IDE Plugins for Secure Android Applications Development: Analysis & Classification Study

Mohammed El Amin TEBIB (U. Grenoble)

Oum-El-Kheir Aktouf (U. Grenoble)

Pascal André (U. Nantes)

Mariem Graa (IMT Atlantique)

SECUREWARE

Context

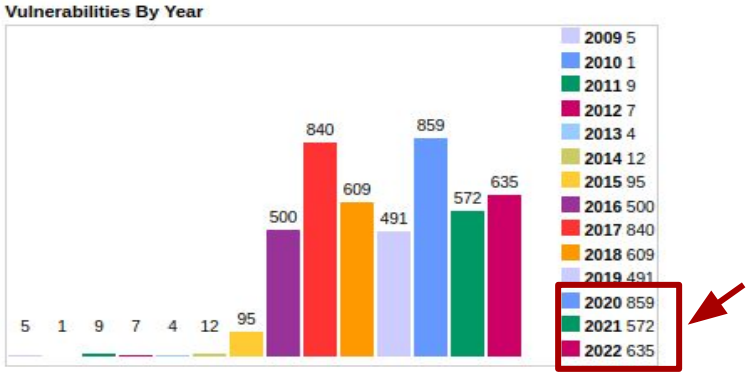
Android apps security

Key values findings

- Android is a leading mobile operating system

Android market share close to 73 % (05/2021) [Statista].

- Android applications mark a continuous increase of vulnerabilities (security threats)
  - 2066 vulnerabilities (2020, 2021, 2022) (CVE)



➔ Resulted Security Attacks: Malwares, Private Date Exfiltration.

https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/

https://www.cvedetails.com/product/19997/Google-Android.html?vendor_id=1224

Context

Ways to secure mobile ecosystem!

Protection solutions are proposed from different perspectives

- Educate users

Security-Policies

- Improve developer awareness about vulnerabilities (security threa

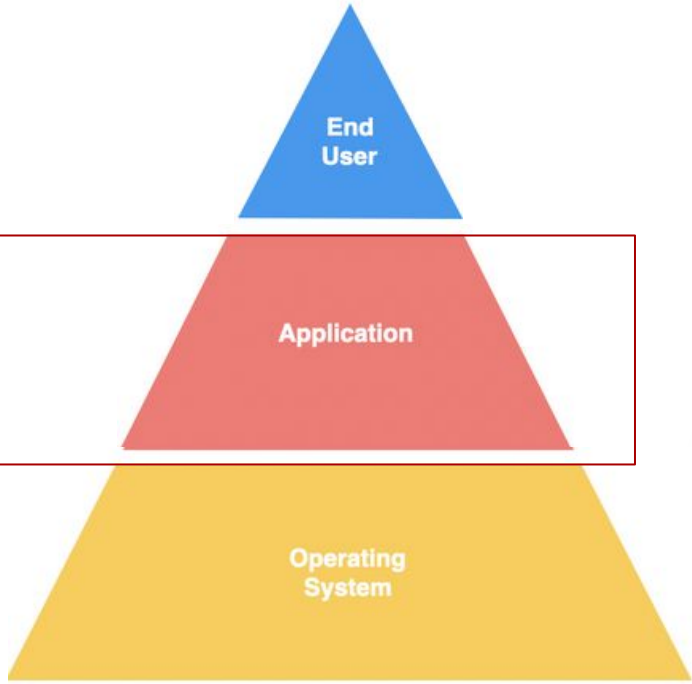
Our focus!

