AppSpear: Bytecode Decrypting and DEX Reassembling for Packed Android Malware

Yang Wenbo, Zhang Yuanyuan, Li Juanru, Shu Junliang, Li Bodong, Hu Wenjun, Gu Dawu

Sudeep Nanjappa Jayakumar

Agenda

- Introduction
- AppSpear – Goals, Contributions
- Code Packed Android Malware
- Analysis
- AppSpear – Overview
- DEX Reassembling
- Experimental Evaluation
- Accuracy of DEX reassembling
- Related Work
- Conclusion

Introduction

- A range of static and dynamic analysis approaches(using code similarity comparison to detect malware) have been proposed for detecting malicious Android apps.
- These techniques are initially designed to counter reverse engineering and effectively resist many program tampering attempts, they are becoming a common measure of malware detection circumvention.
- Current unpacking approaches are either based on manual efforts, which are slow and tedious, or based on coarse-grained memory dumping.
- Investigation on 37,688 Android malware samples is conducted to take statistics of the android apps.
- This paper conducts a systematic study of packed Android malware, and examines the feasibility of universal and automated un-packing for Android applications.

AppSpear

- AppSpear is a generic and fine grained system for automatic malware unpacking.
- Its core technique is a bytecode decrypting and Dalvik executable (DEX) reassembling method, which is able to recover any protected bytecode effectively without the knowledge of the packer.
- AppSpear directly instruments the Dalvik VM to collect the decrypted bytecode information from the Dalvik Data Struct (DDS), and performs the unpacking by conducting a refined reassembling process to create a new DEX file.
- The unpacked application is then available for analyzing by malware detection systems.
- AppSpear is the first automatic and generic unpacking system for current commercial Android packers.

Goal

1. Thorough investigation is done on large-scale Android malware samples to take statistics of how widespread those malware samples are protected by Android packers.

    a) 10 popular commercial android packers are studied which are used by malware authors, also which covers the majority of existing techniques and then the investigation is conducted on 37,688 malware samples.

    b) This contains 490 code packed malware.

2. To address the challenge of analyzing code packed malware, Authors have proposed AppSpear which is a generic and fine grained system for automatic malware unpacking.

Contributions

- Thorough investigation on both existing mainstream Android packers and code packed Android malware is done in the wild and further summarizing typical anti-analysis defenses of Android packers.
- A bytecode decrypting and DEX reassembling technique is proposed to rebuild protected apps. Our APK rebuilding process transforms a code packed malware to an unpacked one, which is a feasible form for commodity program analysis.
- Design of automated and generic unpacking system, AppSpear is done to deal with most mainstream Android packers and the unpacked apps can be validated by state-of-the-art analysis tools, which are not able to handle the packed form beforehand.

Code Packed Android Malware

- Investigation of 10 popular commercial Android packers (Bangcle, Ijiami, Qihoo360, etc) and build a signature database.
- Collected 37,668 malware samples from 2012 to May 2015 using SandDroid, which detects malware according to the feedback results of 12 main virus scan engines from VirusTotal (F-Secure, Symantec, AntiVir, ESET-NOD32, Kaspersky, BitDefender, McAfee, Fortinet, Ad-Aware, AVG, Baidu-International, Qihoo-360).
- An app is regarded as malware if more than three virus scan engines detect it.

Summary of Packed Android Malware

Table 1. Summary of Packed Android Malware

(a) Annual statistics

Year	Malware collected	Packed	Ratio
2012	16157	6	0.04%
2013	15443	89	0.58%
2014	5819	376	6.46%
2015	249	19	7.63%

(b) Distribution of packers

Packer	Number of Samples
APKProtect	37
Bangcle	402
NetQin	10
Naga	1
Qihoo360	23
Ijiami	27

Analysis

- Analysis indicates that anti-analysis defenses employed by those packers can be classified into three categories.

  a) The first category of anti-analysis defenses involve functions that check the static and dynamic integrity of the app.

  b) The second category of anti-analysis measures involve source code level obfuscation, which requires the source code to employ the protection.

  c) The third category, which is most complex, involves bytecode hiding.

