Camden Kronhaus & Panat Taranat

What is the project

We are proposing the design and development of an automated security auditing system for Docker containers. The system will conduct vulnerability analyses of container images based on CVE entries and user-defined custom policies. The input is a container image and policies, and the output is a report of vulnerability details, CVE identifiers, threat level, and other details.

Why is tackling this project important

Docker is used to package, send, and run many applications within containers on user machines and cloud servers. According to Docker, there are over 3 million desktop installations of Docker, and 242 billion Hub pulls[1]. Twenty-five percent of companies have adopted Docker and the adoption rate is increasing each year[2]. By running apps in containers, developers can provide higher portability and control over deployed applications. However, if there is a vulnerability within a container image, any container built from that image can be affected, potentially affecting users at a large scale. Automating the security auditing of container images is a major benefit for users, allowing teams to catch vulnerabilities early in the build pipeline.

How do you plan on tackling the project

Week 1: This week will involve researching common vulnerabilities affecting Docker images and programs that typically run in containers. We will inspect and test open source container security tools that scan container images and its contents. The result will be an artifact document containing our findings. We will look at projects like Docker Bench, Anchor, and Clair to understand how container security testing is typically done.
Week 2: We will develop a proof of concept, built on Week 1 research. Given a container image, the tool will reveal the container's contents and compare the contents to the CVE database.
Week 3: We will containerize our tool such that it can be embedded as part of a build/test pipeline, and run alongside other containers. We will demonstrate its usefulness by deploying it on publically available Docker hosts on Shodan. We will show how these Docker hosts were vulnerable, and model similar systems to show the exploits themselves.
Week 4: We will interpret our data from field-testing on Shodan, and improve our tool based on our findings.

List of 3 sets of deliverables

- **Passing**: A docker container with a generic mechanism that can automatically scan container images, reveal its contents, and generate a report, based on CVE identifiers that match the contents. The tool is useful for small projects and educational purposes.

[1] https://www.docker.com
[2] https://www.datadoghq.com/docker-adoption/

- **A**: The functionality of the tool is improved after testing on Shodan, and presentation of large scale data. A CLI for the tool that can perform a vulnerability scan or provide information on an image. The tool is useful for medium-sized projects.
- **A+**: A GUI and a REST API to allow higher control over the configuration and deployment of our tool. The tool is useful at enterprise level.

Github Repository

https://github.com/camdenkr/EC700_Epoch1