system with use-cases and examples and lastly, chapter 6 concludes the thesis.

Chapter 2

Background and related work

2.1 Background

This section serves to give background on the problem, which can then be used to more effectively discuss the related work in the next sections. To do this, we refer back to the problem dimensions given in Section 1.3. For each of the literature or existing solution discussed throughout this section we note their relation to one or more of the problem dimensions defined earlier. Additionally, the literature and solutions are categorized and assessed according to whether they solve the problem dimensions. A summary of the results can be found in Figure 2.1.

2.2 Related work

2.2.1 Literature

There is not much recent literature on automated reporting regarding the security monitoring domain. The reason for this is that cybersecurity is relatively new. Therefore, in this section we discuss some literature that does not necessarily fit our problem definition, but is a bit broader.

Most literature on automated document generation seems to focus their solution on a specific domain. The systems proposed in those solutions are usually tailored to a specific input format that is static. For example, the paper by Kiekebusch et al. [10] describes the design of a system that generates documents from the operational data of an observatory. The paper discusses that the system is capable of parsing log files and producing reports based on a given configuration. While it seems that these configuration files make the system adaptive, the input specification is limited to log files from the instruments at the observatory. This makes the scope for configurability narrow. Another paper that talks about automatic document generation is by Peterson et al. [11]. They describe a document generator that can be coupled to a database storing information related to spent nuclear fuel. The methods that were developed include two approaches to automatic generation of data reports based on the information in this database. Both approaches follow the same idea where the user provides template files that the system fills in with information from the database. Similarly to the paper by Kiekebusch et al. the proposed system in this paper is specifically made to work with the existing database and its structure.

A solution that is somewhat more adaptive in its input, and thus does not suffer as much from the aforementioned drawbacks is proposed by Gjorgjevikj et al. [9]. It is again an automated document generation system, however this time it can be applied to any SQL database. The advantage of this design over the others is that it does not have take any assumptions on the database layout.

Existing Technology	Multi-client nature	Semi-structured large data	Multi-system architecture	Client feedback
Crystal Reports	●	○	◐	●
JasperReports	●	○	○	●
ElastAlert	◐	○	○	◐
Skedler	●	◐	◐	●
Kibana Reports	●	◐	○	●
Proposed system	●	●	●	●

Table 2.1: Related work assessed on criteria

2.2.2 Existing solutions

Similarly to literature, existing solutions in our scope are limited. Given that our domain is focused on cybersecurity, there are not many solutions that specifically tackle automation of reporting. Nevertheless, there are a few solutions that are interesting to discuss.

An existing application in the domain of automated document generation is Crystal Reports [5]. This application allows the dynamic creation of reports based on a broad variety of data sources. It enables users to design report templates and specify data connections in a graphical interface. However, due to it being closed source it seems to be limited in its integration with custom data sources. In our context, this is clear drawback, since there is no support for data sources that are commonly used in SOCs (for example ElasticSearch).

This drawback is addressed by JasperReports [2], which is a library that also tackles automated document generation. JasperReports however is an open-source library, thus making it easier to incorporate in other systems. While it does not specifically support many SOC related data sources, it offers an API that allows adding implementations for other data sources.

ElastAlert [1] is a framework that works in conjunction with ElasticSearch to alert on anomalies in the data from ElasticSearch. It can be configured to send alerts to different outputs, for example to Slack or email. ElastAlert unfortunately is limited to data from ElasticSearch and does not generalize to other kinds of inputs.

Similarly, Skedler [7] also pulls information from ElasticSearch, but also from Logstash, Kibana and Grafana. In contrast to ElastAlert, it is more sophisticated in the sense that it can automate the creation of reports from these data sources, and then mail these reports to corresponding parties. The limitation of Skedler lies in the types of data sources it supports, namely, the ones mentioned above (ElasticSearch, Logstash, Kibana, Grafana).

Another example of report automation is Kibana Reports [4]. As the name suggests it takes its data from various Kibana dashboards and compiles this into reports that can be distributed manually or periodically. Again, the main limitation in the context of this project is that it can only interface with Kibana dashboards.

Chapter 3

Design

This chapter will discuss the design of the system. Firstly, we make the translation from the problem dimensions defined in 1.3 to concrete requirements that solve these dimensions. Then, a high-level overview of the architecture of the system is given that will list the subsystems it comprises of. This overview contains for each of the requirements a small description of how the design tackles them. After this we delve into more detail regarding the architecture and lastly, some implementation details are given.

3.1 Requirements

The requirements that our proposed design should have to address the problem dimensions are given below. A summary of which requirements map to which problem dimensions is given in Table 3.1

1. **Multiple data source support**. The system should support retrieving data from multiple data sources. This does not necessarily mean that a system that supports two data sources satisfies this requirement. Rather, it is important that a system has the concept of an abstract data source and has the possibility of supporting multiple data source implementations. It needs to be extendable to new data sources. This directly addresses the 'multi-system architecture' dimension of the problem in the sense that the system should be able to support a SOC containing multiple systems.

2. **Correlate different data sources**. Given the first requirement, the system needs to be able to correlate data from different data sources. Since there is no constraint on the type of the supported data sources, this correlation should be generic. This requirement is mostly aimed at addressing the 'multi-system architecture' dimension. If a SOC has multiple systems with interesting information, this might imply that data in different systems is related. Thus it may be important to correlate this data to obtain interesting information. Additionally, this requirement can also be seen as targeting the 'semi-structured large data' dimension, since the unstructuredness can be attributed to the data being in distinct systems. Allowing data sources to be correlated can solve this.

3. **Configurable data definition**. The system should in some capacity allow the user to configure the data that is retrieved. This requirement states that a pre-configured way of retrieving data does not suffice. The system should support a wide range of options for the user to choose. The targets of this requirement are the dimensions 'multi-client nature' and 'client feedback'. Allowing the type and structure of the data to be configurable makes it suitable for a multi-client environment. As noted earlier, different clients have different needs and thus want to see different information in their reports. A configurable data definition provides this. Similarly, in order to present clients with an understandable report, it is not

Requirement	Multi-client nature	Semi-structured large data	Multi-system architecture	Client feedback
Multiple data source support			●	
Correlate different data sources		●	●	
Configurable data definition	●			●
Configurable report template	●			●

Table 3.1: Requirements mapped to problem dimensions

possible to put raw data in it. Therefore, again, a configurable data definition provides a solution to this.

4. **Configurable report template**. Similar to the previous requirement, the user should be able to configure and customize the aesthetics of report templates. This is different to the configurability of the data in the sense that it only refers to the visual elements of the report. It does not affect the data that might be contained in these visual elements. While the definition of this requirement is slightly different than the previous requirement, the dimensions it targets are not. The 'multi-client nature' and 'client feedback' dimensions are addressed with this requirement. The reason simply being that the aesthetics of a report could be bound to the client it is meant for. Different clients might use different logos, images or titles.

3.2 Evaluating existing architectures

To evaluate possible architectural decisions for our solution we look at the system proposed by Gjorgjevikj et al.[9] and Peterson et al.[11], as the propositions therein are closely related to our problem.

In their architecture, Gjorgjevikj et al. propose an architecture with a component that parses document templates and a component that processes data from a database. These components are interesting to consider for our architecture as well, since they fit the requirements very well (namely 'configurable data definition' and 'configurable report template'). What their architecture is lacking however, is that there is no way to separately define data definitions and document templates. We can modify their approach of only using a single document template and divide this into two parts that each tackle one of our requirements. Firstly, a component that parses and handles document templates separate from any data. And secondly, a component that parses and handles data definitions. The document template will then not become an object within the component that processes data, but rather decoupled and only interacted with later once data is retrieved from the database. This divide is important as it is useful to independently change templates from data definitions and vice versa.

If we look at the work by Peterson et al.[11], we can see that their design (for the Java/LaTeX method) specifically targets PDF documents as output. While the method they propose directly after (the Python/Sphinx method), is free from this dependency as they mention that Sphinx can output to "several other formats related to documentation". The second approach is more adaptable to other document types. This is an architectural choice that we will adopt as well. The reason being is that we want to avoid that the internal representation of data in our system is specific to any type of output document. The part of the Python/Sphinx method Peterson et al. describe, where Sphinx is a central component that produces a form of output given a template and data, is something we will utilize as well.

3.3 High-level architecture

Following the evaluation of existing architecture in the previous section, we will now discuss our architecture. The subsystems described aim to implement one or more of these desired requirements. First, we will go over each subsystem and note their function and the requirements they address. Then, to show the relationship between the subsystems, a flow indicating report creation from start to finish is given. After this, the subsystems are described in more detail.

Data manager The first subsystem to note, and perhaps also the most complex, is the data manager. The data manager is responsible for managing data sources, querying these data sources and correlating data sources. This means that it already aims to cover the first two requirements: 'Multiple data source support' and 'Correlate different data sources'.

Module manager The second subsystem is the module manager, which has the purpose of handling so-called 'modules'. Modules in this design refer to objects that are the result of a user-specified configuration file. The main function of the module manager is to load and store these modules and act on the configured contents accordingly. The requirement this subsystem targets is the 'Configurable data definition'.

Report manager The last subsystem, which addresses the last requirement ('Configurable report template'), is the report manager. The task of this system is to bridge the gap between internal data and compiled reports. The report manager is responsible for reading and interpreting report templates, fill them with the requested data and output the compiled documents.

To illustrate how these subsystems are related, consider the following flow through the system: A user-specified configuration file is parsed and interpreted by the module manager. The module manager then instructs the data manager to query the data sources that are defined in the configuration file. The data manager will process and execute the query according to user specification and store the data in an intermediate format. This data is communicated back to the module manager, which then instructs the report manager to read the report template and fill it with the obtained data. The report manager will parse the report template, fill and compile it and then output the resulting document. This flow can be visualized by looking at the relationship diagram in Figure 3.1.

3.3.1 Module manager

To satisfy the third requirement ('Configurable data definition'), the system needs to be highly configurable. Providing a way for a user to define anything from simple to complex configurations makes the system usable in a wide range of scenarios. Moreover, since we intend to allow support of multiple data sources, the user should not only be able to configure from what data source to query from, but also what to query from that data source. The way this is accomplished is by having the input of the system be user-written configuration files. Subsequently, the module manager will handle these configuration files by parsing them into an internal structure that is then used to obtain the correct data.

3.3.2 Data manager

To address the 'Correlate different data sources' requirement, the design of the system will allow the correlation of data from different data sources via sub-queries. The configuration allows users to define sub-queries that can make use of the result data from an earlier finished query. In order for this to work, the result from querying any data source needs to be stored in an implementation-independent way. This immediately poses the problem that storing the entirety of the query result is infeasible. In ElasticSearch for example, the query result will contain a plethora of information that was not explicitly requested by the user. Therefore, the system requires users to specify which

exact fields from the result they are interested in. Only the specified fields will become available for sub-queries to use and will ultimately be available for the report. Querying and correlated queried data is handled by the data manager.

3.3.3 Report manager

The report manager will use template files to dictate the layout and aesthetics of the reports. This implements the fourth requirement ('Configurable report template') described at the start of this chapter. The template files should not only visual elements, but also references to dynamic elements that should be filled with user-defined data. Given that the data definition is given in configuration files, the report manager needs to match this data definition to the appropriate figures that the user defines in the template file.

3.4 Architectural details

Following the structure of the architecture described in Section 3.3, we can now go over the architectural details of the subsystems. A diagram showing the relation between the subsystems and the input and output can be found in Figure 3.1.

3.4.1 Module manager

To address configurability the input of the system is configuration files. Users of the system are supposed to write these configuration files, which allows them to define exactly what the system should do. Firstly, the configuration files contain some general information about the desired report, such as which report template file to use, where to output the resulting document and its file type. Then, the configuration file contains the definition for "widgets". A "widget" refers to a single data-containing object in the report. Examples of widgets are tables, charts, graphs or even dynamic text labels. These widgets then contain the specific configuration on what data source to retrieve the data from and how this should be done. This configuration includes filtering, sorting, and grouping, which should address the requirement that users can configure the data definition. It is expressive enough to allow for a broad range of configurability regarding what data the user is interested in. Next to this, the configuration allows for sub-querying, which should address the requirement of correlating different data sources. These sub-query configurations allow a user to define that a sub-query should be run from each resulting row of an existing query. That way, the user can define configurations that correlate different data sources with each other, or correlate the same data source with itself. Apart from this, the configuration can also contain data source specific configuration. This is dependent on the type of the data source that the data should be retrieved from and will not work with other data sources.

In order for the system to be able to interpret these configuration files, the module manager is responsible for parsing them into an internal structure that is easier to work with. This structure is hierarchical in nature, which gives the advantage that it can be easily distributed throughout the subsystems. For example, the configuration for filtering. Only specific routines that handle filtering in query creation need access to this data, thus this part of the structure is forwarded to those routines. Moreover, different data sources might handle filtering differently, while the configuration for filtering can stay the same. Therefore, these parts of the configuration can be reused for varying data sources. This holds for all the aforementioned configurable options for widgets: filtering, sorting, grouping, and sub-querying.

Another advantage of this hierarchical structure is that each component in the configuration provides its own way of parsing the corresponding section of the configuration. Thus the design is extensible in the sense that if new configuration needs to be added, it can be added to this structure and provide its own parsing. This can then be incorporated into the existing hierarchy.

In relation to the other subsystems, the module manager ties everything together. Based on a configuration file it will instruct the data manager to retrieve data according to the specification

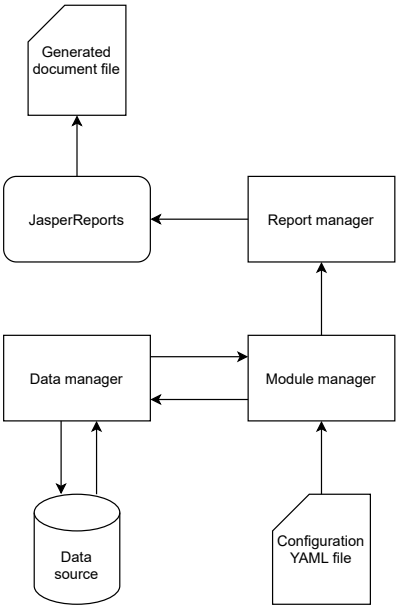


Figure 3.1: Diagram showing relations between subsystems and input/output

in the configuration and instruct the report manager to create a report from this data with the corresponding template file.

