Konstantinos Papadopoulos
Jeremy Mwenda
Rahul Sharma
Renqing Gao
Mentor: Sastry S Duri

Container Safety Determination

Project Proposal

Vision and Goals of the project

Container safety determination project is aimed at providing feedback to engineers about the safety of the code running within the containers. Container technologies such as docker have made it easy to deploy new applications, maintain and upgrade them. To deploy a set of services, one needs to pull a container image from registry, wire them together, push back to the registry and then deploy in their environments. But how can a cloud provider who provides the infrastructure where they are running, or application developer who deployed them be sure that the containers are not running some malicious code? This project aims to propose a solution to this real-world problem.

Scope and features of the project

This project provides a way to verify if a docker image is safe to run on not. We will provide an API through which users can verify whether an image is safe. The user can submit an image or an sdhash of the image, and our system will check and tell the user whether the image is suspicious or not. Our solution can also be used to crawl a docker registry to check if the images contained in the registry are safe. This can be useful for organizations that own private registries. We will provide a tool that will monitor a registry for newly uploaded container images and verify them.

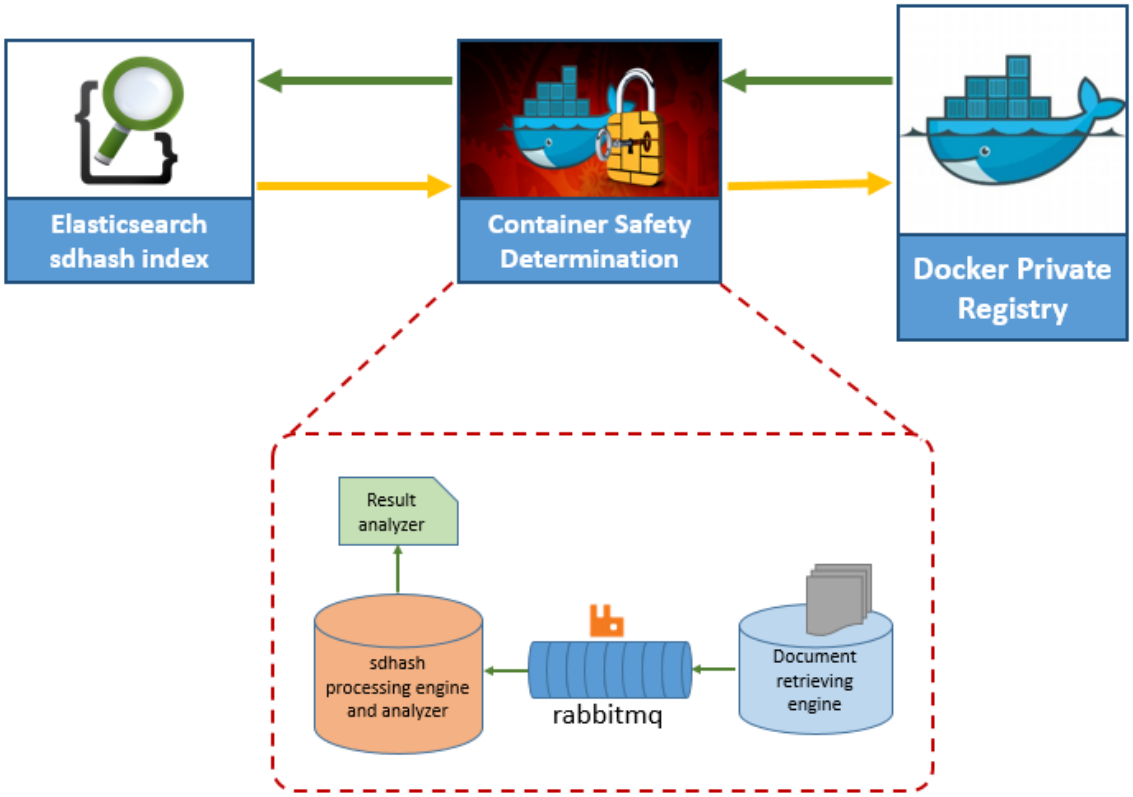
Solution concept

To determine if a piece of software is suspicious or not, we will check it against a reference data set provided by the National Software Reference Library (NSRL). This data set contains pre-computed sdhash values of applications or code that may be considered malicious, such as steganography tools and hacking scripts.

sdhash is a similarity digest hash. The Hamming distance of two sdhashes gives an estimate of the similarity between the files. An sdhash tool is available for calculating similarities sdhashes. The similarity is expressed as a score between 0-100. According to the sdhash guide, this score should be interpreted as a confidence value that indicates how certain the tool is that the two data objects being compared have non-trivial amounts of commonality.

We will download and store the pre-computed sdhashes into an elasticsearch server. Elasticsearch is a search server built on Lucene. It indexes text data and enables fast search on the data. Our container code classification will scan through the files in a docker container image, will compute the hash of each file and will search for matching or similar hashes in elasticsearch to determine whether the file contains suspicious code.

Design overview



Docker Registry: We will be creating a private Docker registry. This will be an example of a docker environment which needs to be scanned for suspicious code.

Elasticsearch server: An index with pre-computed sdhashes of known suspicious software.

Container Safety Determination: Application to verify the safety of a docker image. This application will have three components, which depending on scale can run on separate VMs.

1. Document retrieving engine to fetch container images from docker registry, unpack them and compute sdhashes of their contents.
2. RabbitMQ: Message queue to hold sdhashes produced by the Document retrieving engine.
3. sdhash analyzer: Get sdhashes from RabbitMQ and verify them against elasticsearch index.

The Container Safety Determination will also expose a REST API that can be used to verify individual docker images.

Minimum acceptance criteria

A python script that consults elasticsearch index before starting a docker container. If an image has suspicious software it would fail with a warning message.

A service exposed through REST API that allows experts to check if some software is suspicious.

Release planning

Sprint 1: Jan 24 - Jan 31
Setup elasticsearch server.
Install docker.
Setup RabbitMQ

Sprint 2: Feb 1 - Feb 14
Setup a private docker registry.
Download pre-computed sdhashes, index and store in elasticearch
Parse a docker image and retrieve files in the image.

Sprint 3: Feb 15 - Feb 28

Python code to calculate sdhash of a file.
Python code to compare a sample sdhash with sdhashes in elasticsearch. Need to figure out an efficient way of searching for similar hashes in elasticsearch.

Sprint 4: Mar 1 - Mar 14
Automatically fetch an image from a registry and verify it against elasticsearch data
Script to monitor a registry for new images.

Sprint 5: Mar 15 - Mar 28
REST API

Sprint 6: Mar 29 - April 10
Sprint 7: April 11 - April 24