- Secure/Harden Android apps

Secure code

- Keep malicious apps out of the system

Anti-malwares

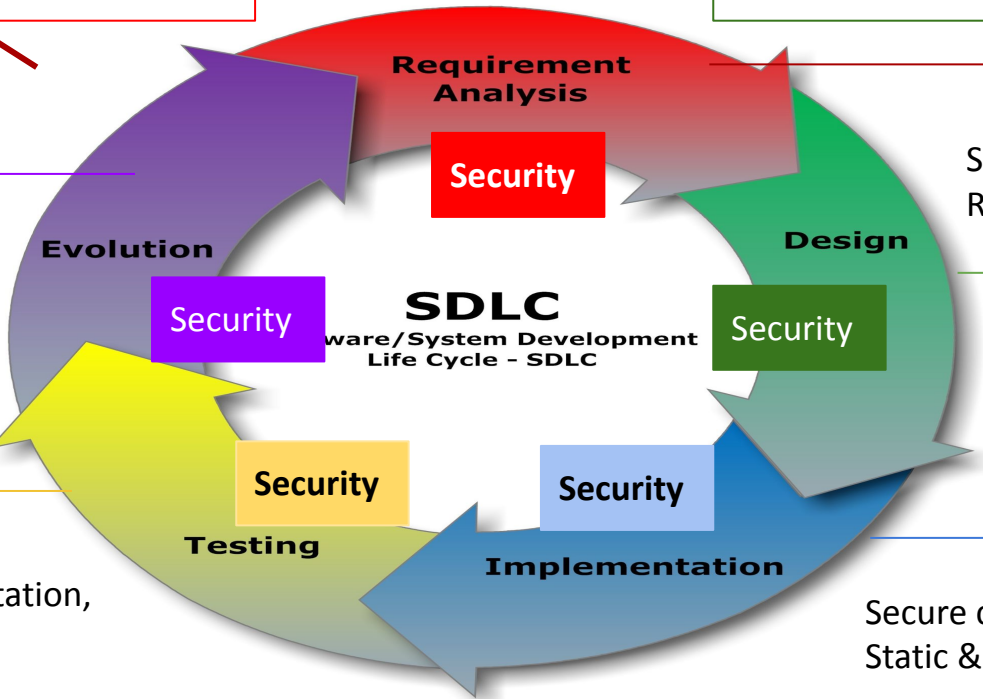


Motivation

Our Focus: Android Apps Vulnerability Detection From Developer's Perspectives

Android developers are the main reason of security vulnerabilities [R. Balebako et al. 2017; Scoccia; SCAM, 2019]

~~Security is an afterthought concern!~~ → New SD paradigms like: DevSecOps



Security Requirements
Risk assessment

Server Config Reviews
Network Config reviews

Arch & Design reviews
Threat Modeling

Tests: unit, instrumentation, fuzzing

Secure coding best practices
Static & Dynamic Analysis

Contribution

Common research works

Provide Android developers an overview of existing security analysis plugins

Mitra et al. [ESE, 2020]	Li. J et al. [PEASE, 2019] Baset, A et al. [SPW, 2017]	J. Mejia et al. [WCIST, 2020]
Vulnerabilities detection assessment Only pentesting tools Only academic tools	IDE plugins for secure development! Vulnerabilities detection assessment No focus on the Android apps	Systematic Review Too generic! No tools evaluation

Proposal:

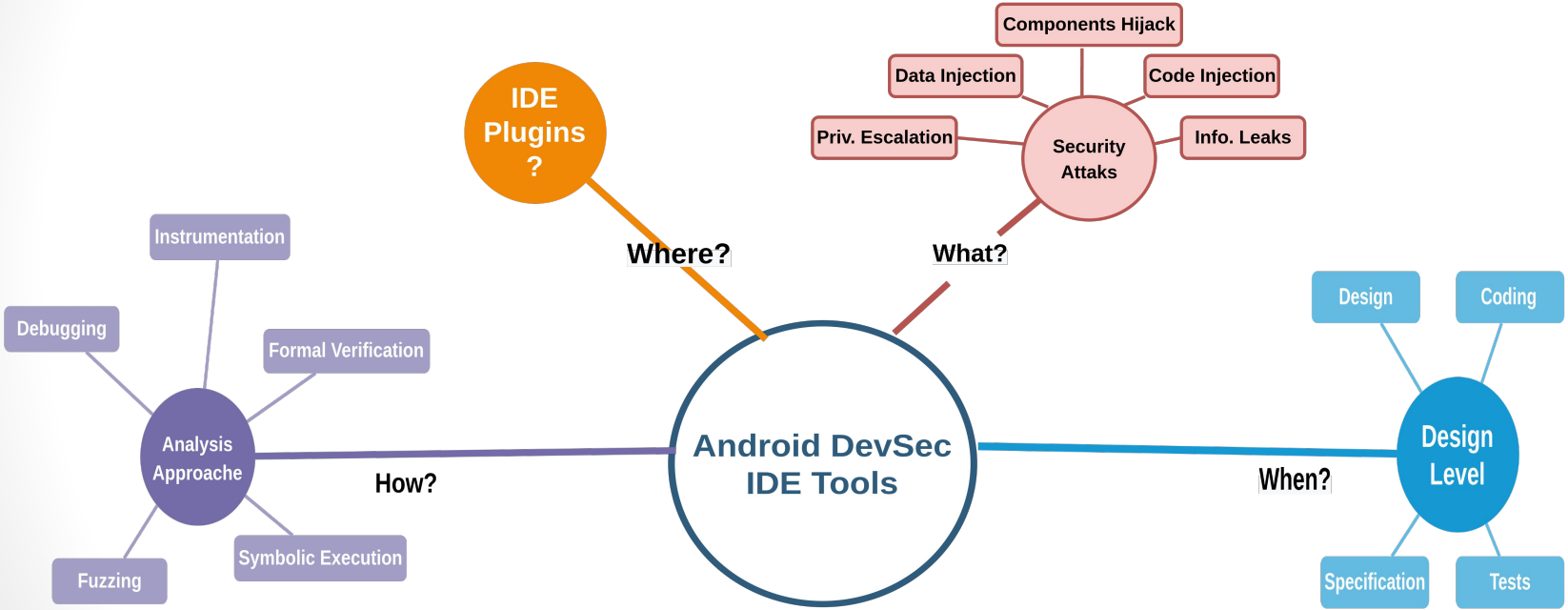
- Studying Android IDE plugins from security perspectives
- Studying different SD stages
- Experimental evaluation on real Android apps
- Academic and Industrial tools