3.4.2 Data manager

Recall from the discussion in Section 3.3 that implementation-independent data source correlation is only possible given that the user specifies which fields to use from the result of a data source. The structure for the intermediate data representation that we can then use is quite simple. In essence, it consists of a matrix that stores values indexed on field name and row number. With this structure, sub-queries (that query for each row separately) can easily retrieve the value associated with a given field and their current row. Note that the size of this matrix is not static, meaning that it is possible to add values for new fields. This is important, as the result of sub-queries is also stored in this intermediate representation to allow for recursive sub-querying. The direct limitation we introduce with this matrix form is handling higher dimensional data. Fortunately, this can be circumvented by reducing dimensions by creating new fields.

The way the system deals with data sources is by providing an interface that implementations need to extend through which the data manager can access and manage it. Different types of data sources have different ways of querying data. Therefore, an implementation is needed for each different type that handles querying. The shared interface to accomplish this consists of three parts:

- A way to configure the data source. In a lot of cases in order to query a data source you need to provide credentials to contact it for example. This would be part of the configuration for the data source.

- A way to process a query that is to be executed later. An implementation of a data source needs to be able to process a query before executing it to validate whether user configuration is correct. If it succeeds, the implementation should yield a processed query that can be executed later.

- A way to execute a processed query. Only once the whole configuration is validated and all queries are processed will the data manager start querying data sources. The implementation

should take the given processed query and put the result of the query in the intermediate data representation.

To extend the system with a new implementation for a data source, one needs to adhere to this interface and register the data source in the data manager.

In regards to the other subsystems, the data manager only exposes the following functionality:

- A way to pass the configuration for all available data sources. This will instantiate all configured data source implementations in that configuration for later use.
- A way to process a query given configuration. This will find the corresponding data source and ask it to return a processed query through its interface.

3.4.3 Report manager

The report manager is responsible for loading and parsing template files, inserting data into the templates and compiling them into reports. In order for the report manager to know which data to insert at which point of the template, the user should define identifiers in the template files that match identifiers of the widgets in the configuration file. An important detail to discuss is how the report manager needs to transform its internal data representation, as discussed in the section above, into a representation that matches the figures that the user defines in the template file. To provide this, the report manager makes use of 'adapters' that transform the data from the internal representation into a format suitable to insert in the report based on the figure type.

As for the interaction with the other subsystems, the report manager requires some configuration along with data for each widget to fill reports. The configuration consists of where the report template is located, and what the location and type of the desired output file should be. The widget data consists of an instance of the internal data representation for each figure of the report. These instances should already be filled with data coming from the data manager.

Chapter 4

Implementation

This chapter will describe the implementation of the design in Chapter 3. The codebase for this implementation can be found in the Gitlab repository[1] associated with this project.

4.1 Configuration

As talked about in Section 3.4, the input of the system is done through configuration files. These configuration files are offered through YAML[2]. The choice for this file type is based on the fact that YAML is easy to create, read and modify. In essence it is a very human-friendly configuration language, which is needed since users of the system should be able to work with these configuration files. The parsing of these YAML files is handled by SnakeYAML [8], a YAML 1.1 processor for Java. SnakeYAML parses the configuration files and returns a Java Object representing the configuration file. The module manager will then convert this simple representation into an internal class structure. A UML diagram of this structure can be seen in Figure 4.1. Each of the classes in this structure are responsible for parsing a certain part of the configuration file. So given the SnakeYAML representation of the configuration file, each class will parse the root level objects in the representation and store them in their class. After that, they will construct instances of other classes in the hierarchy that handle lower-level objects in the configuration. The advantage of this is that the scope of each class is limited and it is easy to see how certain configurable elements are parsed and validated. Moreover, given that the configuration is the input to the system, it is already possible to do basic validation of whether the user-input is correct. This validation will at least tell us whether the structure of the configuration is valid and whether the types used for various keys are correct. Additionally, having distinct components in the structure makes it so that it is easy to distribute the configuration throughout the necessary parts of the system. For example, the ElasticSearchConfiguration can be directly passed on to the implementation of the ElasticSearch data source so it has access to all the ElasticSearch relevant configuration. Another advantage of this structure is that components related to ElasticSearchConfiguration, such as the entire FilterConfiguration sub-structure can also be utilised in other data source implementations. This is useful, since the FilterConfiguration features a complete filter system with the composite filters AND, OR and NOT and a few implementations of base filters that can offer filter configuration in any data source that supports that behaviour.

4.1.1 Examples

This section will contain a few examples of YAML configuration files that result in interesting reports. For space considerations, the JasperReports template files to actually generate these reports are not included in this section, but can be found in Appendix A.1.

[1] https://gitlab.tue.nl/esh-soc/automated-reporting-esh-soc
[2] https://yaml.org

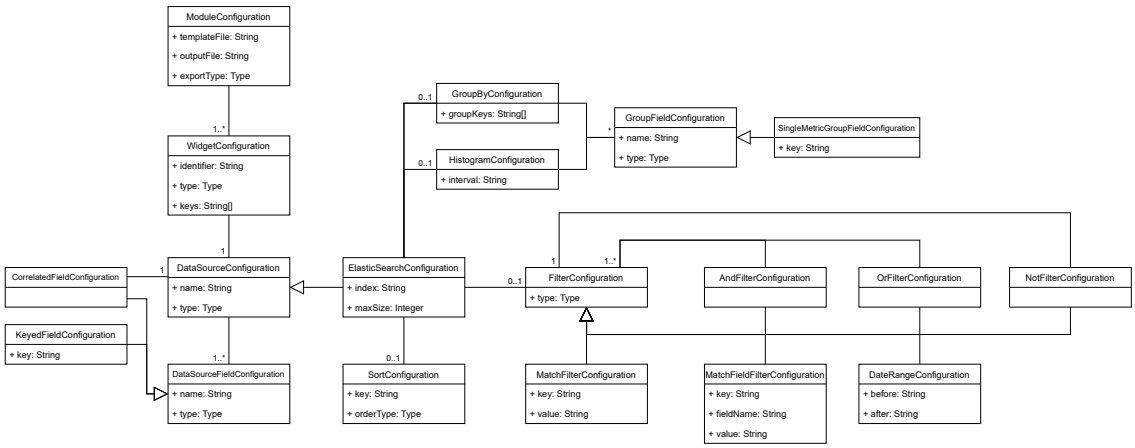


Figure 4.1: UML diagram of configuration class structure

The example in Listing 4.1 shows a configuration file that denotes a report with a barchart containing the most commonly occurring IP addresses in the last day queried from an ElasticSearch datasource.

```
template_file: /template-files/common-ips.jasper
output_file: common-ips.pdf
export_type: PDF
widgets:
  - id: commonly-occurring-ips
    type: bar-chart # specify the type of the figure for the report
    datasource:
      name: elasticsearch # this name and type should be a datasource that is
          defined in the datasource configuration file
      type: elasticsearch
      config:
        max_size: 20 # we want the top 20 results
        sort:
          key: num-occurrences # this key refers to the group field we define below
          order: desc
        filters:
          type: and
          filters:
            - type: match # only looking at IPs from Suricata Alerts
              key: event.module
              value: suricata
            - type: match
              key: event.dataset
              value: alert
            - type: range # only results from the last day
              key: "@timestamp" # the value for this key needs to be in quotes for
                  YAML
              to: now/d
              from: now-1d/d
        group_by:
          group_keys:
            - source.ip # by grouping on the source IP, we can count the
                occurrences within groups
          group_fields:
            - name: num-occurrences
              type: group-size # this type indicates that we want the group size
      fields:
        - name: source-ip
          type: keyed
          key: source.ip
        - name: num-occurrences
```

```
        type: keyed
        key: num-occurrences
  keys: # the keys that the barchart needs to display data from
    - source-ip
    - num-occurrences
```

Listing 4.1: Configuration example 1

The example in Listing 4.2 shows a configuration file that denotes a report with a table containing some data that is correlated by combining data from two ElasticSearch queries.

```
template_file: /template-files/alerts.jasper
output_file: alerts.pdf
export_type: PDF
widgets:
  - id: alert-data
    type: table # the type of figure for the report is a table
    datasource:
      name: elasticsearch
      type: elasticsearch
      config:
        max_size: 20 # we only want 20 results
        filters:
          type: and
          filters:
            - type: match # these filters ensure that we query Suricata alerts
              key: event.module
              value: suricata
            - type: match
              key: event.dataset
              value: alert
            - type: range # only results from the last month
              key: "@timestamp" # the value for this key needs to be in quotes for
                  YAML
              to: now/M
              from: now-1M/M
      fields:
        - name: rule-name
          type: keyed
          key: rule.name
        - name: destination-ip # although we dont use this field in the final table
            , we define it so that the correlated query can use it
          type: keyed
          key: destination.ip
        - name: hostname
          type: correlation # the type of this field is correlation, so we defined
              a new datasource configuration
          missing: # define that if there is no hostname for the given IP, we
              simply use the IP as fallback value
            type: fallback
            key: destination-ip
          correlate:
            name: elasticsearch
            type: elasticsearch
            config:
              max_size: 1 # we only need the hostname from a single IP
              filters:
                - type: and
                  filters:
                    - type: match # the hostname can be queried from Zeek DNS logs
                      key: event.module
                      value: zeek
                    - type: match
                      key: event.dataset
                      value: dns
                    - type: match-field # the match-field filter can filter results
                        based on the value of a field
                      key: dns.answers
```

```
                field_name: destination-ip
        fields:
          - name: hostname # this field name needs to correspond exactly to the
              name of the field with type correlation
            type: keyed
            key: dns.query.name
  keys:
    - rule-name # the table will consists of only the rule name and hostname per
      row
    - hostname
```

Listing 4.2: Configuration example 2

4.2 Data source querying

To be able to support multiple implementations of data sources, the implementation of the system makes use of an interface that defines methods that each data source needs to support. This way the data manager can easily handle different implementations via a shared interface. First of all, since most data sources need a way to be contacted (for example remotely hosted databases), the interface contains a method definition for configuring the data source with a configuration parameter. The parameter passed to this method is part of the data source configuration file, which contains entries for each data source that needs to be loaded and their respective details. An example of this file can be seen in Listing 4.3. This example shows the configuration containing an ElasticSearch data source that is reachable on hostname 127.0.0.1 and port 9200 via HTTPS. Secondly, the data source interface contains a method definition for processing a query given a data source configuration (or formally, a DataSourceConfiguration as seen in Figure 4.1). This method should return a processed query that is ready to be executed on the given data source. The reason that queries are first processed and not immediately executed is that with this intermediate method, we can validate user configuration of the data source query. Moreover, given that the system allows the configuration of sub-queries, it would not be user-friendly to execute the top-level query and then discover that the configuration for a sub-query is not valid. Thus processing queries before-hand will validate the entire querying configuration and eliminate runtime issues due to misconfiguration. Lastly, the interface has a method that queries the data source given a processed query. An additional parameter to this method is the intermediate data representation to store the result of the query in. The reason for passing this as a parameter instead of as the return type is that the system supports nested queries. Sub-queries might make use of earlier queried data and need to store the result of their own query.

```
data_sources:
  - type: elasticsearch
    name: elasticsearch
    config:
      hostname: 127.0.0.1
      port: 9200
      scheme: https
```

Listing 4.3: Data source configuration

4.3 Internal data representation

In Section 3.4 the general structure is given of the internal data representation that is used to store data while resolving queries. The matrix structure described is implemented as a mapping from field names to another mapping that maps row indices to string values. As there is no guarantee on the size of the result from a query it is not possible to use statically sized array. And, for sub-querying purposes, the system needs to index specific values in the structure rather quickly, we do not want to use dynamically allocated arrays either. The implementation of both the mappings used in our matrix structure is a hash-map to allow for quick lookup times.

4.4 Report creation

As discussed in Section 2.2, JasperReports [2] is an open source reporting engine. It provides a library that can be instructed to compile reports given a template and corresponding data to put into the figures of that template.

JasperReports works with template files in XML that can either be created by hand or with their report designer called Jaspersoft Studio [3]. These templates can contain static elements such as images or text, but they can also contain dynamic elements such as bar charts or tables. These dynamic elements are interesting, since we need to provide the data to fill them through the JasperReports library.

The implementation of the report manager is responsible for making the translation from queried data to generated reports. Before putting actual data in reports however, the template file for the report is validated. This validation checks whether parameters have been specified in the template file for each of the widgets defined in the configuration YAML. The IDs of these parameters should correspond to the IDs of the widgets, otherwise the data from the widgets will not show up in the generated report.

JasperReports expects a way to iterate linearly over the data and requests specific values in each iteration by their field name. This field name is defined in the report template and thus requires the user to correctly fill it in corresponding to the configuration file. To provide the iteration JasperReports demands, the report manager uses different adapters (as discussed in 3.4.3) that are each tailored to a format from JasperReports. All these adapters are able to read from the matrix representation discussed earlier and provide the data for JasperReports given that it is available. In case that for a given requested field or row index in the iteration, the value in the matrix is not defined, the report manager will provide a fallback value to ensure that JasperReports is still capable of producing a valid report. The adapters are implemented as simple iterators that provide the correct value based on the current iteration index and the requested field name. These requests are done by the JasperReport library as soon as the data for a figure is required. From the internal data representation discussed earlier, a hash-map is used to store the data and thus this iterator can look up the values in constant time.

Chapter 5

Showcasing the system

This section contains a showcase of the proposed system. We will first discuss the strategy we will employ to show that the system addresses the problem. This strategy is made in correspondence with ESH-SOC, a company that provides security monitoring services. After this we will present use-cases that indicate that the system has the capabilities to generate the reports corresponding to these use-cases. After internal discussion with ESH-SOC and the showcase of the system functionalities as described in this chapter, the system is now being integrated as part of the operational environment of ESH-SOC.

5.1 Strategy

To verify that the implementation adheres to the design and thus addresses the problem, we need to show that the implemented system can generate suitable reports. The reports used for the showcase need to have two properties. Firstly, they need to be interesting from an end-user perspective. The end-user in this case is a company that provides monitoring with a SOC. Therefore, interesting reports will contain information for either internal or external purposes, respectively to aid SOC processes or containing information relevant to clients of the company. Secondly, the showcase should show that the system can produce complex, non-trivial reports. This directly translates to the problem dimensions described in Section 1.3.

Following this, we have contacted ESH-SOC and together with two of their experts, we have discussed prototype reports. During this discussion, the experts provided existing document templates that show what kinds of data and layout ESH-SOC normally uses to provide as feedback to clients. Moreover, we discussed what kinds of data are interesting to put in the prototype reports. For example, for external reports the data could show how many alerts are generated for a certain category of rules, what the origin countries are for generated alerts or which local systems are the target of generated alerts. For internal reports, data such as alerts with a specific rule category, established connections for SSH sessions or hostname resolution of source IPs could be interesting.

