SD MAY 20-58: K8S CHECKMATE

Daniel Brink, Jacob Cram, Sean Sailer, Alex Stevenson, John Young — Workiva
Julie Rursch — Workiva
2020

Problem Statement

Companies such as Google, Slack, and Shopify all rely on Kubernetes containerized environments to run their applications. As containerized deployment of applications becomes more and more popular, so too does their use as attack vectors by malicious parties. An incorrect configuration of these containers can lead to exploitable code, private network access, or malicious image injections that can shut down the application, and even extract sensitive information of its users.

Recent security breaches, such as the Equifax breach of 2017, are usually caused by a very small vulnerabilites that are overlooked due to a lack of knowledge or experience, and they result in the destruction of a system, or even worse, a leak of sensitive information. These breaches lead to a mistrust in technology and the companies who develop them. These breaches cost companies millions in lost revenue, millions in legal fees, and an immeasurable amount in consumer trust.

Solution

Our solution to the problem of containerized security is K8s Checkmate, a simple-to-use, lightweight CLI program that allows a user to verify their Kubernetes configuration meets a pre-defined ruleset.

The user gets real-time visual feedback on their security policies, and if they fail, what caused them to fail. Additionally, log files of the results are generated and stored to a user-specified directory. These log files allow the user to analyze historical results, as well as perform big-data analysis on how their security has evolved over time.

Users

- Developers working with Kubernetes containerized environments.
- System administrators who deal with log files

Uses

- Enforce user-defined security policies for Helm charts and values.
- Analyze historical outputs via log files for record-keeping, analysis, or debugging purposes.

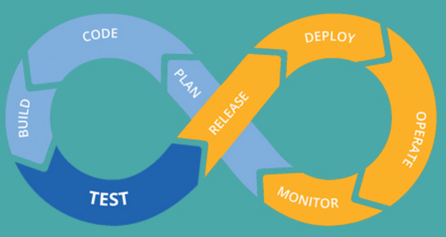
Technical Details

Python

Command-Line Interface

Open-Source

Continuous Integration



Engineering Standards & Design Practices

In our project, we used many engineering standards and design practices that are reccomended and followed by the software community. We followed the Agilr Scrum methodology, along with Test-Driven Development (TDD).

- Agile (12207-2017 IEEE System life cycle process)
- Unit Testing (1008-1987 - IEEE Standard for Software Unit Testing)

Design Requirements

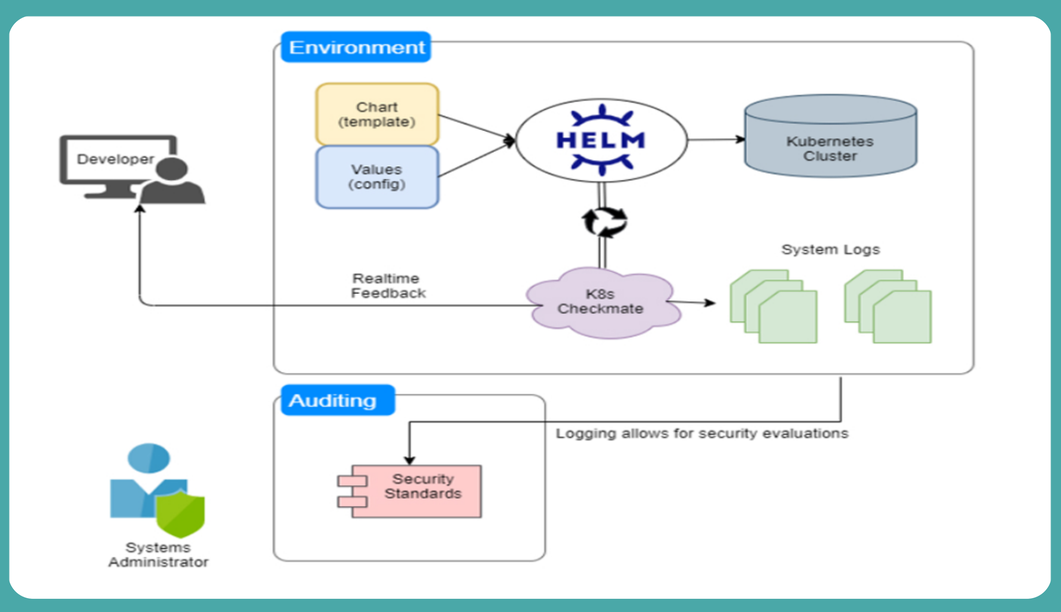
Functional Requirements

- Parse Helm charts and check them against a user's security policy
- Provide real-time feedback on policy status and provide feedback on how to fix failures
- Generate log files for record-keping and long-term analysis
- Compatible with *NIX systems, such as Linux, MacOS, etc..

Non-Functional Requirements

- Fast execution time relative to the size and contents of chart and policy files (< 1 second).
- Lightweight footprint that allows for a quick install, regardless of network performance (< 1 MB).
- Extensive documentation for new users, including setup and execution of the application.
- Must support verification of multiple files per run

Design Approach



Design Approach

```
Finished parsing "Project/TestCharts/test1.yaml" with security policy in 0.029476165771484375 seconds
Policy "MAX_OPEN_PORTS" PASSED
Policy "BANNED_PORTS" FAILED
        Found Banned Artifact: [81]
        Banned List: [80, 81, 82]
Policy "NO_ROOT" FAILED
        Expected: True
        Was: False
Policy "BANNED_IMAGES" PASSED
Policy "BANNED_USERS" FAILED
        Found Banned Artifact: [1000]
        Banned List: [1081, 91, 1000]
Successfully wrote output to file: Project/Output\k8scheckmate_2020-04-26_21_14_42.txt
Finished parsing "Project/TestCharts/test2.yaml" with security policy in 0.030471324920654297 seconds
Policy "MAX_OPEN_PORTS" PASSED
Policy "BANNED_PORTS" PASSED
Policy "NO_ROOT" PASSED
Policy "BANNED_IMAGES" FAILED
        Found Banned Artifact: ['bninedge']
        Banned List: ['bninedge']
Policy "BANNED_USERS" PASSED
Successfully wrote output to file: Project/Output\k8scheckmate_2020-04-26_21_14_42.txt
```