Research Questions

- RQ1. What are the existing IDE plugins that assist developers in preventing the security issues in Android apps?
- RQ2. Is security considered in all the design levels during the development process of Android apps?
- RQ3. Which analysis techniques are being adopted by the existing security development solutions?
- RQ4. Are the studied IDE plugins effective in detecting known vulnerabilities ?

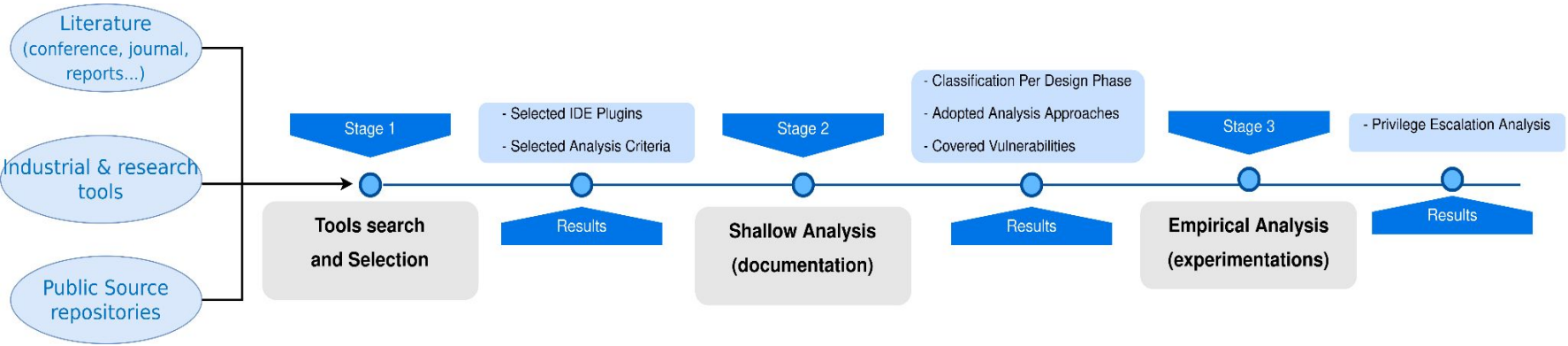
Proposal

Android DevSec classification framework



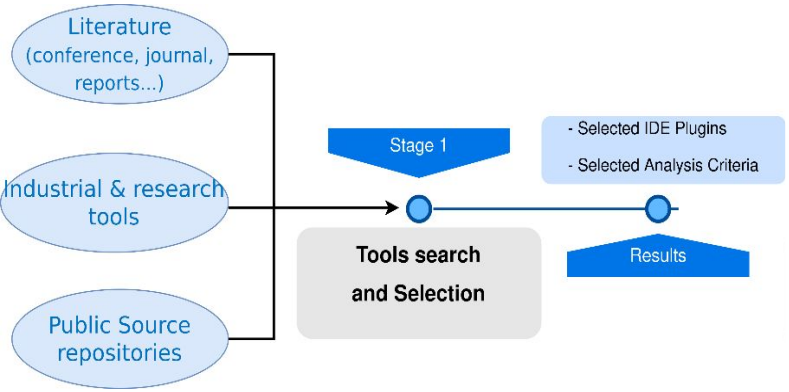
Methodology

Followed steps



Methodology

Step1: Search & Selection



Sources

Github Repositories [1]