Then, the experts gave a clear definition of the figures and the data within those figures that should be present in the reports. The exact details for these definitions are described in the sections below. Combining this with the layout from the document templates provided earlier, we created the configuration and report templates to have the system generate the prototype reports as shown below. The resulting documents were then shown to one of the experts at ESH-SOC again, which decided they were satisfactory.

5.2 Prototype reports

The showcase consists of two prototype reports, one internal and one external report. Both reports should use imagery and layout similar to existing documents of ESH-SOC. Additional requirements for these reports are listed in their respective subsections. For the internal report,

the requirements are based on what kind of data would benefit an ESH-SOC engineer in their work. The external report is based on typical reports that are delivered to existing clients of ESH-SOC and are normally made manually.

5.2.1 Internal report

The internal report should consist of a table that shows data about logged Suricata alerts from the last day. The columns of the table should display destination IP, source IP and the timestamp at which the alert was generated. Alerts should only be included in this table if the rule of the alerts was about SSH scanning. After this, another table should be included in the report that shows the exact same data, but the results should only be displayed if the Zeek connection logs show that a connection was established for this scan where the number of packets exchanged was at least 50.

These tables are interesting for engineers of ESH-SOC to more accurately narrow down potential threats of successful SSH scans.

5.2.2 External report

The external report should contain a few figures. Firstly, a table that shows data about logged Suricata alerts from the last month. The data should be grouped by rule name, rule category and severity. The columns of the table should display rule name, severity of the event, rule category and number of occurrences of that group. Next, a pie chart should be displayed that shows the number of occurrences of rule categories within Suricata alerts of the last month. The number of occurrences should show in percentage of the total. After this, another pie chart containing the number of occurrences of alerts grouped by their origin country and rule category. Again, the number of occurrences should show in percentage of the total. Lastly, a final pie chart that shows the number of occurrences of alerts grouped by destination hostname (or IP if hostname cannot be resolved). The number of occurrences here should also display in percentage of the total.

The figures in this report should give a client of the SOC insight on what alerts are being generated from their network in terms of occurrences and geographic location.

5.3 Showcase reports

This section will describe how the files are generated from the above specifications, what the input of the system consists of, and what the corresponding output looks like. For the full file contents of the configuration, template and output files, see Appendix A.2.

5.3.1 Internal report

The internal report consists of two figures, which are represented as “widgets” in the configuration file:

```
widgets:
  - id: ssh-scan-alerts
  - id: ssh-scan-alerts-successful
```

The IDs given here are used to reference these widgets in the template file as parameters:

```
  <parameter name="ssh-scan-alerts" class="net.sf.jasperreports.engine.data.
      JRAbstractBeanDataSource" isForPrompting="false"/>
  <parameter name="ssh-scan-alerts-successful" class="net.sf.jasperreports.engine.
      data.JRAbstractBeanDataSource"/>
```

The configuration for these widgets denotes that the figures should be tables and the data to be displayed is filtered to be only from Suricata alerts. Moreover, filters exists that ensures we only include data from rules that were about SSH scanning and that occurred in the last day. The following snippet indicates a condensed version of these filters:

```
        filters:
          type: AND
          filters:
            - type: match
              key: event.module
              value: suricata
            - type: match
              key: event.dataset
              value: alert
            - type: match
              key: rule.uuid
              value: "2001219"
            - type: range
              key: "@timestamp"
              to: now/d
              from: now-1d/d
```

The data source configuration in the YAML file for both widgets is largely the same as they retrieve the same data, except that the second widget correlates this data. This correlation ensures that only results are shown that have a connection in the Zeek logs of 50 packets or more. See the following (condensed) configuration snippet:

```
          correlate:
            name: elasticsearch
            type: elasticsearch
            config:
                  - type: range
                    key: server.packets
                    from: "50"
```

The entire output PDF file can be found in Appendix A.2.1.

5.3.2 External report

Similarly to the other report, the external report consists of four figures, which are represented as "widgets" in the configuration file:

```
widgets:
  - id: alerts-last-month
  - id: rule-category-occurrences
  - id: country-rule-occurrences
  - id: hostname-occurrences
```

These IDs again correspond to parameters in the template file:

```
  <parameter name="alerts-last-month" class="net.sf.jasperreports.engine.data.
      JRAbstractBeanDataSource" isForPrompting="false"/>
  <parameter name="rule-category-occurrences" class="net.sf.jasperreports.engine.
      data.JRAbstractBeanDataSource"/>
  <parameter name="country-rule-occurrences" class="net.sf.jasperreports.engine.
      data.JRAbstractBeanDataSource"/>
  <parameter name="hostname-occurrences" class="net.sf.jasperreports.engine.data.
      JRAbstractBeanDataSource"/>
```

Also similar to the internal report, the data to be retrieved is defined by using filters that denote which dataset and module to get the data from and in which time period. For example, the filter configuration from the first widget is shown as follows:

```
        filters:
          type: AND
          filters:
            - type: match
              key: event.module
              value: suricata
            - type: match
```

```
            key: event.dataset
            value: alert
          - type: range
            key: "@timestamp"
            to: now/M
            from: now-1M/M
```

All of these widgets feature a group-by clause in their configuration as can be seen in the first widget:

```
      group_by:
        group_keys:
          - rule.name
          - event.severity_label
          - rule.category
        group_fields:
          - name: num-occurrences
            type: group-size
```

The last widget requires hostnames instead of IP addresses for the pie chart. Since hostnames are not included in the default alert data from Suricata, we need to correlate the IP addresses with Zeek DNS logs to try and find hostnames for them. The (condensed) configuration for the correlation can be seen below:

```
        correlate:
          name: elasticsearch
          type: elasticsearch
          config:
            filters:
              type: AND
              filters:
                - type: match
                  key: event.module
                  value: zeek
                - type: match
                  key: event.dataset
                  value: dns
                - type: match-field
                  key: dns.answers
                  field_name: destination-ip
```

Again, the entire output PDF file can be found in Appendix A.2.2.

Chapter 6

Conclusions

6.1 Discussion

The proposed architecture of a system that is configurable and can correlate different data sources is scarcely explored in literature. Most literature and state-of-the-art focus on specific domains or specific use cases, which they then tackle effectively. While the motivation for this system is also grounded in a domain, namely cybersecurity, the discussed architecture is still more widely applicable. The problems that are tackled in this thesis are not necessarily only found in SOC environments. This implies that with slight modification to the described design, this system could be adapted for other domains that require automation of reporting procedures.

The importance of the system can clearly be seen in the cybersecurity domain. SOCs and their largely unstructured data stores are a prime example of why configurability and data correlation are crucial for automation of reporting. Moreover, informing non-technical people of ongoing activity in a SOC is difficult given the threshold of understanding SOC systems. Therefore, it is necessary to obtain and organise data into easy to read reports.

6.1.1 Improvements

The concepts described in this thesis leave room for future improvement. Most notably, the implementation is merely a starting point for a fully featured system. More specifically, the following points can be considered:

- The system features a single implementation of a data source, namely ElasticSearch. This can be expanded to support other commonly used data sources, such as SQL servers.
- While the architecture of the configuration is quite enabling, the implementation is missing some key components. Specifically, there are numerous base filters that are not implemented that could prove useful for querying purposes (for example a 'contains' filter).
- Currently supported export types are PDF and HTML, this can be expanded to other document types.

Apart from the implementation, the design can also be extended. Currently, the design does not handle complete automation like periodic scheduling of report generation. Additionally, for further automation of reporting procedures, one could think about features that Skedler [7] also offers, such as automatic mailing of generated documents.

6.2 Conclusion

In this thesis, we have proposed a novel approach to automating reporting in SOC environments. Problem dimensions have been identified relevant to this domain, from which features have been

extracted that a proposed solution should exhibit. These features are the following: “Multiple data source support”, “Correlate different data sources”, “Configurable data defintion” and “Configurable report template”. Following this, we have described the architecture of such a solution by going over the features and what they entail in the architecture. After this, the thesis delves into the implementation details of the system, that mention the configuration, data source querying and internal data representation aspects of the implementation. Finally, to validate the system, prototype reports have been discussed with ESH-SOC that should indicate the capability of the system to generate interesting and complex reports. These prototype reports have been generated with the proposed system and adhere to the specifications.

Bibliography

[1] Elastalert. https://github.com/Yelp/elastalert. Accessed: 16-02-2021. 4

[2] JasperReports. https://community.jaspersoft.com/project/jasperreports-library. Accessed: 11-02-2021. 4, 15

[3] Jaspersoft Studio. https://community.jaspersoft.com/project/jaspersoft-studio. Accessed: 11-02-2021. 15

[4] Kibana Reports. https://github.com/opendistro-for-elasticsearch/kibana-reports. Accessed: 15-02-2021. 4

[5] SAP Crystal Reports. https://www.crystalreports.com/reports/. Accessed: 11-02-2021. 4

[6] Security onion solutions. https://securityonionsolutions.com. Accessed: 15-10-2021. 1

[7] Skedler. https://www.skedler.com. Accessed: 15-02-2021. 4, 20

[8] SnakeYAML. https://bitbucket.org/snakeyaml/snakeyaml/. Accessed: 20-12-2021. 11

[9] Dejan Gjorgjevikj, Gjorgji Madjarov, Ivan Chorbev, Martin Angelovski, Marjan Georgiev, and Bojan Dikovski. Asgrt–automated report generation system. In *International Conference on ICT Innovations*, pages 369–376. Springer, 2010. 3, 6

[10] M Kiekebuscha and J Pavlich. Automatic report generator. In *Proceedings of SPIE*, volume 4009, pages 0277–786X, 2000. 3

[11] Josh Peterson, Bret van den Akker, Riley Cumberland, Paul Miller, and Kaushik Banerjee. Unf-st&dards unified database and the automatic document generator. *Nuclear Technology*, 199(3):310–319, 2017. 3, 6

Appendix A

Additionals

A.1 Configuration examples JasperReports templates

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Created with Jaspersoft Studio version 6.17.0.final using JasperReports
    Library version 6.17.0-6d93193241dd8cc42629e188b94f9e0bc5722efd  -->
<jasperReport xmlns="http://jasperreports.sourceforge.net/jasperreports" xmlns:xsi=
    "http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
    jasperreports.sourceforge.net/jasperreports http://jasperreports.sourceforge.
    net/xsd/jasperreport.xsd" name="use-case2" pageWidth="595" pageHeight="842"
    columnWidth="555" leftMargin="20" rightMargin="20" topMargin="20" bottomMargin=
    "20" uuid="d1b7a913-926e-4b49-ace4-4bae00e7c01d">
    <parameter name="commonly-occurring-ips" class="net.sf.jasperreports.engine.
        data.JRAbstractBeanDataSource" isForPrompting="false"/>
    <subDataset name="BarChartDataset" uuid="78a61432-e5ca-4585-a1ef-9da96eae5b45">
        <queryString>
            <![CDATA[]]>
        </queryString>
        <field name="source-ip" class="java.lang.String"/>
        <field name="num-occurrences" class="java.lang.Integer"/>
    </subDataset>
    <queryString>
        <![CDATA[]]>
    </queryString>
    <background>
        <band splitType="Stretch"/>
    </background>
    <title>
        <band height="74" splitType="Stretch">
            <staticText>
                <reportElement x="90" y="5" width="380" height="69" uuid="e7554391-
                    d1d3-4f74-b409-5869546e57bf"/>
                <textElement textAlignment="Center">
                    <font size="20"/>
                </textElement>
                <text><![CDATA[Commonly occurring IP addresses from last day]]></
                    text>
            </staticText>
        </band>
    </title>
    <detail>
        <band height="210">
            <barChart>
                <chart isShowLegend="false" evaluationTime="Report">
                    <reportElement x="80" y="10" width="400" height="200" uuid="
                        f44b30d2-a006-4ced-b49e-e1771e000a5a"/>
                    <chartTitle/>
                    <chartSubtitle/>
                    <chartLegend/>
```

