OUTLINE OF THE DEMO

The steps shown in the video are outlined in the following.

Step 1. Cloning the project "KB" GitHub repository, https://github.com/SAP/project-kb.
Step 2. Execution of the script `run_prospector.sh` from the `prospector` subfolder.
The script automatically builds and starts all the necessary docker containers.
Step 3. The command line flags are shown on the screen; for the demo, we use the strictly required inputs only, which are: *(A)* a vulnerability identifier and *(B)* the URL of the source code repository of the project affected by the vulnerability.
Step 4. As illustrative example, Prospector is executed on *CVE-2020-1925* and the *Apache Olingo* repository. As the tool runs, we give a high-level explanation of the processing it performs (advisory record extraction, candidate commits retrieval and processing, rule application, report generation).
Step 5. The report generated at the end of the previous step is shown and its key elements are described.
Step 6. We highlight the fact that the advisory content is processed to extract important tokens (keywords, file names, etc.).
Step 7. We explain that commits are ranked by their relevance, which is computed by applying a set of rules to each of them. The sum of the weights of the rules that match a commit determine its relevance. The list of commits shown in the report can be filtered by a applying a relevance threshold using a slider.
Step 8. As a concrete example, we point out that the tool detected that the first commit in the list modifies a class that is mentioned in the textual description of the advisory.

TOOL DOWNLOAD

The tool described in the submission (Prospector), is available for download from the GitHub repository of project "KB", in the context of which it is developed and maintained:

https://github.com/sap/project-kb

The tool is released under the Apache 2.0 license to encourage 3rd party contributions and wide industry adoption.