Literature Reviews
Baset et .al (SPW, 2017)

Industrial tools Known by authors

Inclusion

Tools integrated with the IDE

Industrial free tools

[1] https://github.com/impillar/AndroidReferences

Results

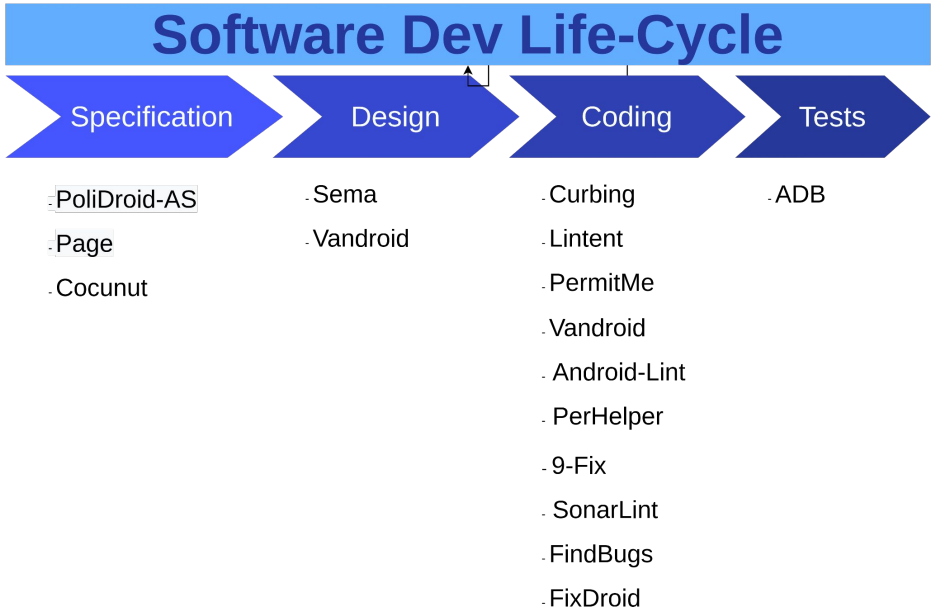
Answer RQ1: What are the existing IDE plugins that assist developers in preventing Android security issues ?

Tool Name	Ref.	Year	SD Stage	Focus	Approach	Method	Availability
Curbing	[7]	2011	CR	Permission Over-privilege	Static, Manual	AST	No
Lintent	[24]	2013	CR	Communication	static	FM	Ye
PermitMe	[25]	2014	CR	Permission Over-privilege	Static	AST	No
Page	[26]	2014	Spec	Privacy policies	Static	NL	No
Vandroid	[27]	2018	CR	Communication	Static	FM	No
Androidlint	[28]	2019	CR	Communication	Static	AST	Yes
Sema	[29]	2019	Design	General Security Properties	Static	FM	Yes
PerHelper	[30]	2019	CR	Permission Over-privilege	Static	AST	No
PoliDroid-As	[31]	2017	Spec	Privacy security policies	Static	NLP	No
9Fix	[32]	2021	CR	General Code smells	Static	AST	No
Sonarlint	[33]	2021	CR	General Code smells	Static	TA	Yes
FindBugs	[34]	2016	CR	General Code smells	Static	AST	Yes
Cocunut	[35]	2018	Spec	Privacy policies	Static	H	Yes
FixDroid	[36]	2017	CR	General Code smells	Static	AST	Yes

[1] AST: Abstract Syntax Tree; CR: Code Review; FM: Formal Methods; Spec: Specification;
[2] SD Stage: Software Development Stage; AV: Android Version

Results

Answer RQ2: Is security considered in all the design levels during the development process of Android apps?