```
                </chart>
                <categoryDataset>
                    <dataset resetType="Report">
                        <datasetRun subDataset="BarChartDataset" uuid="4f0d29bb-3
                            d2c-405f-bbe0-4b75d3e1a2ca">
                            <dataSourceExpression><![CDATA[$P{commonly-occurring-
                                ips}]]></dataSourceExpression>
                        </datasetRun>
                    </dataset>
                    <categorySeries>
                        <seriesExpression><![CDATA[""]]></seriesExpression>
                        <categoryExpression><![CDATA[$F{source-ip}]]></
                            categoryExpression>
                        <valueExpression><![CDATA[$F{num-occurrences}]]></
                            valueExpression>
                    </categorySeries>
                </categoryDataset>
                <barPlot>
                    <plot labelRotation="-45.0"/>
                    <itemLabel/>
                    <categoryAxisFormat labelRotation="-45.0">
                        <axisFormat labelColor="#000000" tickLabelColor="#000000"
                            axisLineColor="#000000"/>
                    </categoryAxisFormat>
                    <valueAxisFormat>
                        <axisFormat labelColor="#000000" tickLabelColor="#000000"
                            axisLineColor="#000000"/>
                    </valueAxisFormat>
                </barPlot>
            </barChart>
        </band>
    </detail>
</jasperReport>
```

Listing A.1: JasperReports template for configuration example 1

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Created with Jaspersoft Studio version 6.17.0.final using JasperReports
    Library version 6.17.0-6d93193241dd8cc42629e188b94f9e0bc5722efd  -->
<jasperReport xmlns="http://jasperreports.sourceforge.net/jasperreports" xmlns:xsi=
    "http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
    jasperreports.sourceforge.net/jasperreports http://jasperreports.sourceforge.
    net/xsd/jasperreport.xsd" name="use-case1" pageWidth="595" pageHeight="842"
    columnWidth="555" leftMargin="20" rightMargin="20" topMargin="20" bottomMargin=
    "20" uuid="e16262c1-6f67-4da2-9c88-0daf50fce8ae">
    <parameter name="alert-data" class="net.sf.jasperreports.engine.data.
        JRAbstractBeanDataSource" isForPrompting="false"/>
    <subDataset name="TableDataset" uuid="621cf54c-68fb-4486-81ce-69659b6c1ef3">
        <queryString>
            <![CDATA[]]>
        </queryString>
        <field name="rule-name" class="java.lang.String"/>
        <field name="hostname" class="java.lang.String"/>
    </subDataset>
    <style name="Table_TH" mode="Opaque" backcolor="#F0F8FF">
        <box>
            <pen lineWidth="0.5" lineColor="#000000"/>
            <topPen lineWidth="0.5" lineColor="#000000"/>
            <leftPen lineWidth="0.5" lineColor="#000000"/>
            <bottomPen lineWidth="0.5" lineColor="#000000"/>
            <rightPen lineWidth="0.5" lineColor="#000000"/>
        </box>
    </style>
    <style name="Table_CH" mode="Opaque" backcolor="#BFE1FF">
        <box>
            <pen lineWidth="0.5" lineColor="#000000"/>
            <topPen lineWidth="0.5" lineColor="#000000"/>
```

```
            <leftPen lineWidth="0.5" lineColor="#000000"/>
            <bottomPen lineWidth="0.5" lineColor="#000000"/>
            <rightPen lineWidth="0.5" lineColor="#000000"/>
        </box>
    </style>
    <style name="Table_TD" mode="Opaque" backcolor="#FFFFFF">
        <box>
            <pen lineWidth="0.5" lineColor="#000000"/>
            <topPen lineWidth="0.5" lineColor="#000000"/>
            <leftPen lineWidth="0.5" lineColor="#000000"/>
            <bottomPen lineWidth="0.5" lineColor="#000000"/>
            <rightPen lineWidth="0.5" lineColor="#000000"/>
        </box>
    </style>
    <queryString>
        <![CDATA[]]>
    </queryString>
    <background>
        <band splitType="Stretch"/>
    </background>
    <title>
        <band height="50" splitType="Stretch">
            <staticText>
                <reportElement x="87" y="10" width="380" height="30" uuid="f068e4d2
                    -5872-41b9-960f-4ad068d7715a"/>
                <textElement textAlignment="Center">
                    <font size="20"/>
                </textElement>
                <text><![CDATA[Alert data from last month]]></text>
            </staticText>
        </band>
    </title>
    <detail>
        <band height="240" splitType="Stretch">
            <componentElement>
                <reportElement x="62" y="40" width="430" height="200" uuid="19
                    d27766-ef9f-4b73-b3a2-0a5c99040f90">
                    <property name="com.jaspersoft.studio.layout" value="com.
                        jaspersoft.studio.editor.layout.VerticalRowLayout"/>
                    <property name="com.jaspersoft.studio.table.style.table_header"
                         value="Table_TH"/>
                    <property name="com.jaspersoft.studio.table.style.column_header
                        " value="Table_CH"/>
                    <property name="com.jaspersoft.studio.table.style.detail" value
                        ="Table_TD"/>
                    <property name="com.jaspersoft.studio.components.autoresize.
                        proportional" value="true"/>
                    <property name="com.jaspersoft.studio.components.autoresize.
                        next" value="true"/>
                </reportElement>
                <jr:table xmlns:jr="http://jasperreports.sourceforge.net/
                    jasperreports/components" xsi:schemaLocation="http://
                    jasperreports.sourceforge.net/jasperreports/components http://
                    jasperreports.sourceforge.net/xsd/components.xsd">
                    <datasetRun subDataset="TableDataset" uuid="e917f09b-cdf7-4e08
                        -8dd3-e4a9083c7286">
                        <dataSourceExpression><![CDATA[$P{alert-data}]]></
                            dataSourceExpression>
                    </datasetRun>
                    <jr:column width="108" uuid="7cd25200-3382-4bd7-905c-
                        b7f6ea9ce3bd">
                        <jr:tableHeader style="Table_TH" height="30"/>
                        <jr:columnHeader style="Table_CH" height="30">
                            <staticText>
                                <reportElement x="0" y="0" width="108" height="30"
                                    uuid="e9086ad8-f413-4e0c-aee6-831668a93899"/>
                                <text><![CDATA[rule-name]]></text>
```

```
                        </staticText>
                    </jr:columnHeader>
                    <jr:detailCell style="Table_TD" height="30">
                        <textField>
                            <reportElement x="0" y="0" width="108" height="30"
                                uuid="413f9599-c808-4418-83d3-3966b39ffd61"/>
                            <textFieldExpression><![CDATA[$F{rule-name}]]></
                                textFieldExpression>
                        </textField>
                    </jr:detailCell>
                </jr:column>
                <jr:column width="108" uuid="962886cb-66bd-4949-8859-511
                    d76611b66">
                    <jr:tableHeader style="Table_TH" height="30"/>
                    <jr:columnHeader style="Table_CH" height="30">
                        <staticText>
                            <reportElement x="0" y="0" width="108" height="30"
                                uuid="7f6e2950-4422-4dec-a683-73f122363bd3"/>
                            <text><![CDATA[hostname]]></text>
                        </staticText>
                    </jr:columnHeader>
                    <jr:detailCell style="Table_TD" height="30">
                        <textField>
                            <reportElement x="0" y="0" width="108" height="30"
                                uuid="63d34561-e032-4f23-89aa-6188580b8b18"/>
                            <textFieldExpression><![CDATA[$F{hostname}]]></
                                textFieldExpression>
                        </textField>
                    </jr:detailCell>
                </jr:column>
            </jr:table>
        </componentElement>
    </band>
</detail>
</jasperReport>
```

Listing A.2: JasperReports template for configuration example 2

A.2 File contents for showcase

A.2.1 Internal report

YAML configuration file

```
template_file: /template-files/validation2.jasper
output_file: validation2-output.pdf
export_type: PDF
widgets:
  - id: ssh-scan-alerts
    type: table
    datasource:
      name: elasticsearch
      type: elasticsearch
      config:
        max_size: 1000
        filters:
          type: AND
          filters:
            - type: match
              key: customer
              value: kembit
            - type: match
              key: event.module
              value: suricata
            - type: match
```

```
            key: event.dataset
            value: alert
          - type: match
            key: rule.uuid
            value: "2001219"
          - type: range
            key: "@timestamp"
            to: now/d
            from: now-1d/d
      group_by:
        group_keys:
          - destination.ip
          - source.ip
        group_fields:
          - name: earliest-timestamp
            type: minimum
            key: "@timestamp"
    fields:
      - name: destination-ip
        type: keyed
        key: destination.ip
      - name: source-ip
        type: keyed
        key: source.ip
      - name: earliest-timestamp
        type: keyed
        key: earliest-timestamp
  keys:
    - destination-ip
    - source-ip
    - earliest-timestamp
- id: ssh-scan-alerts-successful
  type: table
  datasource:
    name: elasticsearch
    type: elasticsearch
    config:
      max_size: 1000
      filters:
        type: AND
        filters:
          - type: match
            key: customer
            value: kembit
          - type: match
            key: event.module
            value: suricata
          - type: match
            key: event.dataset
            value: alert
          - type: match
            key: rule.uuid
            value: "2001219"
          - type: range
            key: "@timestamp"
            to: now/d
            from: now-1d/d
      group_by:
        group_keys:
          - destination.ip
          - source.ip
        group_fields:
          - name: earliest-timestamp
            type: minimum
            key: "@timestamp"
    fields:
      - name: destination-ip
```

```
          type: keyed
          key: destination.ip
        - name: source-ip
          type: keyed
          key: source.ip
        - name: earliest-timestamp
          type: keyed
          key: earliest-timestamp
        - name: packets
          type: correlation
          missing:
            type: discard
          correlate:
            name: elasticsearch
            type: elasticsearch
            config:
              max_size: 1
              filters:
                type: AND
                filters:
                  - type: match
                    key: event.module
                    value: zeek
                  - type: match
                    key: event.dataset
                    value: conn
                  - type: match
                    key: customer
                    value: kembit
                  - type: match-field
                    key: destination.ip
                    field_name: destination-ip
                  - type: match-field
                    key: source.ip
                    field_name: source-ip
                  - type: match
                    key: destination.port
                    value: "22"
                  - type: range
                    key: server.packets
                    from: "50"
                  - type: range
                    key: "@timestamp"
                    to: now/d
                    from: now-1d/d
            fields:
              - name: packets
                type: keyed
                key: server.packets
    keys:
      - destination-ip
      - source-ip
      - earliest-timestamp
```

Listing A.3: Showcase internal configuration file

JasperReports template file

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Created with Jaspersoft Studio version 6.17.0.final using JasperReports
    Library version 6.17.0-6d93193241dd8cc42629e188b94f9e0bc5722efd  -->
<jasperReport xmlns="http://jasperreports.sourceforge.net/jasperreports" xmlns:xsi=
    "http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
    jasperreports.sourceforge.net/jasperreports http://jasperreports.sourceforge.
    net/xsd/jasperreport.xsd" name="use-case2" pageWidth="595" pageHeight="842"
```

```
    columnWidth="555" leftMargin="20" rightMargin="20" topMargin="20" bottomMargin=
    "20" uuid="d1b7a913-926e-4b49-ace4-4bae00e7c01d">
  <import value="org.apache.commons.codec.binary.Base64" />
  <style name="Table_TH" mode="Opaque" backcolor="#FFFFFF">
    <box>
      <pen lineWidth="0.5" lineColor="#000000" />
      <topPen lineWidth="0.5" lineColor="#000000" />
      <leftPen lineWidth="0.5" lineColor="#000000" />
      <bottomPen lineWidth="0.5" lineColor="#000000" />
      <rightPen lineWidth="0.5" lineColor="#000000" />
    </box>
  </style>
  <style name="Table_CH" mode="Opaque" backcolor="#FFFFFF">
    <box>
      <pen lineWidth="0.5" lineColor="#000000" />
      <topPen lineWidth="0.5" lineColor="#000000" />
      <leftPen lineWidth="0.5" lineColor="#000000" />
      <bottomPen lineWidth="0.5" lineColor="#000000" />
      <rightPen lineWidth="0.5" lineColor="#000000" />
    </box>
  </style>
  <style name="Table_TD" mode="Opaque" backcolor="#FFFFFF">
    <box>
      <pen lineWidth="0.5" lineColor="#000000" />
      <topPen lineWidth="0.5" lineColor="#000000" />
      <leftPen lineWidth="0.5" lineColor="#000000" />
      <bottomPen lineWidth="0.5" lineColor="#000000" />
      <rightPen lineWidth="0.5" lineColor="#000000" />
    </box>
  </style>
  <style name="Table 1_TH" mode="Opaque" backcolor="#FFFFFF">
    <box>
      <pen lineWidth="0.5" lineColor="#000000" />
      <topPen lineWidth="0.5" lineColor="#000000" />
      <leftPen lineWidth="0.5" lineColor="#000000" />
      <bottomPen lineWidth="0.5" lineColor="#000000" />
      <rightPen lineWidth="0.5" lineColor="#000000" />
    </box>
  </style>
  <style name="Table 1_CH" mode="Opaque" backcolor="#FFFFFF">
    <box>
      <pen lineWidth="0.5" lineColor="#000000" />
      <topPen lineWidth="0.5" lineColor="#000000" />
      <leftPen lineWidth="0.5" lineColor="#000000" />
      <bottomPen lineWidth="0.5" lineColor="#000000" />
      <rightPen lineWidth="0.5" lineColor="#000000" />
    </box>
  </style>
  <style name="Table 1_TD" mode="Opaque" backcolor="#FFFFFF">
    <box>
      <pen lineWidth="0.5" lineColor="#000000" />
      <topPen lineWidth="0.5" lineColor="#000000" />
      <leftPen lineWidth="0.5" lineColor="#000000" />
      <bottomPen lineWidth="0.5" lineColor="#000000" />
      <rightPen lineWidth="0.5" lineColor="#000000" />
    </box>
  </style>
  <subDataset name="SSHScanAlerts" uuid="5fd8e6a8-b466-43e4-ba0f-a1f4e8e5bf81">
    <queryString>
      <![CDATA[]]>
    </queryString>
    <field name="destination-ip" class="java.lang.String" />
    <field name="earliest-timestamp" class="java.lang.String" />
    <field name="source-ip" class="java.lang.String" />
  </subDataset>
  <subDataset name="SSHScanAlertsSuccessful" uuid="e79e3b99-391c-4dc6-b225-48
      f0e135bbbf">
```

```
    <queryString>
      <![CDATA[]]>
    </queryString>
    <field name="destination-ip" class="java.lang.String"/>
    <field name="earliest-timestamp" class="java.lang.String"/>
    <field name="source-ip" class="java.lang.String"/>
  </subDataset>
  <subDataset name="Dataset1" uuid="a11faca0-2d3b-4049-8a8c-39adab3849e8">
    <queryString>
      <![CDATA[]]>
    </queryString>
  </subDataset>
  <parameter name="ssh-scan-alerts" class="net.sf.jasperreports.engine.data.
      JRAbstractBeanDataSource" isForPrompting="false"/>
  <parameter name="ssh-scan-alerts-successful" class="net.sf.jasperreports.engine.
      data.JRAbstractBeanDataSource"/>
  <queryString>
    <![CDATA[]]>
  </queryString>
  <variable name="esh-title-logo" class="java.lang.String">
    <variableExpression><![CDATA[""]]></variableExpression>
  </variable>
  <variable name="esh-header-logo" class="java.lang.String">
    <variableExpression><![CDATA[""]]></variableExpression>
  </variable>
  <background>
    <band splitType="Stretch"/>
  </background>
  <title>
    <band height="313">
      <image evaluationTime="Report">
        <reportElement x="134" y="40" width="292" height="90" uuid="dc4aa2e1-c936-4
            ba4-928b-7373e2b51a23">
          <property name="com.jaspersoft.studio.unit.width" value="px"/>
          <property name="com.jaspersoft.studio.unit.height" value="px"/>
        </reportElement>
        <imageExpression><![CDATA[new ByteArrayInputStream(Base64.decodeBase64($V{
            esh-title-logo}.getBytes()))]]></imageExpression>
      </image>
      <staticText>
        <reportElement x="80" y="180" width="245" height="50" uuid="a9971375-0f59
            -48a2-8cd7-a6af61f8b1d7"/>
        <textElement>
          <font size="11"/>
        </textElement>
        <text><![CDATA[Daily internal update document for SOC analysts containing
            SSH scanning alerts and successful connections.]]></text>
      </staticText>
      <staticText>
        <reportElement x="80" y="270" width="99" height="19" forecolor="#808080"
            uuid="feaab7c5-c7ca-4089-994d-78c6fedeaaf1"/>
        <textElement>
          <font size="11"/>
        </textElement>
        <text><![CDATA[Date]]></text>
      </staticText>
      <staticText>
        <reportElement x="180" y="270" width="99" height="19" forecolor="#808080"
            uuid="a4044c4f-679f-4d47-abc1-716b27a98ab6"/>
        <textElement>
          <font size="11"/>
        </textElement>
        <text><![CDATA[Version]]></text>
      </staticText>
      <staticText>
        <reportElement x="280" y="270" width="99" height="19" forecolor="#808080"
            uuid="356f91a9-890e-4e90-97cd-76c92814e3ac"/>
```

