NebulaGraph

Graph Your Code Vulnerabilities

Presenter: Goran Cvijanovic

NebulaGraph

Project scope

Analyze GitHub projects structure

Collect project elements and dependencies

Detect libraries and code and store as graph

Verify exposure to CVE vulnerabilities

Inspiration

When log4j vulnerability CVE-2021-44228 exposes many Java servers to the possible exploration, there should be the easy way to validate for our software are we exposed

Why we wouldn’t be able to visualize our software structure as graph of dependencies and analyze which components are exposed to CVE security issues

Dependencies to other repositories, libraries and projects increase complexity of the problem, but can be easy represented and visualized using graph technology

Security Public Information

Common Vulnerability Exposure

The mission of the CVE® Program is to identify, define, and catalog publicly disclosed cybersecurity vulnerabilities

Currently, there are 166,475+ CVE Records accessible

National Vulnerability Database

Repository of standards based vulnerability management data

The NVD performs analysis on CVEs that have been published

Common Platform Enumeration - CPE

Common Vulnerability Scoring System - CVSS

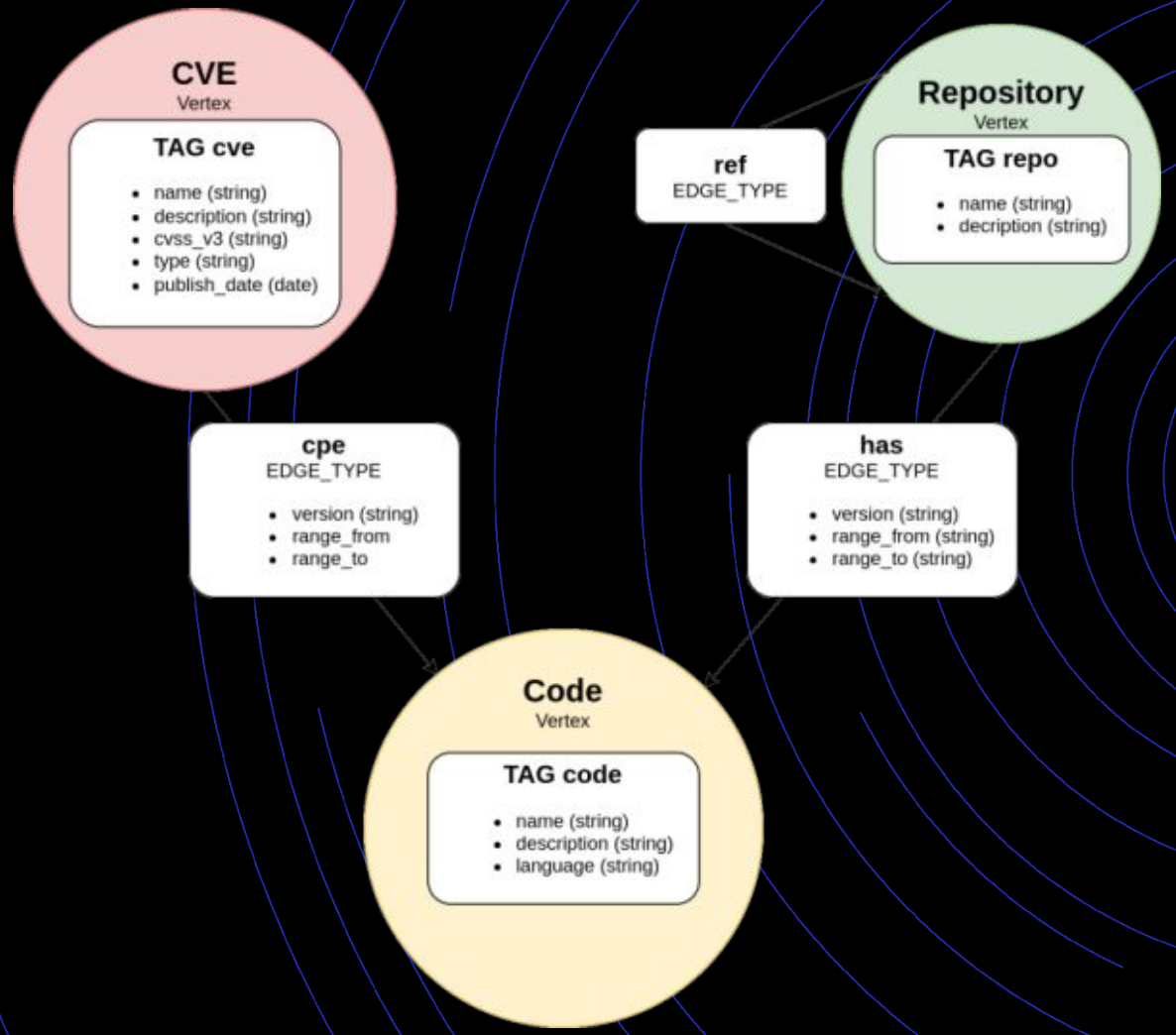
NebulaGraph

Using Graph Models

Nebula Graph Database

Simple data model to organize and store data from security area

Repositories and libraries are hierarchical and it can be hard to cover all dependencies but they are forming natural graph structure



NebulaGraph

Implementation

- **Python code**

- Custom python code for web application service

- Nginx web server

- Nebula Graph database
  integrated with Elasticsearch engine

- Javascript visualization library

- **Data collector for CVE**

Custom python code integrated with CVE and CPE sources about new vulnerabilities detected

NebulaGraph

Thank you

NebulaGraph