Specification

Code source guided by textual specification of security requirement

Design

Analysis of Security properties at app models and graphical storyboards

Coding

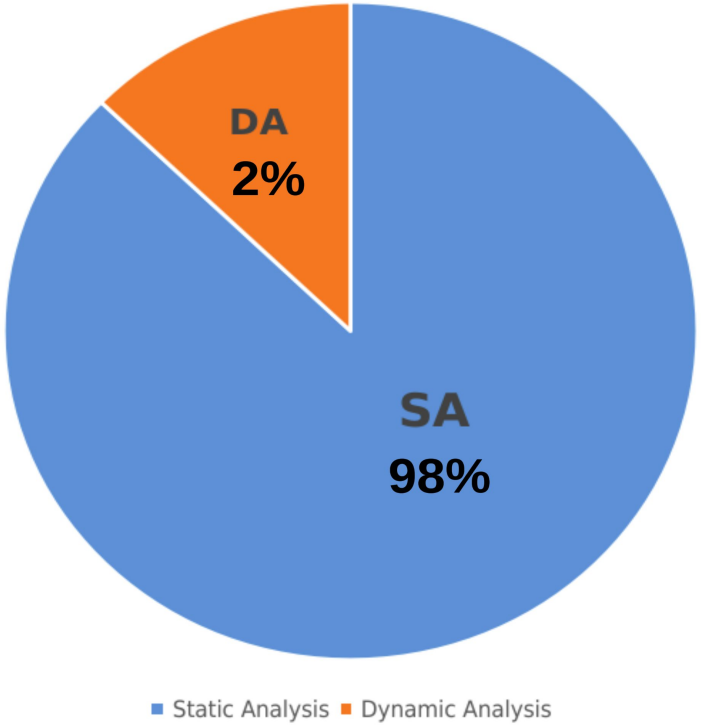
Code reviews; Security Smells

Testing

Pentesting tools

Results

Answer RQ3: Which analysis approaches or techniques are being adopted by the existing security development solutions?



Static Analysis

Abstract Syntax Tree (code smells)

Formal methods:
Reasoning about security aspects:
Inter Component Communication (ICC),
Permissions;
Verify security properties