```
        <textElement>
          <font size="11"/>
        </textElement>
        <text><![CDATA[Status]]></text>
      </staticText>
      <staticText>
        <reportElement x="380" y="270" width="99" height="19" forecolor="#808080"
            uuid="b9765293-1874-4035-bc37-852e8121f720"/>
        <textElement>
          <font size="11"/>
        </textElement>
        <text><![CDATA[Classification]]></text>
      </staticText>
      <staticText>
        <reportElement x="80" y="289" width="99" height="19" forecolor="#808080"
            uuid="268bef46-7e06-4783-a9d1-f1d1dc955ec8"/>
        <textElement>
          <font size="11"/>
        </textElement>
        <text><![CDATA[10-12-2021]]></text>
      </staticText>
      <staticText>
        <reportElement x="180" y="289" width="99" height="19" forecolor="#808080"
            uuid="b42c1a49-5588-4a69-91e6-dd05a0f15be5"/>
        <textElement>
          <font size="11"/>
        </textElement>
        <text><![CDATA[1.0]]></text>
      </staticText>
      <staticText>
        <reportElement x="280" y="289" width="99" height="19" forecolor="#808080"
            uuid="75bf409c-3f4d-4319-82d6-7b9d98b006dc"/>
        <textElement>
          <font size="11"/>
        </textElement>
        <text><![CDATA[Draft]]></text>
      </staticText>
      <staticText>
        <reportElement x="380" y="289" width="99" height="19" forecolor="#808080"
            uuid="39c5df0c-c289-44ad-8372-a7d2108f2d38"/>
        <textElement>
          <font size="11"/>
        </textElement>
        <text><![CDATA[ESH-SOC internal]]></text>
      </staticText>
      <break>
        <reportElement x="0" y="312" width="98" height="1" uuid="b1c5f8bd-6ff0-401d
            -8b75-27f653474f16">
          <property name="com.jaspersoft.studio.unit.y" value="px"/>
        </reportElement>
      </break>
    </band>
  </title>
  <pageHeader>
    <band height="75">
      <printWhenExpression><![CDATA[$V{PAGE_NUMBER} != 1]]></printWhenExpression>
      <image evaluationTime="Report">
        <reportElement x="394" y="10" width="138" height="35" uuid="0ff4d513-59b3
            -4080-8b7d-8d52846f9d3d">
          <property name="com.jaspersoft.studio.unit.width" value="px"/>
          <property name="com.jaspersoft.studio.unit.height" value="px"/>
        </reportElement>
        <imageExpression><![CDATA[new ByteArrayInputStream(Base64.decodeBase64($V{
            esh-header-logo}.getBytes()))]]></imageExpression>
      </image>
      <staticText>
```

```
        <reportElement x="230" y="10" width="99" height="19" uuid="a33d4189-9f3f
            -4648-93f6-a650abff8034"/>
        <textElement textAlignment="Center">
          <font size="11"/>
        </textElement>
        <text><![CDATA[1.0]]></text>
      </staticText>
    </band>
  </pageHeader>
  <detail>
    <band height="270">
      <staticText>
        <reportElement x="90" y="0" width="380" height="30" forecolor="#00AD8E"
            uuid="1db6aa74-67e0-4752-8ea6-ebac6e15193c"/>
        <textElement textAlignment="Center">
          <font size="20"/>
        </textElement>
        <text><![CDATA[Alerts for SSH scanning]]></text>
      </staticText>
      <componentElement>
        <reportElement x="40" y="70" width="480" height="200" uuid="4f066657-b0db
            -415e-a407-e1e5192b2617">
          <property name="com.jaspersoft.studio.layout" value="com.jaspersoft.
              studio.editor.layout.VerticalRowLayout"/>
          <property name="com.jaspersoft.studio.table.style.table_header" value="
              Table_TH"/>
          <property name="com.jaspersoft.studio.table.style.column_header" value="
              Table_CH"/>
          <property name="com.jaspersoft.studio.table.style.detail" value="Table_TD
              "/>
          <property name="com.jaspersoft.studio.components.autoresize.proportional"
               value="true"/>
        </reportElement>
        <jr:table xmlns:jr="http://jasperreports.sourceforge.net/jasperreports/
            components" xsi:schemaLocation="http://jasperreports.sourceforge.net/
            jasperreports/components http://jasperreports.sourceforge.net/xsd/
            components.xsd">
          <datasetRun subDataset="SSHScanAlerts" uuid="8ec71464-7b7e-4464-a29f-26
              b653d937ac">
            <dataSourceExpression><![CDATA[$P{ssh-scan-alerts}]]></
                dataSourceExpression>
          </datasetRun>
          <jr:column width="160" uuid="b225f4d4-d549-47e1-b810-a153597ddaab">
            <jr:columnHeader style="Table_CH" height="15">
              <property name="com.jaspersoft.studio.unit.height" value="px"/>
              <staticText>
                <reportElement x="0" y="0" width="160" height="15" uuid="a18e9c18-6
                    ee7-41fc-a85f-0122a6a57ebb">
                  <property name="com.jaspersoft.studio.unit.leftIndent" value="px"
                      />
                </reportElement>
                <textElement>
                  <font isBold="true"/>
                  <paragraph leftIndent="5"/>
                </textElement>
                <text><![CDATA[Destination IP]]></text>
              </staticText>
            </jr:columnHeader>
            <jr:detailCell style="Table_TD" height="15">
              <property name="com.jaspersoft.studio.unit.height" value="px"/>
              <textField>
                <reportElement x="0" y="0" width="160" height="15" uuid="caaaa92b
                    -7170-4556-b504-ff640e2568be">
                  <property name="com.jaspersoft.studio.unit.leftIndent" value="px"
                      />
                </reportElement>
                <textElement markup="none">
```

```
            <paragraph leftIndent="5"/>
          </textElement>
          <textFieldExpression><![CDATA[$F{destination-ip}]]></
              textFieldExpression>
        </textField>
      </jr:detailCell>
    </jr:column>
    <jr:column width="160" uuid="30b80141-361f-466e-976b-dbcc11fb5a26">
      <jr:columnHeader style="Table_CH" height="15">
        <staticText>
          <reportElement x="0" y="0" width="160" height="15" uuid="3161b312-
              cbf5-4770-8032-ee3d85670241">
            <property name="com.jaspersoft.studio.unit.leftIndent" value="px"
                />
          </reportElement>
          <textElement>
            <font isBold="true"/>
            <paragraph leftIndent="5"/>
          </textElement>
          <text><![CDATA[Source IP]]></text>
        </staticText>
      </jr:columnHeader>
      <jr:detailCell style="Table_TD" height="15">
        <textField>
          <reportElement x="0" y="0" width="160" height="15" uuid="45da8fcd
              -36a4-4368-9611-eabd58bac214">
            <property name="com.jaspersoft.studio.unit.leftIndent" value="px"
                />
          </reportElement>
          <textElement>
            <paragraph leftIndent="5"/>
          </textElement>
          <textFieldExpression><![CDATA[$F{source-ip}]]></textFieldExpression
              >
        </textField>
      </jr:detailCell>
    </jr:column>
    <jr:column width="160" uuid="97e67bec-ef6f-4563-b1bd-17b7b1c4ea5f">
      <jr:columnHeader style="Table_CH" height="15">
        <staticText>
          <reportElement x="0" y="0" width="160" height="15" uuid="6aeee253-4
              d97-41b9-a97b-a3897f4592a8">
            <property name="com.jaspersoft.studio.unit.leftIndent" value="px"
                />
          </reportElement>
          <textElement>
            <font isBold="true"/>
            <paragraph leftIndent="5"/>
          </textElement>
          <text><![CDATA[Earliest timstamp]]></text>
        </staticText>
      </jr:columnHeader>
      <jr:detailCell style="Table_TD" height="15">
        <textField>
          <reportElement x="0" y="0" width="160" height="15" uuid="8eb78351-
              de3f-4acd-b6b2-028b080da889">
            <property name="com.jaspersoft.studio.unit.leftIndent" value="px"
                />
          </reportElement>
          <textElement>
            <paragraph leftIndent="5"/>
          </textElement>
          <textFieldExpression><![CDATA[$F{earliest-timestamp}]]></
              textFieldExpression>
        </textField>
      </jr:detailCell>
    </jr:column>
```

```
      </jr:table>
    </componentElement>
    <staticText>
      <reportElement x="60" y="30" width="439" height="39" uuid="3cc8a7b5-2ad5
          -4485-a7df-ccf2efb1def7"/>
      <text><![CDATA[Table containing information about alerts from the last day.
           The entries in the table are only alerts for which the rule was SSH
          scanning. The results are grouped by the destination and source IPs and
           the table contains an additional entry for the earliest timestamp of
          each group.]]></text>
    </staticText>
  </band>
  <band height="300">
    <staticText>
      <reportElement x="90" y="0" width="380" height="30" forecolor="#00AD8E"
          uuid="098ada48-9acb-4e28-bf47-92ba16acfd8a"/>
      <textElement textAlignment="Center">
        <font size="20"/>
      </textElement>
      <text><![CDATA[Alerts for successful SSH scanning]]></text>
    </staticText>
    <componentElement>
      <reportElement x="39" y="100" width="480" height="200" uuid="9039dcee-ce3a
          -41c3-8faf-590df2e278a2">
        <property name="com.jaspersoft.studio.layout" value="com.jaspersoft.
            studio.editor.layout.VerticalRowLayout"/>
        <property name="com.jaspersoft.studio.table.style.table_header" value="
            Table 1_TH"/>
        <property name="com.jaspersoft.studio.table.style.column_header" value="
            Table 1_CH"/>
        <property name="com.jaspersoft.studio.table.style.detail" value="Table 1
            _TD"/>
        <property name="com.jaspersoft.studio.components.autoresize.proportional"
             value="true"/>
      </reportElement>
      <jr:table xmlns:jr="http://jasperreports.sourceforge.net/jasperreports/
          components" xsi:schemaLocation="http://jasperreports.sourceforge.net/
          jasperreports/components http://jasperreports.sourceforge.net/xsd/
          components.xsd">
        <datasetRun subDataset="SSHScanAlertsSuccessful" uuid="be0718a4-3740-4c00
            -8d00-2de6b4fb124e">
          <dataSourceExpression><![CDATA[$P{ssh-scan-alerts-successful}]]></
              dataSourceExpression>
        </datasetRun>
        <jr:column width="160" uuid="13079a29-e5a5-4e1d-8ca8-8a5a47702734">
          <jr:columnHeader style="Table 1_CH" height="15">
            <property name="com.jaspersoft.studio.unit.height" value="px"/>
            <staticText>
              <reportElement x="0" y="0" width="160" height="15" uuid="659bcb53-8
                  d11-4dc7-a3ec-e4fcaa3c35a6">
                <property name="com.jaspersoft.studio.unit.leftIndent" value="px"
                    />
              </reportElement>
              <textElement>
                <font isBold="true"/>
                <paragraph leftIndent="5"/>
              </textElement>
              <text><![CDATA[Destination IP]]></text>
            </staticText>
          </jr:columnHeader>
          <jr:detailCell style="Table 1_TD" height="15">
            <property name="com.jaspersoft.studio.unit.height" value="px"/>
            <textField>
              <reportElement x="0" y="0" width="160" height="15" uuid="a4b6d8e6-
                  d55b-4678-a5e8-e54e3a7ec46f">
                <property name="com.jaspersoft.studio.unit.leftIndent" value="px"
                    />
```

```
            </reportElement>
            <textElement>
              <paragraph leftIndent="5"/>
            </textElement>
            <textFieldExpression><![CDATA[$F{destination-ip}]]></
                textFieldExpression>
          </textField>
        </jr:detailCell>
      </jr:column>
      <jr:column width="160" uuid="04ec916d-16cc-4bd7-88e6-55066c6160f7">
        <jr:columnHeader style="Table 1_CH" height="15">
          <staticText>
            <reportElement x="0" y="0" width="160" height="15" uuid="086b7bd8-4
                cab-4c40-9663-0b997302af26">
              <property name="com.jaspersoft.studio.unit.leftIndent" value="px"
                  />
            </reportElement>
            <textElement>
              <font isBold="true"/>
              <paragraph leftIndent="5"/>
            </textElement>
            <text><![CDATA[Source IP]]></text>
          </staticText>
        </jr:columnHeader>
        <jr:detailCell style="Table 1_TD" height="15">
          <textField>
            <reportElement x="0" y="0" width="160" height="15" uuid="aa0357f6-
                b6d6-45aa-a590-d3a1413c94c3">
              <property name="com.jaspersoft.studio.unit.leftIndent" value="px"
                  />
            </reportElement>
            <textElement>
              <paragraph leftIndent="5"/>
            </textElement>
            <textFieldExpression><![CDATA[$F{source-ip}]]></textFieldExpression
                >
          </textField>
        </jr:detailCell>
      </jr:column>
      <jr:column width="160" uuid="3a699b35-1283-48d9-944b-6736b88cef87">
        <jr:columnHeader style="Table 1_CH" height="15">
          <staticText>
            <reportElement x="0" y="0" width="160" height="15" uuid="0316bc97
                -36fc-48a3-9e08-ecb6f5c85fe7">
              <property name="com.jaspersoft.studio.unit.leftIndent" value="px"
                  />
            </reportElement>
            <textElement>
              <font isBold="true"/>
              <paragraph leftIndent="5"/>
            </textElement>
            <text><![CDATA[Earliest timestamp]]></text>
          </staticText>
        </jr:columnHeader>
        <jr:detailCell style="Table 1_TD" height="15">
          <textField>
            <reportElement x="0" y="0" width="160" height="15" uuid="13a7b28a
                -9083-4d59-8a74-245943943245">
              <property name="com.jaspersoft.studio.unit.leftIndent" value="px"
                  />
            </reportElement>
            <textElement>
              <paragraph leftIndent="5"/>
            </textElement>
            <textFieldExpression><![CDATA[$F{earliest-timestamp}]]></
                textFieldExpression>
          </textField>
```

