



Advanced exploitation method of Android Master Key vulnerability (bug 8219321)

Алюшин Виктор, НИЯУ МИФИ

Структура APK-файла

APK = JAR = ZIP архив

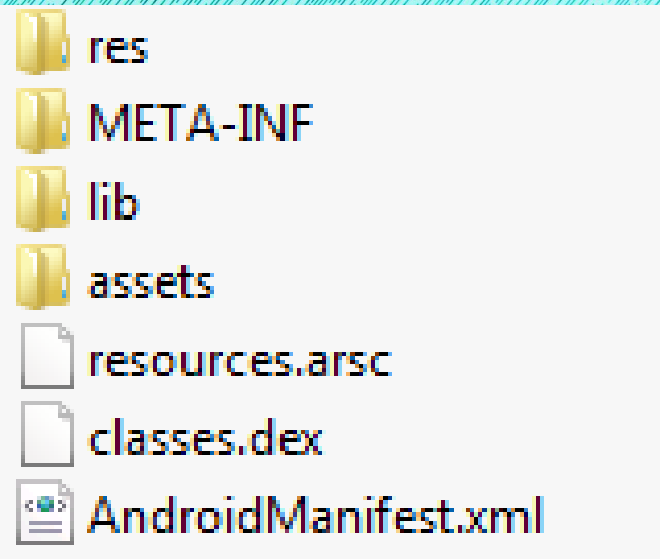
- **AndroidManifest.xml** – информация о приложении
- **classes.dex** - скомпилированный Java-код, выполняемый в Dalvik VM (Android SDK)
- **resources.arsc** - скомпилированный XML-файл, содержит данные о ресурсах
- **res** – директория со структурированными ресурсами
- **assets** – директория с любыми файлами ресурсов
- **lib** – директория с библиотеками, написанными на C++ (Android NDK)
- **META-INF** – директория с контрольными суммами и цифровой подписью приложения

META-INF

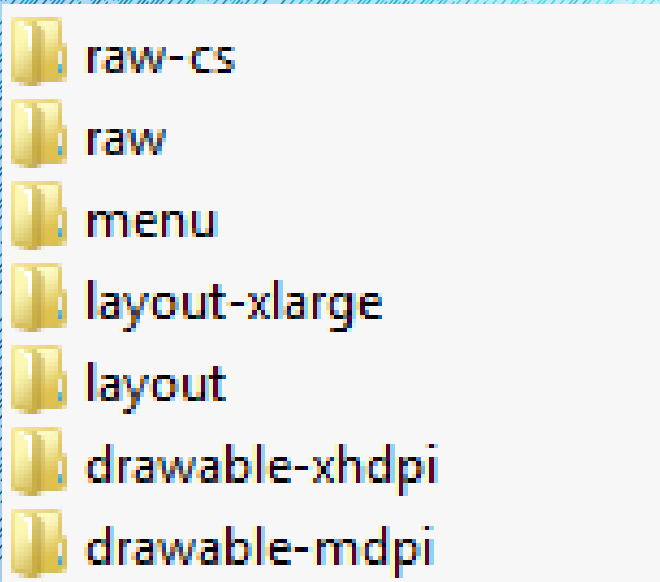
lib

MANIFEST.MF
CERT.SF
CERT.RSA

x86
armeabi-v7a
armeabi



res



META-INF

MANIFEST.MF

CERT.SF

Manifest-Version: 1.0

Created-By: 1.0 (Android)

Name: classes.dex

SHA1-Digest: 8ZVQygOX3TmjiHIffHMaWooj5Rw=

Name: AndroidManifest.xml

SHA1-Digest: JrJaswuwdusQptU3qGu1nU+QmhA=

Name: resources.arsc

SHA1-Digest: rfh8s9GkX94owbIOiH87EX+jLTY=

Manifest-Version: 1.0

Created-By: 1.0 (Android)

SHA1-Digest-Manifest:

O+PILC6t6JnlsyVrVcUqMZKS9Tk=

Name: classes.dex

SHA1-Digest: 4T5irdJ5LZt7XM3OmNVgBUKh6l8=

Name: AndroidManifest.xml

SHA1-Digest: C8wrzCe/1uKzewtEIW4AQydC/zw=

Name: resources.arsc

SHA1-Digest: EFn7Dk1bMwxXzjTZYPwoPVhOuzo=

```
$ openssl sha1 -binary classes.dex | openssl base64
8ZVQygOX3TmjiHIffHMaWooj5Rw=
```

```
$ openssl sha1 -binary MANIFEST.MF | openssl base64
O+PILC6t6JnlsyVrVcUqMZKS9Tk=
```

```
$ echo -en "Name: classes.dex \r\nSHA1-Digest: \r\n8ZVQygOX3TmjiHIffHMaWooj5Rw= \r\n\r\n" | openssl
sha1 -binary | openssl base64
4T5irdJ5LZt7XM3OmNVgBUKh6l8=
```


