DepVis - Guided tour

Thanks for participating! DepVis is a visualization tool that lets you explore software dependencies and vulnerabilities represented in a graph form. It is language independent and one of its goals is to help security specialists and developers improve security of their products.

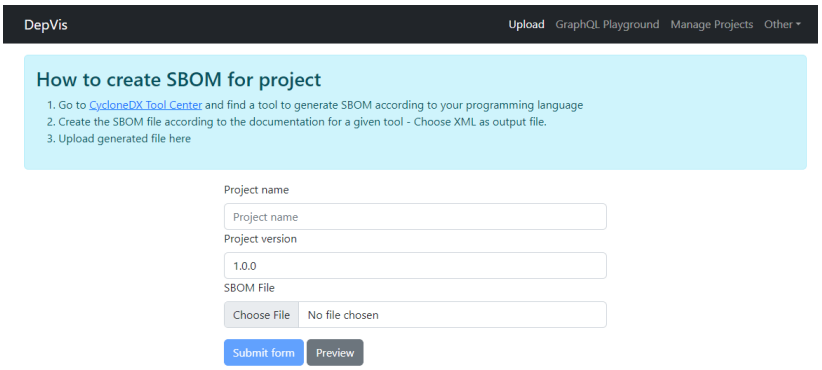
You will be asked to follow several tasks to get familiar with the application and to try it out. Let's get started.

You can find a working application at: http://depvis.eastus.azurecontainer.io:3000/. In case of any issues let me know.

Uploading file

First let's upload input data. It is a Software Bill of Materials in CycloneDX format - a XML file representing packages and dependencies of the application. If you feel like it, you can generate your own or use some of the public ones. Or you can use prepared examples that can be found here: https://drive.google.com/drive/folders/15jK5TJ0R-DPnMof1kGjzz6k_hkeJzzHz?usp=sharing

1. Go to "/upload", you will see following form

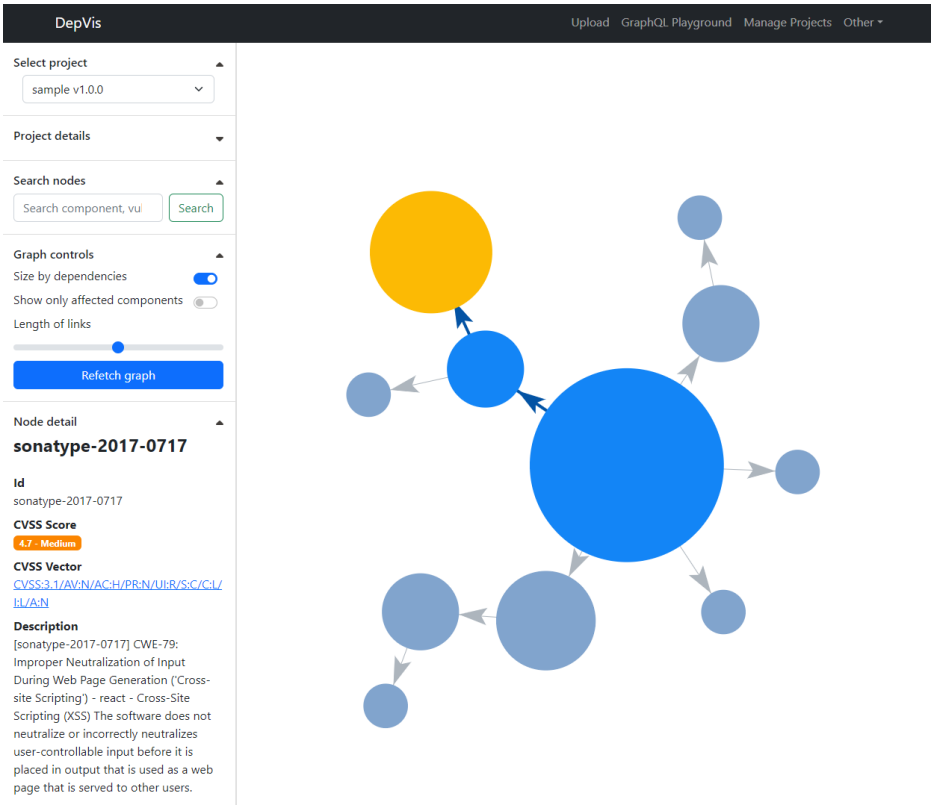


2. Choose your project name and version
3. Select file containing your project and submit form

You will now see the import progress. Application is parsing the XML and looking for vulnerabilities associated with the packages present in the project. You can go to the main page, the process will continue in the background.

Exploring graph

This is the main feature of DepVis. After successful import you can see the project represented as nodes and links. Visualization workspace is present on the root page.



Screen is divided into two parts, on the left side is a sidebar, where you can select projects, see statistics for the selected project, search for components, control graph properties and see details for clicked nodes.

Try following actions:

- Use mouse to interact with the graph
    - Zoom in/out
    - Pan
    - Drag nodes
- Click on a node to see details in the sidebar
- Notice a highlighted path from selected node to the main one

You should be able to use DepVis to answer the following questions.

1. How many components are present in the sample project? How many of them are direct and transitive dependencies?
2. Is there a vulnerability in any of the components? In which ones?
3. Which component should be updated in order to mitigate the vulnerability?

Now please proceed with filling the feedback form. Thank you for your time.