```
            </jr:detailCell>
          </jr:column>
        </jr:table>
      </componentElement>
      <staticText>
        <reportElement x="61" y="30" width="439" height="70" uuid="b8c059f2-03d3-49
            b6-98ef-cc380e6980d1"/>
        <text><![CDATA[Table containing information about alerts from the last day.
             The entries in the table are only alerts for which the rule was SSH
            scanning. Moreover, only entries are shown for which the destination
            and source IP were also found in the zeek connection logs with a
            connection on port 22 that had at least 50 packets. The results are
            grouped by the destination and source IPs and the table contains an
            additional entry for the earliest timestamp of each group.]]></text>
      </staticText>
    </band>
  </detail>
  <pageFooter>
    <band height="75">
      <printWhenExpression><![CDATA[$V{PAGE_NUMBER} != 1]]></printWhenExpression>
      <staticText>
        <reportElement x="230" y="20" width="99" height="19" uuid="86188a6a-31f2
            -4139-adb4-ae8756370f14"/>
        <textElement textAlignment="Center">
          <font size="11"/>
        </textElement>
        <text><![CDATA[Draft]]></text>
      </staticText>
      <staticText>
        <reportElement x="20" y="20" width="99" height="19" uuid="a2b7742f-18bd-4
            efa-8c79-7b7d815fdcd1"/>
        <textElement textAlignment="Left">
          <font size="11"/>
        </textElement>
        <text><![CDATA[10-12-2021]]></text>
      </staticText>
      <textField>
        <reportElement x="440" y="20" width="99" height="19" uuid="296b6657-2cc7
            -409e-86db-f1ddf8f1b5ad">
          <property name="com.jaspersoft.studio.unit.width" value="px"/>
          <property name="com.jaspersoft.studio.unit.height" value="px"/>
        </reportElement>
        <textElement textAlignment="Right">
          <font size="11"/>
        </textElement>
        <textFieldExpression><![CDATA[$V{PAGE_NUMBER}]]></textFieldExpression>
      </textField>
    </band>
  </pageFooter>
</jasperReport>
```

Listing A.4: Showcase internal JasperReports template file

PDF output file

Daily internal update document for SOC analysts containing SSH scanning alerts and successful connections.

Date	Version	Status	Classification
10-12-2021	1.0	Draft	ESH-SOC internal

Alerts for SSH scanning

Table containing information about alerts from the last day. The entries in the table are only alerts for which the rule was SSH scanning. The results are grouped by the destination and source IPs and the table contains an additional entry for the earliest timestamp of each group.

Destination IP	Source IP	Earliest timstamp
172.26.1.11	14.63.222.63	2021-12-09T07:27:06.148Z
172.26.1.11	23.183.81.136	2021-12-09T01:36:15.235Z
172.26.1.11	23.183.81.249	2021-12-09T01:44:32.165Z
172.26.1.11	23.183.82.117	2021-12-09T20:58:37.347Z
172.26.1.11	23.183.82.135	2021-12-09T02:14:22.690Z
172.26.1.11	36.7.159.10	2021-12-09T21:13:43.222Z
172.26.1.11	37.110.19.112	2021-12-09T04:12:58.162Z
172.26.1.11	45.49.5.90	2021-12-09T19:02:20.023Z
172.26.1.11	45.88.137.100	2021-12-09T07:43:42.432Z
172.26.1.11	45.88.137.253	2021-12-09T01:54:08.633Z
172.26.1.11	61.177.173.31	2021-12-09T00:01:47.243Z
172.26.1.11	82.65.33.144	2021-12-09T20:53:08.728Z
172.26.1.11	82.66.59.61	2021-12-09T07:34:35.401Z
172.26.1.11	85.209.0.186	2021-12-09T05:16:11.501Z
172.26.1.11	116.98.59.26	2021-12-09T11:57:53.625Z
172.26.1.11	116.105.77.214	2021-12-09T11:57:14.487Z
172.26.1.11	116.110.97.175	2021-12-09T11:57:42.909Z
172.26.1.11	116.130.175.35	2021-12-09T09:33:36.270Z
172.26.1.11	116.235.94.247	2021-12-09T11:38:22.835Z
172.26.1.11	117.248.249.70	2021-12-09T06:55:56.732Z
172.26.1.11	137.184.51.62	2021-12-09T02:06:01.643Z
172.26.1.11	141.98.10.60	2021-12-09T02:44:59.696Z
172.26.1.11	141.98.10.63	2021-12-09T00:06:06.202Z
172.26.1.11	141.98.10.82	2021-12-09T04:37:59.526Z
172.26.1.11	141.98.10.202	2021-12-09T08:14:59.457Z
172.26.1.11	161.35.153.152	2021-12-09T11:10:57.136Z
172.26.1.11	161.35.153.169	2021-12-09T05:12:07.051Z
172.26.1.11	161.97.69.81	2021-12-09T00:01:51.801Z
172.26.1.11	165.22.195.82	2021-12-09T12:59:36.006Z
172.26.1.11	165.227.13.50	2021-12-09T13:06:54.168Z
172.26.1.11	167.71.48.128	2021-12-09T08:52:11.113Z
172.26.1.11	167.172.43.16	2021-12-09T21:24:32.274Z
172.26.1.11	173.212.209.109	2021-12-09T17:58:40.251Z
172.26.1.11	179.43.187.37	2021-12-09T16:43:45.402Z
172.26.1.11	181.165.67.231	2021-12-09T00:28:43.670Z
172.26.1.11	183.157.172.167	2021-12-09T23:37:34.252Z
172.26.1.11	185.90.136.69	2021-12-09T19:38:49.711Z

Destination IP	Source IP	Earliest timstamp
172.26.1.11	188.166.60.8	2021-12-09T21:54:35.157Z
172.26.1.11	194.85.248.40	2021-12-09T01:47:59.042Z
172.26.1.11	195.133.18.24	2021-12-09T00:29:47.085Z
172.26.1.11	205.185.115.39	2021-12-09T01:55:05.891Z
172.26.1.11	205.185.120.71	2021-12-09T15:00:19.569Z
172.26.1.11	205.185.124.178	2021-12-09T18:51:36.811Z
172.26.1.11	205.185.124.219	2021-12-09T05:30:35.926Z
172.26.1.11	209.141.34.220	2021-12-09T02:42:12.375Z
172.26.1.11	209.141.44.102	2021-12-09T13:36:16.832Z
172.26.1.11	209.141.47.245	2021-12-09T00:42:43.679Z
172.26.1.11	209.141.48.248	2021-12-09T11:53:20.687Z
172.26.1.11	209.141.53.74	2021-12-09T03:06:16.480Z
172.26.1.11	212.192.241.37	2021-12-09T05:37:13.582Z
172.26.1.11	221.131.165.33	2021-12-09T04:41:37.727Z
172.26.1.11	221.131.165.50	2021-12-09T03:57:13.755Z
172.26.1.11	221.131.165.56	2021-12-09T02:29:50.304Z
172.26.1.11	221.131.165.62	2021-12-09T08:02:23.785Z
172.26.1.11	221.131.165.65	2021-12-09T01:18:01.907Z
172.26.1.11	221.131.165.75	2021-12-09T02:10:20.829Z
172.26.1.11	221.181.185.94	2021-12-09T00:24:30.439Z
172.26.1.11	221.181.185.111	2021-12-09T05:43:05.057Z
172.26.1.11	221.181.185.151	2021-12-09T01:06:58.532Z
172.26.1.11	221.181.185.159	2021-12-09T01:40:07.515Z
172.26.1.11	222.186.30.76	2021-12-09T12:53:44.864Z
172.26.1.11	222.186.30.112	2021-12-09T05:11:51.567Z
172.26.1.11	222.186.42.7	2021-12-09T00:40:43.945Z
172.26.1.11	222.186.42.13	2021-12-09T03:02:24.865Z
172.26.1.11	222.186.42.137	2021-12-09T02:30:33.545Z
172.26.1.11	222.186.180.130	2021-12-09T05:44:56.125Z
172.26.1.11	222.187.232.39	2021-12-09T00:11:21.755Z
172.26.1.11	222.187.238.58	2021-12-09T03:33:14.370Z
172.26.1.11	222.187.254.41	2021-12-09T01:24:37.762Z

Alerts for successful SSH scanning

Table containing information about alerts from the last day. The entries in the table are only alerts for which the rule was SSH scanning. Moreover, only entries are shown for which the destination and source IP were also found in the zeek connection logs with a connection on port 22 that had at least 50 packets. The results are grouped by the destination and source IPs and the table contains an additional entry for the earliest timestamp of each group.

Destination IP	Source IP	Earliest timestamp
172.26.1.11	173.212.209.109	2021-12-09T17:58:40.251Z
172.26.1.11	179.43.187.37	2021-12-09T16:43:45.402Z

A.2.2 External report

YAML configuration file

```
template_file: /template-files/validation1.jasper
output_file: validation1-output.pdf
export_type: PDF
widgets:
  - id: alerts-last-month
    type: table
    datasource:
      name: elasticsearch
      type: elasticsearch
      config:
        max_size: 1000
        sort:
          key: num-occurrences
          order: desc
        filters:
          type: AND
          filters:
            - type: match
              key: customer
              value: kembit
            - type: match
              key: event.module
              value: suricata
            - type: match
              key: event.dataset
              value: alert
            - type: range
              key: "@timestamp"
              to: now/M
              from: now-1M/M
        group_by:
          group_keys:
            - rule.name
            - event.severity_label
            - rule.category
          group_fields:
            - name: num-occurrences
              type: group-size
      fields:
        - name: rule-name
          type: keyed
          key: rule.name
        - name: severity-label
          type: keyed
          key: event.severity_label
        - name: rule-category
          type: keyed
          key: rule.category
        - name: num-occurrences
          type: keyed
          key: num-occurrences
    keys:
      - rule-name
      - severity-label
      - rule-category
      - num-occurrences
  - id: rule-category-occurrences
    type: pie-chart
    datasource:
      name: elasticsearch
      type: elasticsearch
      config:
        max_size: 1000
```

```
        filters:
          type: AND
          filters:
            - type: match
              key: customer
              value: kembit
            - type: match
              key: event.module
              value: suricata
            - type: match
              key: event.dataset
              value: alert
            - type: range
              key: "@timestamp"
              to: now/M
              from: now-1M/M
        group_by:
          group_keys:
            - rule.category
          group_fields:
            - name: num-occurrences
              type: group-size
      fields:
        - name: rule-category
          type: keyed
          key: rule.category
        - name: num-occurrences
          type: keyed
          key: num-occurrences
    keys:
      - rule-category
      - num-occurrences
  - id: country-rule-occurrences
    type: pie-chart
    datasource:
      name: elasticsearch
      type: elasticsearch
      config:
        max_size: 1000
        filters:
          type: AND
          filters:
            - type: match
              key: customer
              value: kembit
            - type: match
              key: event.module
              value: suricata
            - type: match
              key: event.dataset
              value: alert
            - type: range
              key: "@timestamp"
              to: now/M
              from: now-1M/M
        group_by:
          group_keys:
            - source.geo.country_name
            - rule.category
          group_fields:
            - name: num-occurrences
              type: group-size
      fields:
        - name: country-name
          type: keyed
          key: source.geo.country_name
        - name: rule-category
```

```
        type: keyed
        key: rule.category
      - name: num-occurrences
        type: keyed
        key: num-occurrences
  keys:
    - country-name
    - rule-category
    - num-occurrences
- id: hostname-occurrences
  type: pie-chart
  datasource:
    name: elasticsearch
    type: elasticsearch
    config:
      max_size: 20
      sort:
        key: num-occurrences
        order: desc
      filters:
        type: AND
        filters:
          - type: match
            key: customer
            value: kembit
          - type: match
            key: event.module
            value: suricata
          - type: match
            key: event.dataset
            value: alert
          - type: range
            key: "@timestamp"
            to: now/M
            from: now-1M/M
      group_by:
        group_keys:
          - destination.ip
        group_fields:
          - name: num-occurrences
            type: group-size
    fields:
      - name: destination-ip
        type: keyed
        key: destination.ip
      - name: num-occurrences
        type: keyed
        key: num-occurrences
      - name: hostname
        type: correlation
        missing:
          type: fallback
          key: destination-ip
        correlate:
          name: elasticsearch
          type: elasticsearch
          config:
            max_size: 1
            filters:
              type: AND
              filters:
                - type: match
                  key: customer
                  value: kembit
                - type: match
                  key: event.module
                  value: zeek
```