CERT.RSA

```
p?J ?-      *?H??
• 1 ??J ?0?J ?1 1
0 - | +# L 1 → | 0
-      *?H??
• ??L 0?L
0? x?L 1 1 1 J \лс?0
-      *?H??

| 071
0 - L UJ - !! 1 US1† 0# - L UJ
!! • Android1т 0# - L UJ L !!
Android Debug0 |
130618204118Z|
430611204118Z071
0 - L UJ - !! 1 US1† 0# - L UJ
!! • Android1т 0# - L UJ L !!
Android Debug0? "0
-      *?H??
| L ? № 0?
1 ? ?t?Ж{y 1 /X6† p???!! ?ыak??L?JЖε=?S??юта}HHObIk?KL%цл??*?ЕбФ
??4$?хй??| {e ?- }XbN^+ ?# б8SJbl?† Dл$Nтя ?P?gha→ ,0 x@т?KN9r*  ??'??ц??-y7XC??м6Wx??2I3? p?if?sXy5][yМи  <j.c)Я+Етев???)ыsx† r%?ж+M?6?Е1 ]?a:ш1?Ю.МвгЯГ>Vnt† ?ox† ь?
Г??//4:~лIP[pXεP6Ж?NG† Se?wPE9† 1И?6???)?L?Ш??1 L ?!0 0 - L U  #† т†  ПЕ?I8<?х-Вй?т?рc† x =0
-      *?H??

| L ? [?]б7{ьбцш??F?
?-?mA74ШZB??1?Ицф??ц?† h?B?| YS?C
7AJ;??яРй† лз?H?dn2† Еп?=dл?дф1?→ Q*цbs'в&XO ?2??† → ?W
+ ?I??!??A?Br??# ?8Cш!|| мм$?† W?yYA?щ?c?/({/Bpл  L ?mUZuv# ?щ† ?н| PPratG;† f_# ?Фл?{{д?арOX† | еь?oD7H?† ?Й† Q??m5jyГ8ъ?c?ат&и?
??qS3Б??нT,?ьz-?д)P7| ЖЮо?aICЮ# ??ьеБс 1? f0? b1 0?071
0 - L UJ - !! 1 US1† 0# - L UJ
!! • Android1т 0# - L UJ L !!
Android Debug† \лс?0 - | +# L 1 → | 0
-      *?H??
| J ? v?~??xSC† ?йж- T04G3?LS?vФшЭ† ЯКСЮVUQt• Фйo6 L Aлэлu GlbSeeya† шLεeNIP?
```

android

```
$ openssl smime -verify -in CERT.RSA -inform DER -content CERT.SF
```

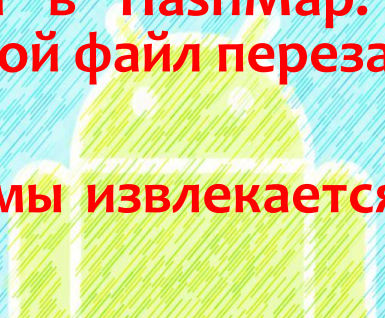
Verification successful

```
$ jarsigner -keystore debug.keystore -sigalg SHA1withRSA test.apk androiddebugkey
```

```
$ jarsigner -keystore debug.keystore -verify -verbose -certs test.apk
```


1st master key vulnerability (Jeff Forristal, Blackhat USA 2013)

- ❖ При проверке целостности файлов арк и цифровой подписи используется парсер zip-архивов, написанный на Java (ZipFile.java), помещающий все файлы в HashMap. В случае двух файлов с одинаковым именем, второй файл перезапишет первый
- ❖ При выполнении программы извлекается первый найденный файл с нужным именем



Имя	Размер	Сжат	Тип	Изменён	CRC32
..			Папка с файлами		
res			Папка с файлами		
META-INF			Папка с файлами		
lib			Папка с файлами		
assets			Папка с файлами		
resources.arsc	18 624	5 279	Файл "ARSC"	27.08.2012 20:32	3B5C9F4A
classes.dex	557 808	191 894	Файл "DEX"	05.08.2013 15:39	5D25F5BB
classes.dex	149 968	69 322	Файл "DEX"	27.08.2012 20:32	7E0435CA
AndroidManifest.xml	1 724	644	Документ XML	05.08.2013 15:39	C043CC53
AndroidManifest.xml	4 152	1 221	Документ XML	27.08.2012 20:32	3F3BA979

1st master key vulnerability

Возможности

Возможность без прав root и system (без лишения гарантии на мобильное устройство) подменить любое приложение на свое.

