Question 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.

We did use a POJO this is a normal java calls similar to the documentation and the epl statement is also near identical to the documentation. The only thing that is changed is that the documentation is the variables are changed to the log message and timestamp so we have a pre formated timestamp and the message containing all relevant infos like the offending user, ip and port

Question 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?

I discarded all not further used information before hand so we basically raised a new event with identical properties to the SSHLogMessage Event. Upon recieving the ssh log event, I than create a new Event like before and go on from there.

Question 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?

I did choose to create a new full Event and fire an event by counting the amount of raised ssh filed log messages and than raising a SSHAlert event when our non resting threshold is matched.

Question 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.

We read the journalctl output line by line. We achive this by piping the into a file in a json format. We than read the file line by line and treat each line as an object. Gson is used to turn out line into an object we can handle in java. We use an intermediate "GenericLogObject" class to deserialize the item into a class. From here on out we than can create the evenet. At this point we also discard all irelevant statements. All relevent log items will create an SSHLogMessage event object. I chose to only discord info like the messageID or unix timestamp since they might not be directly relevant to the user. In case of the messageID since we do not at this point to realtime system survailance, we don't have yet to keep track of old messages. The timestamp might be more comfortable to use for a programmer and allows to be turned into diffrent types of output formats even diffrent timezones but since the feature was not requested and

therefore adds unneded complexity the idea was discarded. After this point we always consider Message and the timestemp as relevent items to keep for out events and log. We could consider keeping the message id to continously read out the log items but this is out of scope of the project at this point.

Question 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.

The outputs are done through a console output, I did choose to not use slf4j since I'd have to deactivate quite a few things in the output since parts of the log might be irelevant to the user and therefore making it harder for the user of the program to recognize important information. We chose to just use System.out.println without many major formating. Except ANSI color, since there might be a lot of data on a screen at once even though not asked for it's a major feature without adding much complexity to the project. Besides that we chose to have out output for Alerts going over multiple line since that can attract more attention to the Problem. We interact with the CEP engien by creating events and listening for these events, on reciving an event of the type SSHLogEvent we check if the event matches the parameters for a failes login attemp. In that case we push out a new even of the Type SSHFailedLogMessage and log the failed event in the command line output. For events of the type SSHAlert we act near identical but we also use a singleton patter to take count of the amount of events that took place and to store the variable of how many SSHFailedLogMessage events we have as a threshold. Then we check if the amount of relevant events is higher than the threshold each time we have a relevant event (please not that this system might have an concurency issue)

Question 6

Explain the essential ideas of SCRUM. How do you want to apply them on the project?

In scrum you would work out user stories which descibe the features we want in out project. User stories should be as minimal as possible so out features have an atomic charater. The combined amount of user stories represents a so called backlog. This backlog than serves as a basis for for the so called sprints which are usually 14-30 day intervals where a certain amount of user stories are picked out of the backlog. The person picking the features for the next sprint is the product owner, who decides the required featureset for their next release of the product. Another major part of Scrum are short daily in person standup meetings where you talk about the goals of the day. The major goal of SCRUM is to deliver high value software in a short period of time aswel as having a so called release kandiate candidate after each sprint. I'd like to implement the daily meetings aswell as the user stories principles implemented in my project team since it keeps you up to date with the other people in the project. Besides that you have well defined problems to work on. Sprints are great way to people working on the project and to set a sense of urgency but might be a bit more preasure since at least from my part, I have a few other projects going on simultaniously therefore I can only work in bursts on each project if I were to keep efficency high.