Practical Course: Container Security
SoSe 2025

Benedikt Hofmann

Dr. Patrick Stöckle

Organization

Lecturers

- Benedikt Hofmann
- Dr. Patrick Stöckle

Company/Department

Siemens AG
  Foundational Technologies
    Research and Pre-Development
      Cybersecurity & Trust
        Security Architecture

SIEMENS
Products & Services
Industries
Company
Company > Innovation > Siemens Company Core Technologies > Cybersecurity & Trust
Cybersecurity & Trust

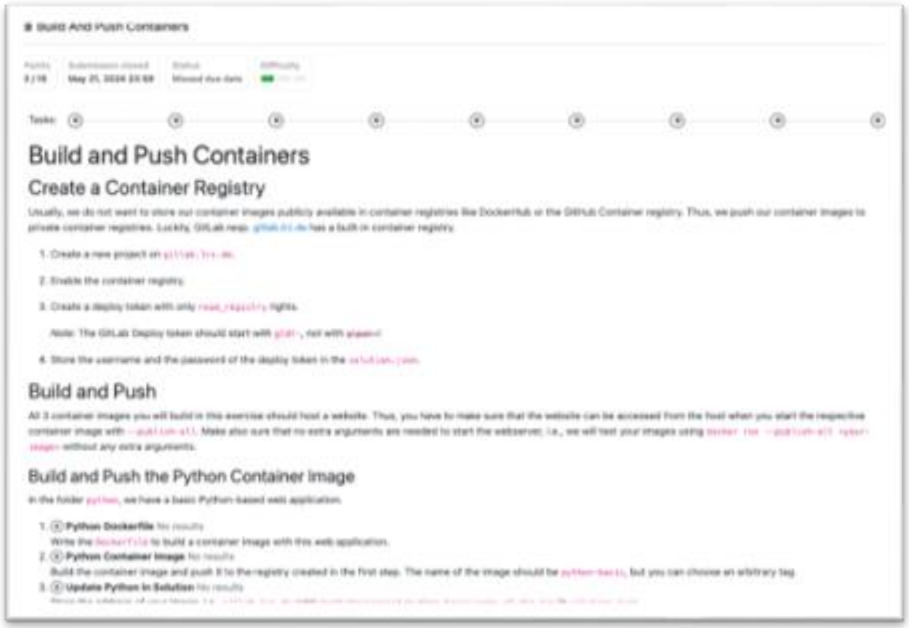
Organization: Theory Sessions

- Biweekly 3h hours
- On-site (no recordings, no stream)
- Probably Wednesday from 14:00 to 17:00
- 5-10min break after 50min

SIEMENS TUM

Organization: Exercises

- Graded Exercises
- Submission via Artemis
- Deadline: Tuesday, 23:59 (biweekly)
- Ca. 350 points
  - ⇒ 175 points are enough for passing
  - Grade based on performance
- You can use the Artemis Communication feature to pose questions, or answer the questions of your colleagues



Build And Push Containers

Points	Submission closed	Status	Difficulty
0 / 10	May 21, 2024 23:59	Missed due date	[illegible]

Tasks:

Build and Push Containers

Create a Container Registry

Usually, we do not want to store our container images publicly available in container registries like DockerHub or the GitHub Container registry. Thus, we push our container images to private container registries. Luckily, GitLab repo. gitlab.lrz.de has a built-in container registry.

1. Create a new project on gitlab.lrz.de.
2. Enable the container registry.
3. Create a deploy token with only read_registry rights.

   Note: The GitLab Deploy token should start with gldt-, not with glpat-.
4. Store the username and the password of the deploy token in the solution.json.

Build and Push

All 3 container images you will build in this exercise should host a website. Thus, you have to make sure that the website can be accessed from the host when you start the respective container image with --publish-all. Make also sure that no extra arguments are needed to start the webserver, i.e., we will test your images using docker run --publish-all <your-image> without any extra arguments.

Build and Push the Python Container Image

In the folder python, we have a basic Python-based web application.

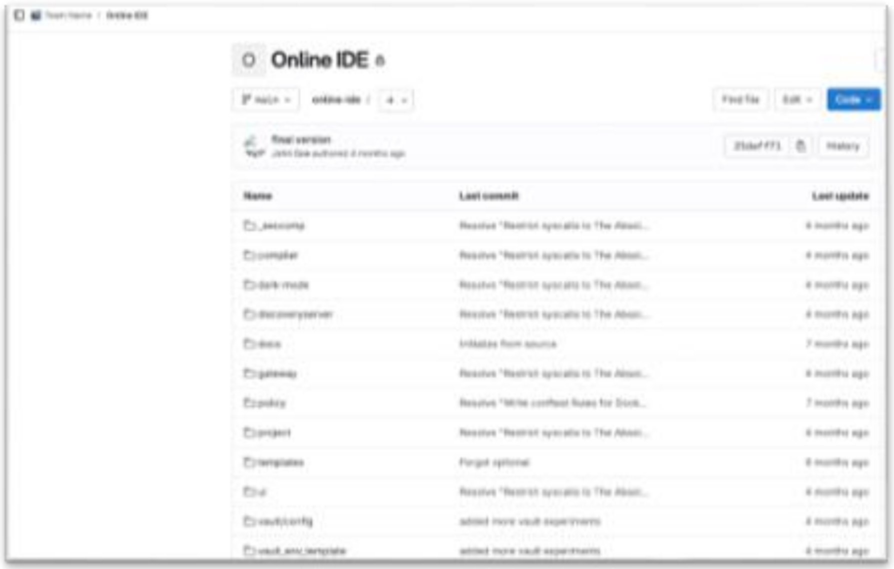
1. **Python Dockerfile** No results
   Write the Dockerfile to build a container image with this web application.
2. **Python Container Image** No results
   Build the container image and push it to the registry created in the first step. The name of the image should be python-basic, but you can choose an arbitrary tag.
3. **Update Python in Solution** No results
   [illegible]