- The integrity is also checked for the packed apps to decide if the apps are tampered and it is checked with both static and dynamic process.

AppSpear - Overview

AppSpear employs the unpacking through three main steps:

- AppSpear introspects the Dalvik VM to transparently monitor the execution of any packed app.
- AppSpear collects DDS in memory and performs a reassembling process on the collected DDS with some modified methods fixed to re-generate a DEX file.
- Finally, AppSpear resects anti-analysis code and further synthesizes the DEX file with the manifest le and other resource les from the original packed APK as an unpacked APK.

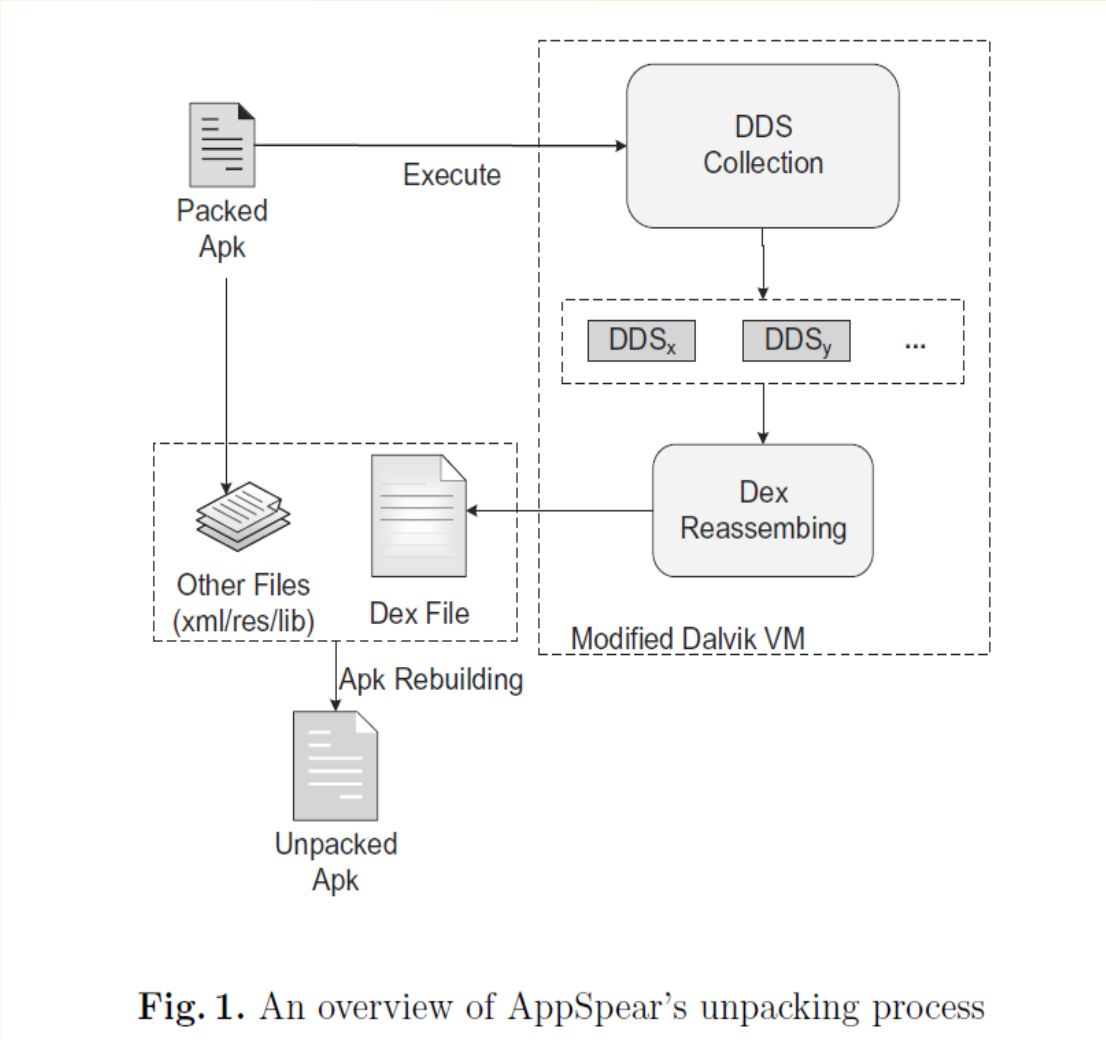

Fig. 1. An overview of AppSpear's unpacking process

DEX Reassembling

- DEX reassembling of AppSpear is a reverse process of the DEX loading procedure.
- AppSpear leverages this to employ the DEX reassembling process.
- Dalvik VM maintains 18 DDS parsed from a DEX le during runtime.
- DDS can classified into two types:

  a) Index DDS (IDDS) including Header, StringId, TypeId, ProtoId, FieldId, MethodId, ClassDef and MapList. The main functionality of IDDS is to index the real offset of the second type of DDS.

  b) Content DDS (CDDS) including TypeList, ClassData, Code, StringData, DebugInfo, EncodedArray and four items related to Annotation. This type of DDS mainly store raw data of byte-code content information.

DEX Reassembling Contd..

