The task - A simple rule engine

The task consists of creating a simple rule engine. A user should be able to upload any number of dependency files (we have provided some example dependency files together with this document) at once, with possible errors dealt with gracefully. The uploads should be done using a RestAPI or GraphQL. You should then forward these uploads to Debricked's open API (https://debricked.com/api/doc/open) and start a scan. Hint: all the API endpoints you need are under the "Dependency files management" category.

When the scan has been completed you should automatically (hint: Symfony Command run in the background inside the docker container) notify the user based on a set of rules which he/she has set. The rules should consist of triggers and actions that result from those triggers. The triggers could include:

- Amount of vulnerabilities found during a scan/upload is greater than X - Upload is in progress
- Upload fails for some reason

The resulting actions could include:

- Send an email to user
- Send a message to a Slack channel

No UI is required for the task. The communication with API for uploading can be done through e.g. postman and the enabling/disabling of rules and their configuration can be done through environment variables, or an API endpoint.

Note: To communicate with the API you need to create an account on https://debricked.com/app and use your credentials to generate a JWT.

Requirements

- Use the latest Symfony version as base. Feel free to utilise whatever libraries/bundles you want to complete the task. Symfony's Messenger and Notifier components might be especially useful.
- The database structure and data should be loaded at first startup and then be re-used.

Estimated time

You should spend roughly one working day = 8 hours

Runtime environment

PHP is an extensible language and for that reason it would be a shame if you were limited to only a specific subset of available extensions. At the same time it would be hard for us to try your code if it were written with a lot of extensions which we do not use. For that reason, and that we use it internally, we would prefer you to utilise a Docker container for running your code.

You are free to utilise any pre-existing Docker image, such as https://github.com/debricked/docker-image-php-xdebug, or build your own. You are also allowed to use multiple Docker containers if you see a need, but please utilise Docker Compose if you do so.

You will be judged on

- Code quality (not quantity)
- Test coverage and quality
- Documentation quality (PHPDocs are fine, no need to generate any documentation) To a lesser degree we will also look at
- Any possible Git use
- Any customisations you may have made to the runtime environment

Submission

Send either a link to a public repository containing your submission (code and a dockerfile) or zip it and send it to oscar.reimer@debricked.com. The submission should contain a short readme on how to run and configure the application.

Questions?

Feel free to contact us at any time at oscar.reimer@debricked.com or james.godfrey@debricked.com. We will answer general questions and may give some hints if you get stuck. At Debricked we always try to help each other out if we get stuck.