Organization: Exercises

- We grade the submissions using automated tests
  - There will be visible tests, i.e., you see the result when submitting, and hidden tests, i.e., they are only executed after the deadline ⇒ Everything green when submitting does **NOT** mean 100% of the points
  - If the tests fail, but you did the *correct* thing ⇒ create a complaint, and we will check your solution
- Host OS:
  - We suggest to use **Debian** Linux to do the exercises. If you setup Debian in a VM,
  - In theory, most exercises will work on macOS or even with Docker on Windows. However, we will **NOT** offer any help to debug problems related to Docker on Windows/macOS. Furthermore, some automatic tests will

Organization: Team Project

- 3 students are a team
  - If you know already people in the class, you can form a team by yourselves
  - If you know nobody in the class, you can write us, and we will bring you together with other students to form a team
- Task: Apply container security concepts to a small, given project
- Team needs a GitLab group on gitlab.lrz.de
- You must apply/commit your changes to projects in the GitLab group
- You present your changes as team
  - Duration: ca 30min
  - Team presentation will happen in the semester break, i.e., end of August and September
  - Only the team attends the session, no other team ⇒ your team can schedule a date with us
  - Team presentation will happen online



Name	Last commit	Last update
_seccomp	Resolve "Restrict syscalls to The Absol...	4 months ago
compiler	Resolve "Restrict syscalls to The Absol...	4 months ago
dark-mode	Resolve "Restrict syscalls to The Absol...	4 months ago
discoveryserver	Resolve "Restrict syscalls to The Absol...	4 months ago
docs	Initialize from source	7 months ago
gateway	Resolve "Restrict syscalls to The Absol...	4 months ago
policy	Resolve "Write conftest Rules for Dock...	7 months ago
project	Resolve "Restrict syscalls to The Absol...	4 months ago
templates	Forgot optional	8 months ago
ui	Resolve "Restrict syscalls to The Absol...	4 months ago
vault/config	added more vault experiments	4 months ago
vault_env_template	added more vault experiments	4 months ago

Organization: Team Project

- Main goal: we want to see whether you **did** the exercises yourselves and **learned** something about container security.
  - If you did so, you should find things in the projects you could improve
  - If you received many points in the exercises but **cannot** explain in the group discussion, e.g., what a multi-stage build is, we might get suspicious…
- Your team will usually get one grade for the presentation. However, if we see a huge difference between the performance of the team members, we might grade them differently

Organization: Final Grade

$$\frac{grade_{exercises} + grade_{presentation}}{2} = grade_{final}$$

Can I pass the practicum only with the Group Discussion?

No, you **must** pass the practical exercises **and** the group discussion.

The 1:1 ratio of practical exercise and group discussion grade **ONLY** applies, if you pass both.

This means a 1.0 in the group discussion **CANNOT** fix a 5.0 in the practical exercises.

Organization: Application

You must participate in the usual matching on the matching platform (and rank this practicum with a high priority)

Furthermore, fill out this form ⇒ we will rank students who filled out the form with a high priority

Previous Knowledge Expected

- IN0006: Introduction to Software Engineering
  - Programming skills
  - Version control with git
- IN0009: Basic Principles: Operating Systems and System Software
  - General knowledge about the Linux OS
- IN0010: Grundlagen Rechnernetze und Verteilte Systeme
  - General knowledge about networking and firewalls
- IN0042: IT Sicherheit
  - General knowledge about security concepts like hash functions, encryption schemas, signatures

We do **NOT** assume that you are a Docker/container expert, but that you **SHOULD** have built/run your own containers already

Topics

Container Fundamentals

Attack surface reduction via multistage builds and minimized base images, e.g., Google distroless

Runtime restrictions, e.g., via seccomp profiles and Linux capabilities

Rootless container runtimes, e.g., podman

Anomaly detection for containers, e.g., via falco

Container image signing and secure container registries

Topics

K8s native network restrictions

Policy enforcement with Open Policy Agent

Role-based access control on orchestrated container runtimes

Secret and certificate management, e.g., Vault and cert-manager

Secure logging

Service meshes, e.g., Istio and Cilium

Vulnerability scanning, e.g., trivy

Feedback from the Practicum ConSec 2024

What did you like about the practicum?

- *How hands-on it was! You could clearly tell the instructors like what they do and that washe d out on me too! I also liked the fact there was a clear roter Faden to follow throughout the l ecture!*
- *You were motivated and as a student, I saw that you wanted to make this practical course as interesting as possible. There were hardly sections in your presentations that were borin g. I also liked that the exercises on artemis were structured really well and with the automat ed tests it was most of the time possible to have a good feeling when finishing an exercise and knowing that most of it should be correct.*

Questions?