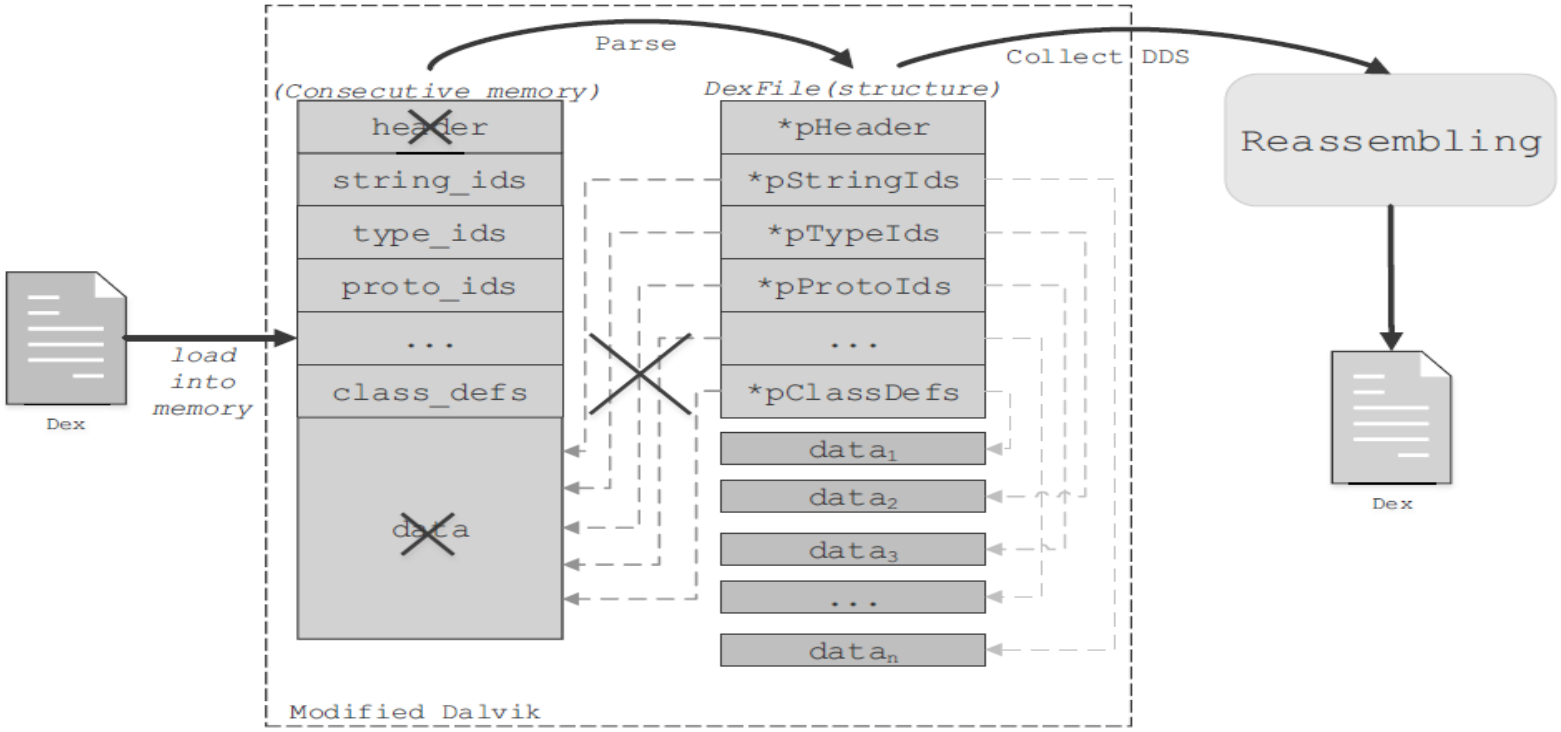

Fig. 2. DEX reassembling

APK Rebuilding

- AppSpear performs a last step APK rebuilding to obstruct analysis.

a) Anti-analysis Code Resecting:

AppSpear maintains an empirical database of code stubs and automatically resects any code stubs in database when encountering.

b) APK Repackaging:

AppSpear combines the reassembled DEX le with materials from the existing packed app including manifests.xml and resource files to repackage the app. The manifests le of an app declares the permissions and the entry points of the app and these are directly used in the repackaged app.

Experimental Evaluation

- 31 packed malware were manually chosen from the collected 490 packed samples of SandDroid to test AppSpear.
- These 31 samples could run without crashes or exceptions before unpacking and are all of different package names.
- **Authors developed a home brewed malicious app that requires many permissions and collects sensitive data.**
- The test app contains all four main components (Activity, Service, Broadcast Receiver, Content Provider) and an Application class.
- Test app is submitted to 7 online packing services of Bangcle (a.k.a Secneo), Ijiami, Qihoo360, Baidu, Alibaba, LIAPP and Dex-Protector.

Experimental Evaluation

- AppSpear is deployed on two devices, Galaxy Nexus and Nexus 4.
- Versions of Android operating system are 4.3 and 4.4.2.
- Modified Dalvik VM is built based on the AOSP source code and replace the default Dalvik VM with the AppSpear integrated one.
- AppSpear conducts the unpacking as soon as the Main Activity class invokes the onCreate method.
- All of the samples are unpacked automatically and the corresponding unpacked APK les are generated.

Accuracy of DEX reassembling

