

Uni App Security Notes

Felix Pojtinger

October 5, 2021

Contents

1	Introduction	2
1.1	Contributing	2
1.2	License	2
2	Organization	3
3	Overview	3
3.1	Elements of a Secure Development Process	3
3.2	Support Hierarchy	3
4	Basics	4
4.1	What is Secure Software?	4
4.2	What is Security?	4
4.3	CISSP Domains/Certificates	4
4.4	Why Security?	4

1 Introduction

1.1 Contributing

These study materials are heavily based on [professor Heuzeroth's "Anwendungssicherheit" lecture at HdM Stuttgart](#).

Found an error or have a suggestion? Please open an issue on GitHub (github.com/pojntfx/uni-appsecurity-notes):



Figure 1: QR code to source repository

If you like the study materials, a GitHub star is always appreciated :)

1.2 License



Figure 2: AGPL-3.0 license badge

Uni App Security Notes (c) 2021 Felix Pojtinger and contributors

SPDX-License-Identifier: AGPL-3.0

2 Organization

- 60 Minutes of test at the end
- Will have practical examples
- Threat detection plays a fundamental role in tests

3 Overview

3.1 Elements of a Secure Development Process

Primary purpose: Analysis of the data flow; data is both protected by the GDPR and represents value of the corporation

- **Requirements**
 - Security-Requirements
 - Anti-Requirements
 - Abuse cases
 - Protection poker
 - → **Security analysis/architecture analysis**
- **Draft**
 - AuthN/AuthZ
 - Drafting concepts
 - **Risk modelling**
- **Implementation**
 - Secure implementation guidelines
 - **Code review, dynamic analysis**
- **Tests**
 - Security testing plans
 - Security testing cases
 - **Ethical hacking, pentesting, dynamic analysis**
- **Operations/Maintenance**
 - Secure initial settings
 - Assumptions of runtimes
 - Observation of logs
 - **Processes for management and reaction to breaches**
- **Documentation**
 - Installation
 - Configuration
 - Customization
 - Operations
 - → **Impact area of security incidents must be visible***

3.2 Support Hierarchy

- **Level 1:** Direct support with customers; call center, non-technical
- **Level 2:** People who know about typical problems with the software

- **Level 3:** Developers of the software

4 Basics

4.1 What is Secure Software?

- Software which is protected against intentional attacks
- Every participant in the software development process should be interested in this objective
- Software must be hardened against all known attacks (and future, unknown attacks)

4.2 What is Security?

- $Risk = \frac{Cost\ of\ breach}{Probability\ of\ breach}$
- A system is protected against threats compromising valuable data using measures which lead to a reduced, accepted risk.
- Accepted risk is defined by context of use (i.e. nuclear power: very low accepted risks)
- **Safety:** Protection of the environment from the functional effects a system
- **Security:** Protection of the system from threats from the environment
- Concrete definitions: [uni-itsec-notes#security-objectives](#); most importantly (“CIA objectives”):
 - Confidentiality
 - Integrity
 - Availability
- If there are contractions between the security objectives (anonymity vs. accountability): The context defines which objectives dominate over others

4.3 CISSP Domains/Certificates

- **Security Engineering:** Engineering and Management of Security
- **Security Assessment and Testing:** Designing, Performing and Analyzing Security Testing
- **Security Operations:** Foundational Concepts, Investigations, Incident Management and Disaster Recovery
- **Software Development Security:** Understanding, Applying and Enforcing Software Security
- → This course strives for 80% of TPSSE compliance

4.4 Why Security?

- Security is context dependent: On `localhost` and unprotected UNIX socket isn’t an issue, but forward it with `socat` and it becomes a massive security vulnerability!
- With every change every test needs to be run again (regression testing)

- Typically ~30 errors in every 1000 lines of code
- Growing application complexity
- Devices are more and more connected which reduces the need for physical access
- Extensible architectures