Computer Systems and Networks 23W

Topic Proposal: Security in Container Environments

Group Work

2023-11-04

Contents

1 Team Members

David Unterholzner (Mat. Nr.: 12009492, d.unterholzner@student.tugraz.at)
Jakob Hofer (Mat. Nr.: 12030367, jakob.hofer@student.tugraz.at)
Jakob Khom (Mat. Nr.: 12025213, jakob.khom@student.tugraz.at)
Leo Lach (Mat. Nr: 12014257, leo.lach@student.tugraz.at)

2 Topic Description

In our presentation, we will talk about Security in Container Environments. To give the audience a complete overview of the concept of containerization and the Security aspects behind it, we decided on the following topics that we will discuss:

1. **Understanding Containers:** What is a container, and how do they differ from traditional virtual machines?

2. **Container vs VM: Different Use Cases and Engines:** Explore the distinctive use cases for containers and virtual machines. Talk about the most popular Container Engine (Docker).

3. **Achieving Isolation in Containers:** Explain the mechanisms behind container isolation, including Linux namespaces, capabilities, and control groups (cgroups).

4. **Runtime and Network Security:** Talk about runtime protection and network security within container environments.

5. **Best Practices for Building Container Images:** Talk about the best practices for constructing secure container images, ensuring a solid foundation for deployment.

6. **Learning from Past Exploits:** Analyze real-world security exploits related to containers, providing insights into preventive measures and lessons learned.

7. **Programming Example:** Illustrate security principles through a practical programming example and showcase how a container is isolated from the rest of the System.

3 Description and plan for the code example

1. **Filesystem restrictions:** Examples of using binds/volumes to share data between container and host, using/escaping from chroot.

2. **Resource limitation:** Show that programs running in a container can't use up all host memory/cpus. The code will contain a C program with a memory leak and a python program with ReDOS.

3. **Syscall limitation:** A C program that runs syscalls that are blocked by default (e.g. reboot, ptrace), and an exploit for breaking out of a misconfigured container (unconfined seccomp).

4. **Container Image Integrity Check:** A C program that creates a hash of a container image and compares that hash to a well-known hash of the image to ensure the image hasn't been tampered with.