```
                - type: match
                  key: event.dataset
                  value: dns
                - type: match-field
                  key: dns.answers
                  field_name: destination-ip
          fields:
            - name: hostname
              type: keyed
              key: dns.query.name
    keys:
      - hostname
      - destination-ip
      - num-occurrences
```

Listing A.5: Showcase external configuration file

JasperReports template file

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Created with Jaspersoft Studio version 6.17.0.final using JasperReports
    Library version 6.17.0-6d93193241dd8cc42629e188b94f9e0bc5722efd -->
<jasperReport xmlns="http://jasperreports.sourceforge.net/jasperreports" xmlns:xsi=
    "http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
    jasperreports.sourceforge.net/jasperreports http://jasperreports.sourceforge.
    net/xsd/jasperreport.xsd" name="use-case1" pageWidth="595" pageHeight="842"
    columnWidth="555" leftMargin="20" rightMargin="20" topMargin="20" bottomMargin=
    "20" uuid="e16262c1-6f67-4da2-9c88-0daf50fce8ae">
  <import value="org.apache.commons.codec.binary.Base64" />
  <style name="Table_TH" mode="Opaque" backcolor="#FFFFFF">
    <box>
      <pen lineWidth="0.5" lineColor="#000000" />
      <topPen lineWidth="0.5" lineColor="#000000" />
      <leftPen lineWidth="0.5" lineColor="#000000" />
      <bottomPen lineWidth="0.5" lineColor="#000000" />
      <rightPen lineWidth="0.5" lineColor="#000000" />
    </box>
  </style>
  <style name="Table_CH" mode="Opaque" backcolor="#FFFFFF">
    <box>
      <pen lineWidth="0.5" lineColor="#000000" />
      <topPen lineWidth="0.5" lineColor="#000000" />
      <leftPen lineWidth="0.5" lineColor="#000000" />
      <bottomPen lineWidth="0.5" lineColor="#000000" />
      <rightPen lineWidth="0.5" lineColor="#000000" />
    </box>
  </style>
  <style name="Table_TD" mode="Opaque" backcolor="#FFFFFF">
    <box>
      <pen lineWidth="0.5" lineColor="#000000" />
      <topPen lineWidth="0.5" lineColor="#000000" />
      <leftPen lineWidth="0.5" lineColor="#000000" />
      <bottomPen lineWidth="0.5" lineColor="#000000" />
      <rightPen lineWidth="0.5" lineColor="#000000" />
    </box>
  </style>
  <subDataset name="AlertsLastMonthDataset" uuid="621cf54c-68fb-4486-81ce-69659
      b6c1ef3">
    <queryString>
      <![CDATA[]]>
    </queryString>
    <field name="rule-name" class="java.lang.String" />
    <field name="severity-label" class="java.lang.String" />
    <field name="rule-category" class="java.lang.String" />
    <field name="num-occurrences" class="java.lang.String" />
  </subDataset>
```

```
<subDataset name="RuleCategoryDataset" uuid="9952b6e2-93de-48e5-931e-c42385cde844
    ">
  <queryString>
    <![CDATA[]]>
  </queryString>
  <field name="rule-category" class="java.lang.String"/>
  <field name="num-occurrences" class="java.lang.Integer"/>
</subDataset>
<subDataset name="CountryRuleDataset" uuid="a41bb180-c55d-4465-9ed6-763d722d6267"
    >
  <queryString>
    <![CDATA[]]>
  </queryString>
  <field name="country-name" class="java.lang.String"/>
  <field name="rule-category" class="java.lang.String"/>
  <field name="num-occurrences" class="java.lang.Integer"/>
</subDataset>
<subDataset name="HostnameDataset" uuid="503d1760-0518-418c-a82f-e80216bc0ed3">
  <queryString>
    <![CDATA[]]>
  </queryString>
  <field name="hostname" class="java.lang.String"/>
  <field name="destination-ip" class="java.lang.String"/>
  <field name="num-occurrences" class="java.lang.Integer"/>
</subDataset>
<parameter name="alerts-last-month" class="net.sf.jasperreports.engine.data.
    JRAbstractBeanDataSource" isForPrompting="false"/>
<parameter name="rule-category-occurrences" class="net.sf.jasperreports.engine.
    data.JRAbstractBeanDataSource"/>
<parameter name="country-rule-occurrences" class="net.sf.jasperreports.engine.
    data.JRAbstractBeanDataSource"/>
<parameter name="hostname-occurrences" class="net.sf.jasperreports.engine.data.
    JRAbstractBeanDataSource"/>
<parameter name="ROOT_DIR" class="java.lang.String"/>
<queryString>
  <![CDATA[]]>
</queryString>
<variable name="esh-title-logo" class="java.lang.String">
  <variableExpression><![CDATA[""]]></variableExpression>
</variable>
<variable name="esh-header-logo" class="java.lang.String">
  <variableExpression><![CDATA[""]]></variableExpression>
</variable>
<background>
  <band splitType="Stretch"/>
</background>
<title>
  <band height="313">
    <image evaluationTime="Report">
      <reportElement x="134" y="40" width="292" height="90" uuid="dc4aa2e1-c936-4
          ba4-928b-7373e2b51a23">
        <property name="com.jaspersoft.studio.unit.width" value="px"/>
        <property name="com.jaspersoft.studio.unit.height" value="px"/>
      </reportElement>
      <imageExpression><![CDATA[new ByteArrayInputStream(Base64.decodeBase64($V{
          esh-title-logo}.getBytes()))]]></imageExpression>
    </image>
    <staticText>
      <reportElement x="80" y="180" width="245" height="40" uuid="a9971375-0f59
          -48a2-8cd7-a6af61f8b1d7"/>
      <textElement>
        <font size="11"/>
      </textElement>
      <text><![CDATA[Monthly update document for [customer] containing figures
          related to network monitoring.]]></text>
    </staticText>
    <staticText>
```

```
        <reportElement x="80" y="270" width="99" height="19" forecolor="#808080"
            uuid="feaab7c5-c7ca-4089-994d-78c6fedeaaf1"/>
        <textElement>
          <font size="11"/>
        </textElement>
        <text><![CDATA[Date]]></text>
      </staticText>
      <staticText>
        <reportElement x="180" y="270" width="99" height="19" forecolor="#808080"
            uuid="a4044c4f-679f-4d47-abc1-716b27a98ab6"/>
        <textElement>
          <font size="11"/>
        </textElement>
        <text><![CDATA[Version]]></text>
      </staticText>
      <staticText>
        <reportElement x="280" y="270" width="99" height="19" forecolor="#808080"
            uuid="356f91a9-890e-4e90-97cd-76c92814e3ac"/>
        <textElement>
          <font size="11"/>
        </textElement>
        <text><![CDATA[Status]]></text>
      </staticText>
      <staticText>
        <reportElement x="380" y="270" width="99" height="19" forecolor="#808080"
            uuid="b9765293-1874-4035-bc37-852e8121f720"/>
        <textElement>
          <font size="11"/>
        </textElement>
        <text><![CDATA[Classification]]></text>
      </staticText>
      <staticText>
        <reportElement x="80" y="289" width="99" height="19" forecolor="#808080"
            uuid="268bef46-7e06-4783-a9d1-f1d1dc955ec8"/>
        <textElement>
          <font size="11"/>
        </textElement>
        <text><![CDATA[10-12-2021]]></text>
      </staticText>
      <staticText>
        <reportElement x="180" y="289" width="99" height="19" forecolor="#808080"
            uuid="b42c1a49-5588-4a69-91e6-dd05a0f15be5"/>
        <textElement>
          <font size="11"/>
        </textElement>
        <text><![CDATA[1.0]]></text>
      </staticText>
      <staticText>
        <reportElement x="280" y="289" width="99" height="19" forecolor="#808080"
            uuid="75bf409c-3f4d-4319-82d6-7b9d98b006dc"/>
        <textElement>
          <font size="11"/>
        </textElement>
        <text><![CDATA[Draft]]></text>
      </staticText>
      <staticText>
        <reportElement x="380" y="289" width="99" height="19" forecolor="#808080"
            uuid="39c5df0c-c289-44ad-8372-a7d2108f2d38"/>
        <textElement>
          <font size="11"/>
        </textElement>
        <text><![CDATA[ESH-SOC internal]]></text>
      </staticText>
      <break>
        <reportElement x="0" y="312" width="98" height="1" uuid="b1c5f8bd-6ff0-401d
            -8b75-27f653474f16">
          <property name="com.jaspersoft.studio.unit.y" value="px"/>
```

```
        </reportElement>
      </break>
    </band>
  </title>
  <pageHeader>
    <band height="75">
      <printWhenExpression><![CDATA[$V{PAGE_NUMBER} != 1]]></printWhenExpression>
      <image evaluationTime="Report">
        <reportElement x="394" y="10" width="138" height="35" uuid="0ff4d513-59b3
            -4080-8b7d-8d52846f9d3d">
          <property name="com.jaspersoft.studio.unit.width" value="px"/>
          <property name="com.jaspersoft.studio.unit.height" value="px"/>
        </reportElement>
        <imageExpression><![CDATA[new ByteArrayInputStream(Base64.decodeBase64($V{
            esh-header-logo}.getBytes()))]]></imageExpression>
      </image>
      <staticText>
        <reportElement x="230" y="10" width="99" height="19" uuid="a33d4189-9f3f
            -4648-93f6-a650abff8034"/>
        <textElement textAlignment="Center">
          <font size="11"/>
        </textElement>
        <text><![CDATA[1.0]]></text>
      </staticText>
    </band>
  </pageHeader>
  <detail>
    <band height="220" splitType="Stretch">
      <componentElement>
        <reportElement x="25" y="70" width="510" height="150" uuid="6af41e8e
            -7097-42e5-b240-8099a9ff13a7">
          <property name="com.jaspersoft.studio.layout" value="com.jaspersoft.
              studio.editor.layout.VerticalRowLayout"/>
          <property name="com.jaspersoft.studio.table.style.table_header" value="
              Table_TH"/>
          <property name="com.jaspersoft.studio.table.style.column_header" value="
              Table_CH"/>
          <property name="com.jaspersoft.studio.table.style.detail" value="Table_TD
              "/>
          <property name="com.jaspersoft.studio.unit.width" value="px"/>
        </reportElement>
        <jr:table xmlns:jr="http://jasperreports.sourceforge.net/jasperreports/
            components" xsi:schemaLocation="http://jasperreports.sourceforge.net/
            jasperreports/components http://jasperreports.sourceforge.net/xsd/
            components.xsd">
          <datasetRun subDataset="AlertsLastMonthDataset" uuid="af3b7e75-ccbe-4297-
              b69a-6a77ee0f0eb0">
            <dataSourceExpression><![CDATA[$P{alerts-last-month}]]></
                dataSourceExpression>
          </datasetRun>
          <jr:column width="240" uuid="a23af4c0-7b20-4e87-ae8d-f930873875d5">
            <jr:columnHeader style="Table_CH" height="15">
              <property name="com.jaspersoft.studio.unit.width" value="px"/>
              <staticText>
                <reportElement x="0" y="0" width="240" height="15" uuid="4e10f2f9-7
                    a78-4731-b8f1-867931e024bb">
                  <property name="com.jaspersoft.studio.unit.height" value="pixel"/
                      >
                  <property name="com.jaspersoft.studio.unit.leftIndent" value="px"
                      />
                </reportElement>
                <textElement>
                  <font isBold="true"/>
                  <paragraph leftIndent="5"/>
                </textElement>
                <text><![CDATA[Rule name]]></text>
              </staticText>
```

```
        </jr:columnHeader>
        <jr:detailCell style="Table_TD" height="15">
          <textField>
            <reportElement stretchType="RelativeToTallestObject" x="0" y="0"
                width="240" height="15" isPrintWhenDetailOverflows="true" uuid=
                "6a40af28-caa1-475a-bcc5-cb5f5926e64f">
              <property name="com.jaspersoft.studio.unit.leftIndent" value="px"
                  />
            </reportElement>
            <textElement>
              <paragraph leftIndent="5"/>
            </textElement>
            <textFieldExpression><![CDATA[$F{rule-name}]]></textFieldExpression
                >
          </textField>
        </jr:detailCell>
      </jr:column>
      <jr:column width="50" uuid="c7521f41-25f8-4297-9c6f-da0b2a2895f6">
        <jr:columnHeader style="Table_CH" height="15">
          <property name="com.jaspersoft.studio.unit.width" value="px"/>
          <staticText>
            <reportElement x="0" y="0" width="50" height="15" uuid="e6ea0926-2
                af8-48cb-a8ee-1a6789bf1fd3">
              <property name="com.jaspersoft.studio.unit.x" value="pixel"/>
              <property name="com.jaspersoft.studio.unit.leftIndent" value="px"
                  />
              <property name="com.jaspersoft.studio.unit.width" value="pixel"/>
            </reportElement>
            <textElement>
              <font isBold="true"/>
              <paragraph leftIndent="5"/>
            </textElement>
            <text><![CDATA[Severity]]></text>
          </staticText>
        </jr:columnHeader>
        <jr:detailCell style="Table_TD" height="15">
          <property name="com.jaspersoft.studio.unit.width" value="px"/>
          <textField>
            <reportElement x="0" y="0" width="50" height="15" uuid="39bec117-62
                f3-4fe7-b698-f62b25b86a15">
              <property name="com.jaspersoft.studio.unit.x" value="pixel"/>
              <property name="com.jaspersoft.studio.unit.leftIndent" value="px"
                  />
            </reportElement>
            <textElement>
              <paragraph leftIndent="5"/>
            </textElement>
            <textFieldExpression><![CDATA[$F{severity-label}]]></
                textFieldExpression>
          </textField>
        </jr:detailCell>
      </jr:column>
      <jr:column width="180" uuid="44ad0745-76fd-41b0-88ca-d31b7d6b2f1e">
        <jr:columnHeader style="Table_CH" height="15">
          <property name="com.jaspersoft.studio.unit.width" value="px"/>
          <staticText>
            <reportElement x="0" y="0" width="180" height="15" uuid="348953e4-9
                eda-4994-a7b5-b43af7c78a1b">
              <property name="com.jaspersoft.studio.unit.leftIndent" value="px"
                  />
            </reportElement>
            <textElement>
              <font isBold="true"/>
              <paragraph leftIndent="5"/>
            </textElement>
            <text><![CDATA[Rule category]]></text>
          </staticText>
```

```
          </jr:columnHeader>
          <jr:detailCell style="Table_TD" height="15">
            <textField>
              <reportElement x="0" y="0" width="180" height="15" uuid="c7a55b2f-
                  c3a3-4028-b3e1-80b66ff8005e">
                <property name="com.jaspersoft.studio.unit.leftIndent" value="px"
                    />
              </reportElement>
              <textElement>
                <paragraph leftIndent="5"/>
              </textElement>
              <textFieldExpression><![CDATA[$F{rule-category}]]></
                  textFieldExpression>
            </textField>
          </jr:detailCell>
        </jr:column>
        <jr:column width="40" uuid="67e77678-8cd9-4dbc-a04b-d2dd19cb7002">
          <jr:columnHeader style="Table_CH" height="15">
            <property name="com.jaspersoft.studio.unit.width" value="px"/>
            <staticText>
              <reportElement x="0" y="0" width="40" height="15" uuid="ff2c303f-
                  feaa-4a66-ab03-8a7bffde7e5d">
                <property name="com.jaspersoft.studio.unit.leftIndent" value="px"
                    />
              </reportElement>
              <textElement>
                <font isBold="true"/>
                <paragraph leftIndent="5"/>
              </textElement>
              <text><![CDATA[#]]></text>
            </staticText>
          </jr:columnHeader>
          <jr:detailCell style="Table_TD" height="15">
            <property name="com.jaspersoft.studio.unit.width" value="px"/>
            <textField>
              <reportElement x="0" y="0" width="40" height="15" uuid="96270465-45
                  e0-4071-b0aa-b3e0f212d6e2">
                <property name="com.jaspersoft.studio.unit.leftIndent" value="px"
                    />
              </reportElement>
              <textElement>
                <paragraph leftIndent="5"/>
              </textElement>
              <textFieldExpression><![CDATA[$F{num-occurrences}]]></
                  textFieldExpression>
            </textField>
          </jr:detailCell>
        </jr:column>
      </jr:table>
    </componentElement>
    <staticText>
      <reportElement x="90" y="0" width="380" height="30" forecolor="#00AD8E"
          uuid="f068e4d2-5872-41b9-960f-4ad068d7715a"/>
      <textElement textAlignment="Center">
        <font size="20"/>
      </textElement>
      <text><![CDATA[Alert data from last month]]></text>
    </staticText>
    <staticText>
      <reportElement x="30" y="30" width="490" height="40" uuid="e5bcd247-e61c
          -4062-a2f9-43937668db9a"/>
      <text><![CDATA[Table containing information about alerts from the last
          month. The alerts are grouped by the name of the rule, severity of the
          event, and category of the rule. The table is sorted on the number of
          occurrences for that particular group in descending order.]]></text>
    </staticText>
  </band>
```

```
<band height="360">
  <pieChart>
    <chart isShowLegend="true" evaluationTime="Report">
      <reportElement x="24" y="70" width="512" height="290" uuid="49b3128f-8cd3
          -4315-8503-fc2a209c9af8"/>
      <chartTitle/>
      <chartSubtitle/>
      <chartLegend position="Right"/>
    </chart>
    <pieDataset minPercentage="0.5">
      <dataset resetType="Report">
        <datasetRun subDataset="RuleCategoryDataset" uuid="00bad03b-adc8-464f
            -99f5-fe62c8db906b">
          <dataSourceExpression><![CDATA[$P{rule-category-occurrences}]]></
              dataSourceExpression>
        </datasetRun>
      </dataset>
      <keyExpression><![CDATA[$F{rule-category}]]></keyExpression>
      <valueExpression><![CDATA[$F{num-occurrences}]]></valueExpression>
      <labelExpression><![CDATA[$F{rule-category}]]></labelExpression>
      <otherKeyExpression><![CDATA[]]></otherKeyExpression>
      <otherLabelExpression><![CDATA[]]></otherLabelExpression>
    </pieDataset>
    <piePlot isShowLabels="false" isCircular="true" legendLabelFormat="{0}
        ({2})">
      <plot/>
      <itemLabel/>
    </piePlot>
  </pieChart>
  <staticText>
    <reportElement x="90" y="0" width="380" height="30" forecolor="#00AD8E"
        uuid="51d089aa-9ba9-4e95-9522-51872fa02063"/>
    <textElement textAlignment="Center" verticalAlignment="Top">
      <font size="20"/>
    </textElement>
    <text><![CDATA[Rule occurrences per category]]></text>
  </staticText>
  <staticText>
    <reportElement x="30" y="30" width="490" height="40" uuid="b4808819-20cb
        -459d-b5ed-855ea1a93346"/>
    <text><![CDATA[Pie chart containing information about the number of
        occurrences of rule categories in alerts. Each part of the chart
        represents the percentage of occurences of that particular category
        occuring in alerts. A distinct category is only displayed if it at
        least occupies 0.5% of the chart.]]></text>
  </staticText>
</band>
<band height="380">
  <pieChart>
    <chart evaluationTime="Report">
      <reportElement x="20" y="90" width="521" height="290" uuid="281b1eef
          -8346-433e-b198-3f041951006c">
        <property name="com.jaspersoft.studio.unit.width" value="px"/>
        <property name="com.jaspersoft.studio.unit.height" value="px"/>
      </reportElement>
      <chartTitle/>
      <chartSubtitle/>
      <chartLegend position="Right"/>
    </chart>
    <pieDataset minPercentage="1.0">
      <dataset resetType="Report">
        <datasetRun subDataset="CountryRuleDataset" uuid="cc2a0839-4a21-4b99-8
            dcb-74d9d3c4a132">
          <dataSourceExpression><![CDATA[$P{country-rule-occurrences}]]></
              dataSourceExpression>
        </datasetRun>
      </dataset>
```

```
        <keyExpression><![CDATA[$F{country-name} + ", " + $F{rule-category}]]></
            keyExpression>
        <valueExpression><![CDATA[$F{num-occurrences}]]></valueExpression>
        <labelExpression><![CDATA[$F{country-name} + ", " + $F{rule-category}]]><
            /labelExpression>
      </pieDataset>
      <piePlot isShowLabels="false" legendLabelFormat="{0} ({2})">
        <plot/>
        <itemLabel/>
      </piePlot>
    </pieChart>
    <staticText>
      <reportElement x="90" y="0" width="380" height="30" forecolor="#00AD8E"
          uuid="b8755fa8-2ebe-4b97-bea2-c7d7cce7f08a"/>
      <textElement textAlignment="Center">
        <font size="20"/>
      </textElement>
      <text><![CDATA[Alert occurences by country and rule category]]></text>
    </staticText>
    <staticText>
      <reportElement x="35" y="30" width="490" height="60" uuid="0a5372d1-af03-42
          e3-9d35-db5b4e42b695"/>
      <text><![CDATA[Pie chart containing information about the number of
          occurrences of alerts by their origin country and rule category. Alerts
           from the last month are grouped by the origin country and rule
          category and the number of occurences within that group is represented
          in the chart. A distinct group is only displayed if it at least
          occupies 1% of the chart.]]></text>
    </staticText>
  </band>
  <band height="370">
    <pieChart>
      <chart evaluationTime="Report">
        <reportElement x="20" y="80" width="521" height="290" uuid="525dc7d9
            -2891-4202-b8c6-bdb6da6fad66">
          <property name="com.jaspersoft.studio.unit.width" value="px"/>
          <property name="com.jaspersoft.studio.unit.height" value="px"/>
        </reportElement>
        <chartTitle/>
        <chartSubtitle/>
        <chartLegend position="Right"/>
      </chart>
      <pieDataset minPercentage="1.0">
        <dataset resetType="Report">
          <datasetRun subDataset="HostnameDataset" uuid="567799ae-28dc-4d14-a171
              -7c0c00d44442">
            <dataSourceExpression><![CDATA[$P{hostname-occurrences}]]></
                dataSourceExpression>
          </datasetRun>
        </dataset>
        <keyExpression><![CDATA[$F{hostname} + ", " + $F{destination-ip}]]></
            keyExpression>
        <valueExpression><![CDATA[$F{num-occurrences}]]></valueExpression>
        <labelExpression><![CDATA[$F{hostname}]]></labelExpression>
      </pieDataset>
      <piePlot isShowLabels="false" legendLabelFormat="{0} ({2})">
        <plot/>
        <itemLabel/>
      </piePlot>
    </pieChart>
    <staticText>
      <reportElement x="90" y="0" width="380" height="30" forecolor="#00AD8E"
          uuid="1a4146d7-f461-42d0-ad5b-c9c8c3d93ef1"/>
      <textElement textAlignment="Center">
        <font size="20"/>
      </textElement>
      <text><![CDATA[Targets of alerts by hostname]]></text>
```

```
      </staticText>
      <staticText>
        <reportElement x="35" y="30" width="490" height="50" uuid="86d1a6ed-3c31-41
            d9-a7e4-0f73e43f6abf"/>
        <text><![CDATA[Pie chart containing information about the number of
            occurrences of alerts by their hostname. Alerts from the last month are
             grouped by the destination hostname and IP, and the number of
            occurences within that group is represented in the chart. A distinct
            group is only displayed if it at least occupies 1% of the chart.]]></
            text>
      </staticText>
    </band>
  </detail>
  <pageFooter>
    <band height="62">
      <printWhenExpression><![CDATA[$V{PAGE_NUMBER} != 1]]></printWhenExpression>
      <staticText>
        <reportElement x="230" y="20" width="99" height="19" uuid="86188a6a-31f2
            -4139-adb4-ae8756370f14"/>
        <textElement textAlignment="Center">
          <font size="11"/>
        </textElement>
        <text><![CDATA[Draft]]></text>
      </staticText>
      <staticText>
        <reportElement x="20" y="20" width="99" height="19" uuid="a2b7742f-18bd-4
            efa-8c79-7b7d815fdcd1"/>
        <textElement textAlignment="Left">
          <font size="11"/>
        </textElement>
        <text><![CDATA[10-12-2021]]></text>
      </staticText>
      <textField>
        <reportElement x="440" y="20" width="99" height="19" uuid="296b6657-2cc7
            -409e-86db-f1ddf8f1b5ad">
          <property name="com.jaspersoft.studio.unit.width" value="px"/>
          <property name="com.jaspersoft.studio.unit.height" value="px"/>
        </reportElement>
        <textElement textAlignment="Right">
          <font size="11"/>
        </textElement>
        <textFieldExpression><![CDATA[$V{PAGE_NUMBER}]]></textFieldExpression>
      </textField>
    </band>
  </pageFooter>
</jasperReport>
```

