Android ICC security smells
1st presentation

A bachelor thesis by Astrid Ytrehorn
Motivation

**Goal** → Spread awareness and fundamentally reduce the attack surface in Android

- Smartphones more widespread and vulnerable
- Android widely used - billions of users could be affected
- Security concerns are often overridden by other concerns
- Android ICC - Inter-Component Communication
Introduction

- 12 security “smells” (issues)
- Linting tool needs to be assessed
- Metadata explored and evaluated
- Placement of smells in apps investigated
- Based on a dataset of open-source apps (F-Droid)
Manual analysis

**Goal** → How reliable is the developed linting tool?

- 100 apps analysed manually
- Evaluate smells reported by lint
- Investigate occurrence of symptoms
- Important: Vulnerability potential of individual smells?
Contributor affiliation

**Goal** → Find correlation between # of contributors and # of smells

- Manual search for Github repositories of around 700 apps
- C# script using Octokit Github API
- Output: All contributors of individual Github repositories
App updates

**Goal** → Find whether updates affect projects positively, negatively or not at all

- Manual analysis of apps that received updates
- Look at lint differences between versions
- Check: have security smells changed?
Influence of project age and Activity

**Goal** → Show differences between more mature and younger apps, more and less frequent updated apps

- C# Octokit program, as for contributor affiliation
- Script extracts all commits + metadata
- It evaluates:
  - Commit ID
  - Timestamp
  - Author
  - Commit message
Unreported smells

**Goal** → Find out if the unreported smells are true negatives or not

- Search for unreported smells using regular expressions
- Example:
  - Mitigation of SM04 “Unauthorized Intent”
  - `grep --include \AndroidManifest.xml -E -rn 'android:exported|<service' *`
- Searches for keywords android:exported and <service in all AndroidManifest.xml files in all sub-directories.
- Manual evaluation of unclear cases
Placement of smells

**Goal** → How are the smells spread out in the different categories? What can we find from this?

- Analyse location of smells, collect the following metadata:
  - Enclosing method
  - Class
  - Package

- Extend linting tool to extract these parameters
Summary

- Further aspect of security smells need to be analyzed
- Part of effort to spread awareness of security issues in android apps
- Evaluate efficiency and correctness of linting tool
- “Human” factors (contributors etc.): How do they affect security?
- Assess if location of smells shows any patterns