DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Analysis of Android Cracking Tools and Investigations in Counter Measurements for Developers

Johannes Neutze

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Analysis of Android Cracking Tools and Investigations in Counter Measurements for Developers

Analyse von Android Crackingtools und Untersuchung geeigneter Gegenmaßnahmen für Entwickler

Author:	Johannes Neutze
Supervisor:	TODO: Supervisor
Advisor:	TODO: Advisor
Submission Date:	TODO: Submission date

I confirm that this master's thesis in informatics is my own work and I have documented all sources and material used.

Munich, TODO: Submission date Johannes Neutze

Acknowledgments

Abstract

Contents

1 Introduction

sis is a text

1.1 Licensing

Was ist licensing und warum? allgemein

1.2 Motivation

enthält als Abschluss SCOPE

1.3 Related Work

related work

2 Foundation

sis is a text

2.1 Android

sis is text

2.1.1 History

sis is text

2.1.2 Basics of Android

sis is text

2.1.3 Evolution of the Android Compiler

sis is text

Java Virtual Machine

sis is text

Dalvik Virtual Machine

sis is text

Android Runtime

im Moment abwärtskompatibilität dex in oat (tools zum extrahieren nennen)

2.1.4 Root on Android

what is it? how is it achieved? what can i do with it? (good/bad sides)

2.2 License Verification Libraries

What is a lvl? why are they used? connection to store

2.2.1 Amazon

Amazon DRM

Implementation

sis is text

Functional Principle

sis is text

Example

anhand eigener app

2.2.2 Google

License Verification Library

Implementation

sis is text

Functional Principle

sis is text

Example

anhand eigener app

2.2.3 Samsung

Zirconium

Implementation

sis is text

Functional Principle

sis is text

Example

anhand eigener app

2.3 Reengineering Tools

main tools

2.3.1 Dex

mein custom script erklären

2.3.2 baksmali

https://github.com/JesusFreke/smali

2.3.3 Java

Androguard

https://github.com/androguard/androguard

jadx

https://github.com/skylot/jadx

2.3.4 Diff

https://wiki.ubuntuusers.de/diff
-N: Treat absent files as empty; Allows the patch create and remove files.
-a: Treat all files as text; Allows the patch update non-text (aka: binary) files.
-u: Set the default 3 lines of unified context; This generates useful time stamps and context.

-r: Recursively compare any subdirectories found; Allows the patch to update subdirectories.
script erklären

3 Cracking Android Applications with LuckyPatcher

http://lucky-patcher.netbew.com/

3.1 What is LuckyPatcher and what is it used for?

wer hat ihn geschrieben?
auf welcher version basiere ich
su nicht vergessen
was kann er alles
was schauen wir uns an?

3.2 Operation

wo arbeitet er?
warum dex und nicht odex anschauen?
patterns und patching modes grob erklären (modi von luckypatcher die verschiedene operationen (pattern) auf app anwenden) => vorgehensweise zur

3.3 What patterns are there and what do they do?

was greift jedes pattern an? wie wird der mechanismus ausgeklingt? was ist das result?

3.4 What are Patching Modes are there and what do they do?

kombination von patterns.
welche modes gibt es? welche patterns benutzen sie?
welche apps getestet und welche results?

3.5 Learnings from LuckyPatcher

was fällt damit weg?
erklären warum (2) 5.1.2 Opaque predicates zb nicht geht, da auf dex ebene einfach austauschbar
simple obfuscation for strings? x -> string (damit name egal)

4 Counter Measurements for Developers

am besten mit example

4.1 Basic Approaches

siehe masterarbeit 2

4.1.1 Simple Approaches

Root Detection

http://stackoverflow.com/questions/10585961/way-to-protect-from-lucky-patcher-play-licensing

LuckyPatcher Detection

http://stackoverflow.com/questions/13445598/lucky-patcher-how-can-i-protect-from-it

Sideload Detection

http://stackoverflow.com/questions/10809438/how-to-know-an-application-is-installed-from-google-play-or-side-load

4.1.2 Obfuscation

master1

4.1.3 Modify the Library

google

4.1.4 Tampar resistent

google

4.1.5 Junkbyte Injection

master1

4.1.6 Checken ob ganzer code abläuft und dann nacheinander elemente aktivieren

master1 - testen

4.1.7 Hidden Classes

master1

4.2 Additional Software

sis is text

4.2.1 Dexguard

master2

4.2.2 Dexprotector

master2

4.3 Additional Hardware and Verification

sis is text

4.3.1 Remote Verification

4.3.2 Secure Elements

5 Conclusion

sis is a text

5.1 Android

sis is text

5.1.1 History

sis is text

5.1.2 Basics of Android

sis is text

5.1.3 Evolution of the Android Compiler

sis is text

Java Virtual Machine

sis is text

Dalvik Virtual Machine

sis is text

Android Runtime

im Moment abwärtskompatibilität dex in oat (tools zum extrahieren nennen)

5.1.4 Root on Android

what is it? how is it achieved? what can i do with it? (good/bad sides)

5.2 License Verification Libraries

What is a lvl? why are they used? connection to store

5.2.1 Amazon

Amazon DRM

Implementation

sis is text

Functional Principle

sis is text

Example

anhand eigener app

5.2.2 Google

License Verification Library

Implementation

sis is text

Functional Principle

sis is text

Example

anhand eigener app

5.2.3 Samsung

Zirconium

Implementation

sis is text

Functional Principle

sis is text

Example

anhand eigener app

5.3 Reengineering Tools

main tools

5.3.1 Dex

mein custom script erklären

5.3.2 baksmali

https://github.com/JesusFreke/smali

5.3.3 Java

Androguard

https://github.com/androguard/androguard

jadx

https://github.com/skylot/jadx

5.3.4 Diff

https://wiki.ubuntuusers.de/diff
-N: Treat absent files as empty; Allows the patch create and remove files.
-a: Treat all files as text; Allows the patch update non-text (aka: binary) files.
-u: Set the default 3 lines of unified context; This generates useful time stamps and context.

-r: Recursively compare any subdirectories found; Allows the patch to update subdirectories.
script erklären

Glossary

computer is a machine that. . . .

Acronyms

TUM Technische Universität München.

List of Figures

List of Tables