Intro to DevSecOps

Why we need it?

- You got everything automated in your CI/CD pipelines; builds, tests, releases, and deployments.
- Time comes for a production deployment and the security team stops you for a security check that might take too long.
- DevSecOps tries to automate that and allow for shorter turnaround times security-wise.

What can go wrong?

- Vulnerabilities in your code
  - SQL Injection
  - Some null pointer causing remote access
- Vulnerabilities in dependencies/libraries you’re using in your project.
  - Log4j as an example
- Security issues with your containers or OSs in general.
  - Running as root
- Security vulnerabilities in your infrastructure.
  - Open Ports
  - Unencrypted connections (TLS)
- Leaked Secrets.
  - Usernames, Passwords, Tokens, …

How can we protect against those?

- Certain Scans during pre-commits.
- Certain Scans inside CI/CD Pipelines before/during builds.
- Certain Scans after deployment and during the actual run-time.

Low-hanging fruits

- Secrets in code
- Null pointers
- Root access for a process on a server or container
- Ports open to the public on a server (inbound & outbound)
- Encryption of data at-rest & rotation of keys
- Encryption of data in-transit using TLS (HTTPS, gRPCs, ...)
- Backups & Replications
- Don’t share host network/ports/pid/ipc with containers (unless you have to)

Links

- https://github.com/aquasecurity/trivy
- https://github.com/zricethezav/gitleaks
- https://github.com/securego/gosec
- https://github.com/anchore/grype
- https://github.com/prowler-cloud/prowler
- https://github.com/aquasecurity/tfsec
- https://github.com/snyk/cli
- https://github.com/devsecops/awesome-devsecops