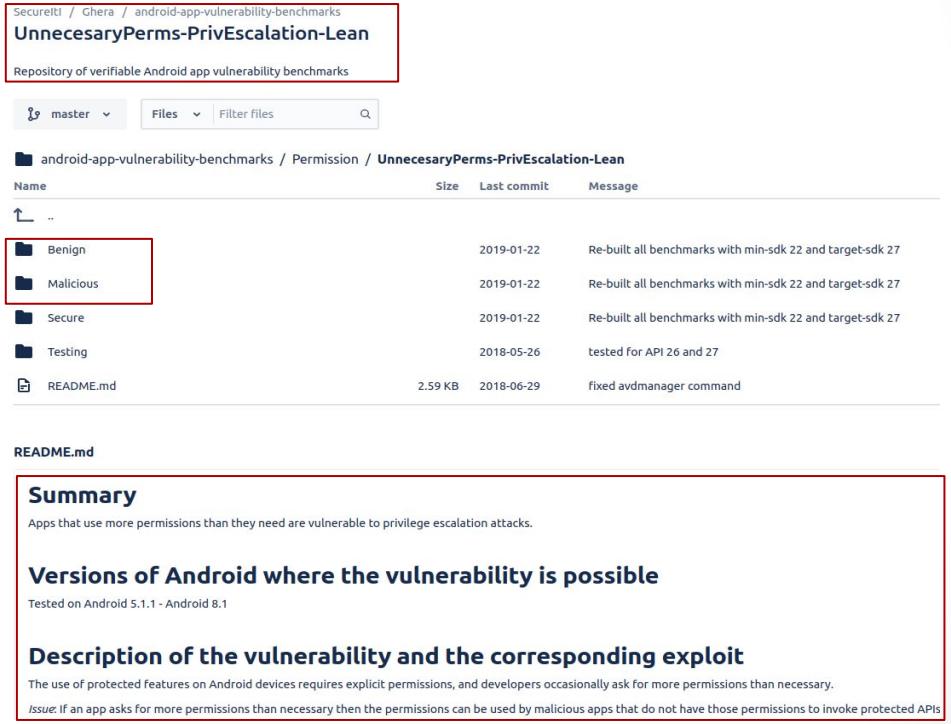
Dynamic Analysis

Run analysis scripts during app run

Vulnerabilities

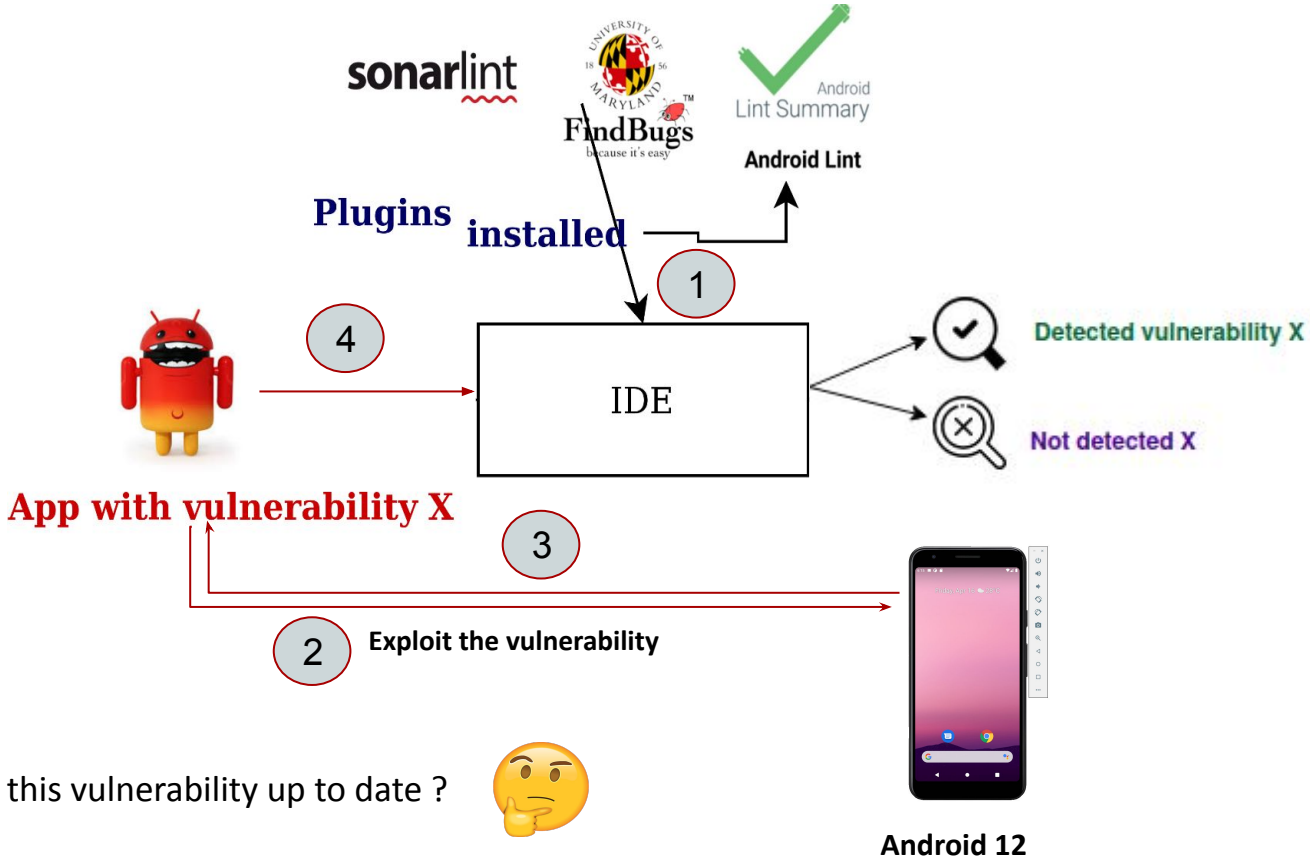
Ghera: A repository of Android Apps vulnerability benchmarks

- OpenSource apps
- +60 Well Known Vulnerabilities
- Vulnerability exploit
  - Benign app
  - Malicious app
- Gives Information about the vulnerability
- Reference to the source vulnerability



