

**Daniel Szameitat**

# Schwachstellenanalyse & Reverse Engineering von Android Apps

**Studienarbeit**

# BEI GRIN MACHT SICH IHR WISSEN BEZAHLT



- Wir veröffentlichen Ihre Hausarbeit, Bachelor- und Masterarbeit
- Ihr eigenes eBook und Buch - weltweit in allen wichtigen Shops
- Verdienen Sie an jedem Verkauf

Jetzt bei [www.GRIN.com](http://www.GRIN.com) hochladen  
und kostenlos publizieren



## **Bibliografische Information der Deutschen Nationalbibliothek:**

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de/> abrufbar.

Dieses Werk sowie alle darin enthaltenen einzelnen Beiträge und Abbildungen sind urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsschutz zugelassen ist, bedarf der vorherigen Zustimmung des Verlanges. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen, Auswertungen durch Datenbanken und für die Einspeicherung und Verarbeitung in elektronische Systeme. Alle Rechte, auch die des auszugsweisen Nachdrucks, der fotomechanischen Wiedergabe (einschließlich Mikrokopie) sowie der Auswertung durch Datenbanken oder ähnliche Einrichtungen, vorbehalten.

## **Impressum:**

Copyright © 2014 GRIN Verlag, Open Publishing GmbH  
ISBN: 9783656749547

## **Dieses Buch bei GRIN:**

<https://www.grin.com/document/280939>

**Daniel Szameitat**

**Schwachstellenanalyse & Reverse Engineering von  
Android Apps**

## **GRIN - Your knowledge has value**

Der GRIN Verlag publiziert seit 1998 wissenschaftliche Arbeiten von Studenten, Hochschullehrern und anderen Akademikern als eBook und gedrucktes Buch. Die Verlagswebsite [www.grin.com](http://www.grin.com) ist die ideale Plattform zur Veröffentlichung von Hausarbeiten, Abschlussarbeiten, wissenschaftlichen Aufsätzen, Dissertationen und Fachbüchern.

### **Besuchen Sie uns im Internet:**

<http://www.grin.com/>

<http://www.facebook.com/grincom>

[http://www.twitter.com/grin\\_com](http://www.twitter.com/grin_com)

# **Schwachstellenanalyse & Reverse Engineering von Android Apps**

Autor: Daniel Szameitat

Datum: 15.09.2014

# Inhaltsverzeichnis

Einleitung.....	5
Voraussetzungen.....	5
Apps herunterladen.....	5
Testumgebung einrichten.....	5
Android Development Umgebung.....	6
Alternativer Emulator.....	9
Santoku.....	10
Schwachstellen in Android Apps.....	11
Intent Spoofing.....	13
Unauthorized Intent Receipt.....	14
Information Leakage.....	15
Backdoor.....	19
SSL.....	19
DOS.....	20
Weitere Schwachstellen.....	21
SQL, JavaScript & XML Injection.....	21
Fragment Injection.....	22
OWASP Mobile Top 10 Risks.....	23
Manuelle Analyse.....	25
Betrachtung der Manifest-Datei.....	25
App Konfiguration herausfinden.....	28
Ressourcen untersuchen.....	29
Statische Analyse.....	30
Grundlagen der Dekompilierung.....	30
Manuelle Dekompilierung von Dex-Bytecode.....	37
Dekompilierung von Dex zu Java.....	40
Prüfung von verdächtigen Konstanten.....	41
Dekompilierung von Dex zu Smali.....	43
Dekompilierung von Dex zu Jimple.....	48
Datenfluss Analyse mit FlowDroid.....	51
Optimierung von FlowDroid.....	52
Dynamische Analyse.....	55
Laufzeit injection mit Drozer.....	55
Schwachstellentests.....	57
Überwachung von Apps mit Logcat.....	58
Netzwerküberwachung.....	60
Anhang: VBS Script FindString.vbs & FindAndroidKonst.vbs.....	62
Anhang: „isSessionValid“ Methode.....	65

## Benutzerdefiniertes Verzeichnis

App Downloader Icon.....	5
Install.....	6
ADT installieren 2.....	7
ADT installieren 3.....	7
Emulator per Konsole starten.....	8
Android Debug Monitor.....	9
Installieren von Apps im Emulator.....	9
Verfügbare Geräte.....	10
Santoku.....	11
GoatDroid Konfiguration.....	12
GoatDroid App auf virtuellem Device.....	12
Mitschneiden von Nachrichten.....	15
SQLite Datenbank in GoatDroid.....	17
Öffnen von Userinfo.db.....	17
Datenbankfelder.....	17
Http Session mit geklauter Session ID.....	18
URLs in GoatDroid.....	19
DOS Angriff auf GoatDroid.....	20
Fenster zur Eingabe eines neuen PINs.....	23
Apktool Reversing.....	29
Der Kompilervorgang in Java.....	31
JVM.....	32
Beispiel einer Class Datei.....	33
Class Datei im Hex-Editor.....	35
Umwandlung von class zu dex.....	37
Vergleich von „*.jar“ und „*.dex“ Formaten.....	37
Hex Code von der Beispiel App.....	39
Apk zu Java.....	40
Unterschiede Dekompiler.....	41
Smali Ordnerstruktur.....	44
Syntax Highlight für Smali.....	47
Soot Projekt.....	48
Runnable JAR File.....	49
Dekompilierung mit Soot.....	49
GUI für FlowDroid.....	55
Drozer Server starten.....	56
Virtuelles Gerät mit Android API 4.0.3.....	58
Screen Lock.....	58
Schwachstellentest.....	58
Logcat GUI.....	60
Netzwerküberwachung mit Whireshark.....	60
Http Überwachung mit Fiddler2.....	61
Fiddler2 Options.....	61