- The accuracy of the DEX reassembling is done using the 5 popular and widely used static tools and they are EXTemplate for 010Editor, Baksmali, Enjarify, IDA Pro and AndroGuard.
- The testing set consists of 7 home brewed samples submitted to online packers and 31 malware samples from the collected 490 packed samples, which covers 10 different packers altogether.

DEXTemplate	Baksmali	Enjarify	IDA Pro	AndroGuard
38/38	37/38	34/38	38/38	38/38

Table 2. success rate of parsing reassembled DEX

- The result above shows that DEXTemplate for 010Editor, IDA Pro and AndroGuard successfully parse all reassembled DEX files.

Related work

- Polyunpack:

  Performs automatic unpacking by emulating the execution of the program and monitoring all memory writes and instruction fetches, and considers all instructions fetched from previously written memory locations to be successfully unpacked.

- Omniunpack:

  Is a real-time unpacker that performs unpacking by looking for written-then-execute pattern.

- Renovo:

  Uses the written-then-execute pattern to perform the unpacking. It instruments the execution of the binary in an emulator and traces the execution at instruction-level.

- Eureka:

  Uses coarse-grained NTDLL system call monitoring for automated malware unpacking, is only available for Windows packers.

Conclusion

- This paper is mainly about the systematic study of code packed Android malware.
- An investigation of 37,688 Android malware samples is conducted and 490 code packed apps are analyzed with the help of AppSpear.
- AppSpear employs a novel bytecode decrypting and DEX reassembling approach to replace traditional manual analysis and memory dump based unpacking.
- Experiments have demonstrated that AppSpear system is able to unpack most malware samples protected by popular commercial Android packers.
- AppSpear is most essential process of current Android malware detection.

Thank You