https://secure-it-i.bitbucket.io/ghera/index.html

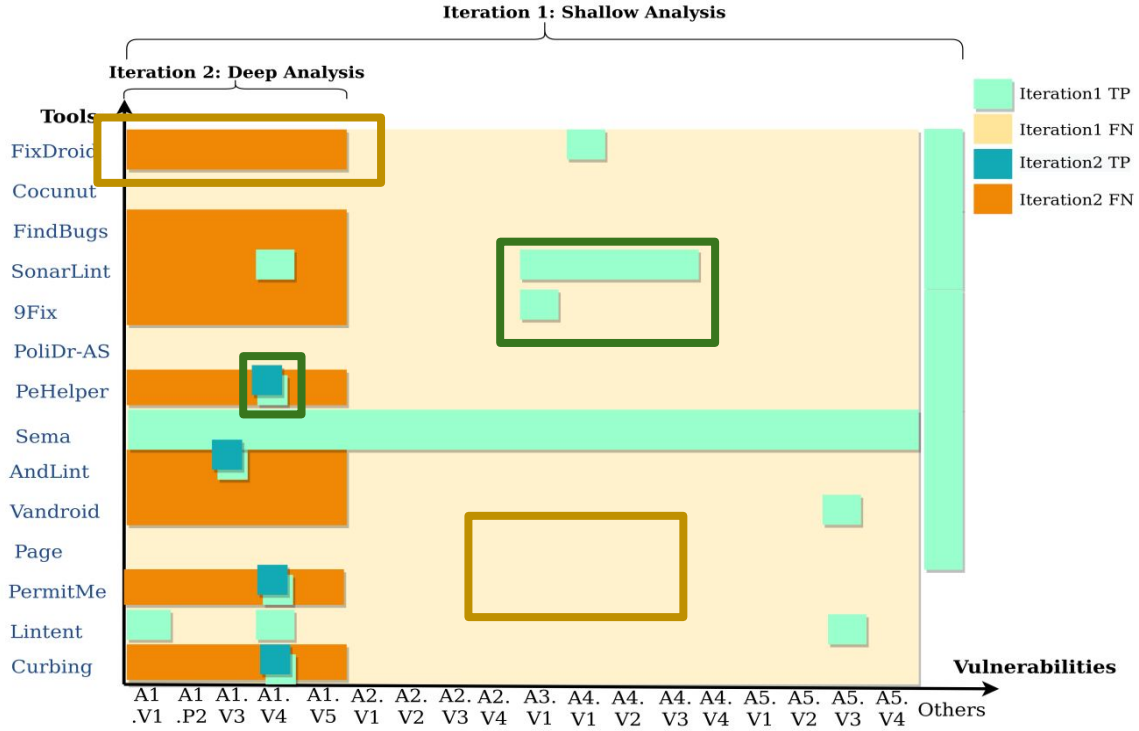
Assess Analysis Capabilities



Is this vulnerability up to date ?

Results

Answer RQ4: Are the studied IDE plugins effective in detecting known vulnerabilities?



TP True Positive: a vulnerability is present and detected by the tool.
FN False Negative: a vulnerability is present but not detected by the tool.

Observations

Tools outdatedness and availability

Most of the security assisting IDE plugins become outdated
Many tools are not available for testing and use

Tools Effectiveness

Much False Negatives (FN)

Analysis approaches for security

Lack of dynamic analysis approaches (important for: hijacking and over; privilege detection …)

Benchmark availability and incompleteness

Availability of more relevant benchmarks could be a real breakthrough towards more thorough security analysis.

Conclusion

Summary

- We provide Android developers an overview of existing security analysis plugins capabilities with regards to Android application development.

- We proposed a classification framework that deeply analyse a sample of IDE plugins based on three dimensions:

    1. The analysis based approach,
    2. Security vulnerabilities,
    3. Design level.

Future work

- Complete the deep analysis step (ongoing work)

- Extend the study with new more vulnerabilities

- Development of a tool with more analysis capabilities (ongoing work)

Thanks for your attention!

IDE Plugins for Secure Android Applications Development: Analysis & Classification Study

Mohammed El Amin TEBIB (U. Grenoble)

Oum-El-Kheir Aktouf (U. Grenoble)

Pascal André (U. Nantes)

Mariem Graa (IMT Atlantique)

SECUREWARE

Context

Vulnerabilities Included in our study

Privilege Escalation
- Unnecessary use of permissions
- Empty Pending Intent

Data Injection
- Ordered Broadcast
- Sticky Broadcast

Android Apps Vulnerabilities

Developer's Perspectives

Unauthorized Access
- Incorrect Handling of Implicit Intents
- Weak permission checks

DOS attacks
- Unhandled Exceptions

Information Leaks
- Internal To External Storage

Code Injection
- Dynamic Code Loading
- Java script execution

https://bitbucket.org/secure-it-i/android-app-vulnerability-benchmarks/

https://developer.android.com/