Listing A.6: Showcase external JasperReports template file

PDF output file

Monthly update document for [customer]
containing figures related to network monitoring.

Date	Version	Status	Classification
10-12-2021	1.0	Draft	ESH-SOC internal

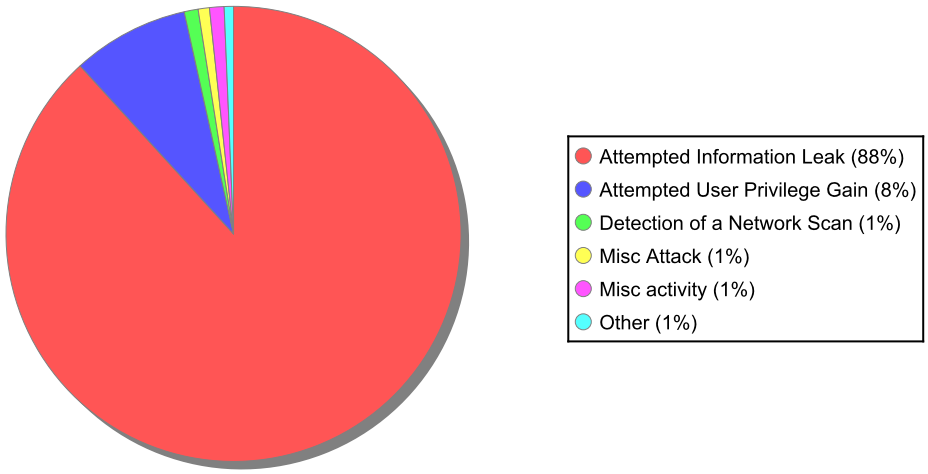
Alert data from last month

Table containing information about alerts from the last month. The alerts are grouped by the name of the rule, severity of the event, and category of the rule. The table is sorted on the number of occurrences for that particular group in descending order.

Rule name	Severity	Rule category	#
ET SCAN Potential SSH Scan	medium	Attempted Information Leak	11669
ET RPC DCERPC SVCCTL - Remote Service	high	Attempted User Privilege Gain	1159
ET SCAN MS Terminal Server Traffic on Non-	medium	Attempted Information Leak	579
ET SCAN Zmap User-Agent (Inbound)	low	Detection of a Network Scan	119
ETPRO EXPLOIT D-Link DCS-2530L	medium	Misc Attack	98
ET SCAN Behavioral Unusual Port 445 traffic	low	Misc activity	72
ET SCAN Behavioral Unusual Port 135 traffic	low	Misc activity	71
ET SCAN Laravel Debug Mode Information	medium	Attempted Information Leak	48
ETPRO USER_AGENTS Observed Suspicious UA	medium	Potentially Bad Traffic	35
ET SCAN NETWORK Incoming Masscan detected	low	Detection of a Network Scan	16
ET SCAN ProxyReconBot CONNECT method to	medium	Misc Attack	14
ET WEB_SERVER /bin/sh In URI Possible Shell	high	Web Application Attack	10
ET WEB_SERVER PyCurl Suspicious User Agent	medium	Attempted Information Leak	10
ET SCAN NMAP SIP Version Detect OPTIONS	medium	Attempted Information Leak	9
ETPRO WEB_SPECIFIC_APPS ipTIME firmware <	high	Web Application Attack	9
ETPRO HUNTING Generic Inbound URI Directory	medium	Potentially Bad Traffic	8
ET SCAN Nmap Scripting Engine User-Agent	high	Web Application Attack	6
ET SCAN ZmEu Scanner User-Agent Inbound	high	A Network Trojan was detected	6
ETPRO EXPLOIT SChannel Possible Heap	high	Attempted Administrator Privilege Gain	4
ET SCAN WordPress Scanner Performing Multiple	low	Detection of a Network Scan	3
ET WEB_SERVER PHP Easteregg Information-	medium	Attempted Information Leak	3
ET WEB_SERVER WEB-PHP phpinfo access	medium	Information Leak	3
ET CHAT MSN status change	high	Potential Corporate Privacy Violation	2
ET EXPLOIT Possible OpenSSL HeartBleed Large	medium	Potentially Bad Traffic	2
ET EXPLOIT SSL excessive fatal alerts (possible	medium	Attempted Information Leak	2
ET WEB_SERVER PHP Easteregg Information-	medium	Attempted Information Leak	2
ET WEB_SPECIFIC_APPS Amateur Photographer	high	Web Application Attack	2
ETPRO WEB_SPECIFIC_APPS WP Mobile Edition	high	Web Application Attack	2
ETPRO WEB_SPECIFIC_APPS WP Theme LFI	high	Attempted Administrator Privilege Gain	2
ET EXPLOIT Apache HTTP Server - Path Traversal	high	Attempted Administrator Privilege Gain	1
ET SCAN Behavioral Unusual Port 139 traffic	low	Misc activity	1
ET WEB_SERVER DFind w00tw00t GET-Requests	medium	Attempted Information Leak	1
ET WEB_SERVER Muieblackcat scanner	medium	Attempted Information Leak	1

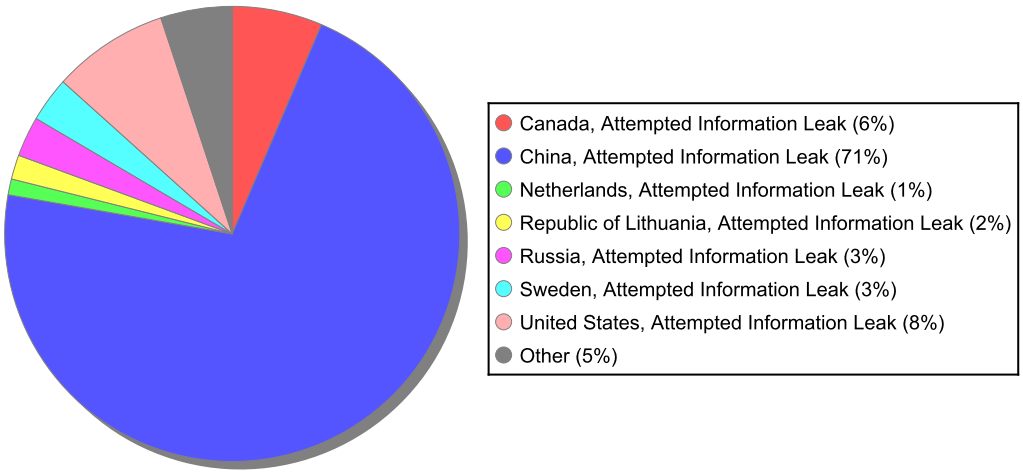
Rule occurrences per category

Pie chart containing information about the number of occurrences of rule categories in alerts. Each part of the chart represents the percentage of occurences of that particular category occuring in alerts. A distinct category is only displayed if it at least occupies 0.5% of the chart.



Alert occurences by country and rule

Pie chart containing information about the number of occurrences of alerts by their origin country and rule category. Alerts from the last month are grouped by the origin country and rule category and the number of occurences within that group is represented in the chart. A distinct group is only displayed if it at least occupies 1% of the chart.



Targets of alerts by hostname

Pie chart containing information about the number of occurrences of alerts by their hostname. Alerts from the last month are grouped by the destination hostname and IP, and the number of occurences within that group is represented in the chart. A distinct group is only displayed if it at least occupies 1% of the chart.