Task Description

Analysing security-related events is a crucial task in today’s computer and network surveillance in order to detect and analyse malicious intrusions. However, detecting single events are often not sufficient to reliably identify hacking attacks. Temporal or causal related chains of events are more sophisticated metrics for detecting intrusions. Today, Security Information and Event Management systems (SIEM systems) are state-of-the-art in order to acquire, detect, alert and analyse security-related events on a large amount of IT devices.

The goal of this project is to research and create a prototypical implementation of such a SIEM system with the help of the Complex Event Processing (CEP) paradigm.

Task of first week

Goal

The goal for the first week is to install and configure the required toolchain for the entire project as well as getting started with the Complex Event Processing engine.

Prerequisites

For the project work we require a running GNU/Linux environment such as a recent Ubuntu version. Although Esper might run under different OSs, we do not support any other environment. Furthermore, we need a working SSH daemon running on the machine as well as a local user having access to the host via SSH.

As the Esper engine is written in Java, a working Java development environment is necessary for the further project work. It is recommended to run the engine on Java 11. The CEP engine itself can be downloaded from the Esper homepage[1]. However, it is recommended to use a build management tool for Java—such as Maven—to easily manage upcoming software dependencies.

The Esper CEP engine has a comprehensive documentation which can be also found on the website[2]. Besides a step-by-step “Getting started” guide including a short introduction into Complex Event Processing there is documentation about the engine itself as well as descriptions for the Event Pattern Language (EPL). Also, the Solution Patterns[3] are a good source for finding EPL statements for various situations.

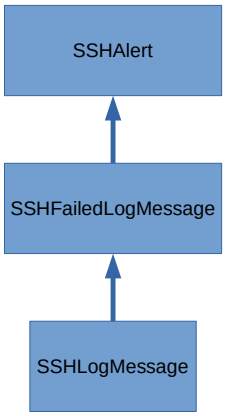
It is highly recommended to follow the “Getting Started” as well as the “Basic Concepts” sections of the documentation to test if the development setup is working properly and to learn how to define new event types, create EPL statements and send events to the CEP engine.

Task: Alerts of failed SSH logins

The goal is to develop a minimal SIEM system which is capable of monitoring incoming SSH connections and show alerts if a failed login appears. Furthermore, after a certain amount of consecutive failed logins an alert of a potential threat is shown. The alerts raised by the system should be shown to the user on the command line.

1 https://www.espertech.com/esper/esper-downloads/
2 https://www.espertech.com/esper/esper-documentation/
3 https://www.espertech.com/esper/solution-patterns/

Below the intended Event Hierarchy for the Complex Event Processing is depicted. From bottom to top, it describes the abstraction levels of the SSH events. As raw event type “SSHLogMessage” is defined. Detected failed logins should be events of type “SSHFailedLogMessage”. If a defined threshold of failed logins is detected, an event of type “SSHAlert” should be raised.



To monitor incoming SSH connections, the logging entries of the SSH daemon (`sshd`) can be used. In recent Linux systems, logging is handled by `journald`, a logging server of the systemd project. To access the `sshd` logging entries, issue the command

```
journalctl -u ssh.service
```

To get a machine readable output format, extend the command line with `-o json`, i.e.

```
journalctl -u ssh.service -o json
```

The produced output can be read with in a Java program with the help of a JSON parsing library such as `org.json`[4]. Parse the appropriate messages within the JSON structure for each log entry to find failed logins and their corresponding meta data such as date and time of the connection. Therefore, define a proper data structure in Java which is accepted as event type by the Esper engine, e.g. a Hash Map or a POJO.

Now feed the log messages into the CEP engine as raw events of event type “SSHLogMessage” and create events using appropriate EPL statements to implement the Event Hierarchy shown above. Keep in mind, that an event of type “SSHAlert” should only appear if a certain amount of consecutive failed logins (which you can define by yourself or can be defined by the user) took place.

Submission

The submission takes place in the Moodle course.

The submission is comprised of three parts: the source code of the task described above as well as a document and a presentation. The document should be two pages at maximum. Regarding the presentation, each question should be answered on one slide. Both the document and the presentation have to cover the following questions:

1. Describe the event type of “SSHLogMessage”. Which event representation did you use and why? Also, show and describe the corresponding Java code snippet and EPL statement.

4 https://search.maven.org/classic/#search%7Cgav%7C1%7Cg%3A%22org.json%22%20AND%20a%3A%22json%22

2. Describe the EPL statement(s) for raising “SSHFailedLogMessage” events in detail. What are the necessary language constructs to create the event out of the raw event type? Which properties of the raw event did you include and why?
3. Describe the EPL statement(s) for raising “SSHAlert” events in detail. What are the necessary language constructs to create the event. What are the necessary temporal/occasional language constructs in order to achieve the threshold logic?
4. Explain the parsing component for extracting log entries from the SSH daemon. Explain your decisions for using certain data structures with respect to the event representations within the CEP engine.
5. How did you realise the output of the alerts to the user? Especially, explain the interaction between the CEP engine and Java for showing the alerts.
6. Explain the essential ideas of SCRUM. How do you want to apply them on the project?

Each participant has to be able to answer and present the questions in the meeting. That is, each student will be asked to talk about one of the questions above and she/he should be able to answer the question by presenting the corresponding slide in the presentation.