1. Кража сохраненных логинов и паролей в других приложениях или чтение и изменение любой информации.
2. Обход лицензионных ограничений.
3. Читерство в играх – редактирование сейвов без root.
4. Подмена системного приложения и получения с его помощью прав system.
5. После получения прав system добавление строки “ro.kernel.qemu=1\r\n” в файл /data/local.prop и получение прав root.

1st master key vulnerability

Что можно подменить?

classes.dex – внедрение своего java-кода (наиболее часто используют вирусы, распространенные в Китае)

lib – внедрение своего native c++ кода в любую из библиотек

AndroidManifest.xml – изменение названия приложения, его версии, системных требований ...

res – подмена существующих ресурсов – изменение иконки приложения и любых картинок, текстовых строк ...

1st master key vulnerability

Как эксплуатировать уязвимость?

1. Извлечь из легального арк-приложения оригинальный `classes.dex` и сохранить его как `classes_dex`.
2. Удалить из легального арк-приложения оригинальный `classes.dex`.
3. Добавить в арк-приложение вредоносный `classes.dex` (он должен идти первым в архиве).
4. Добавить в арк-приложение оригинальный `classes_dex` (он должен идти вторым в архиве).
5. С помощью любого hex-редактора открыть арк-приложение и в двух местах заменить “`classes_dex`” на “`classes.dex`”.

Переименование файла и замена строк необходима, так как большинство архиваторов не позволяют создавать в архиве несколько файлов с одинаковым именем.

1st master key vulnerability

Ограничения и недостатки

1. Нельзя добавлять новые файлы в арк или удалять существующие – можно только изменять файлы в архиве.

Не критично, если требуется выполнить консольное действие (скопировать сейв игры с карты памяти в защищенное хранилище или изменить какой-то файл) – достаточно подменить `classes.dex`.

Нет возможности полностью изменить программу, добавив свой графический интерфейс, – например, добавить новое меню в программу или написать чит с графическим интерфейсом или добавить новое видео в ресурсы.

2. Для изменения арк, необходимо подменить в нем каждый файл по отдельности.

1st master key vulnerability

Можно проще!

При обновлении приложений ОС Android проверяет целостность и корректность подписи нового устанавливаемого приложения, после чего проверяется совпадение сертификатов предыдущей и новой версий приложения. Попробуем подменить сертификат...

META-INF

Имя	Размер	Сжат	Тип	Изменён	CRC32
..			Папка с файлами		
MANIFEST.MF	683	368	Файл "MF"	07.08.2013 21:44	0F32A309
CERT.SF	736	400	Файл "SF"	07.08.2013 21:44	EE3B27CD
CERT.RSA	1 334	1 060	Файл "RSA"	07.08.2013 21:44	43428B44
CERT.RSA	1 714	1 154	Файл "RSA"	27.08.2012 20:32	D0BDFF04

1. Проверка цифровой подписи: используется первое вхождение файла CERT.RSA (что соответствует реализации парсера zip на C++)
2. Проверка совпадения сертификатов предыдущей и новой версий: используется второе вхождение файле CERT.RSA (что соответствует реализации парсера zip на Java).¹⁰

1st master key vulnerability

Новый способ эксплуатации

Новые возможности: Можно создать абсолютно любое приложение с любыми файлами

Как эксплуатируется?

1. В качестве id вредоносного приложения при его создании указывается id легального приложения, которое будет подменено.
2. Вредоносное приложение компилируется, создается арк-файл.
3. Из легального приложения извлекается CERT.RSA и сохраняется как CERT_RSA.
4. В арк-файл в папку META-INF добавляется CERT_RSA из легального приложения.
5. С помощью hex-редактора заменяем в арк-файле “CERT_RSA” на “CERT.RSA”

1st master key vulnerability

Обновление Android

- * ZipFile.java теперь запрещает наличие в одном apk-приложении нескольких файлов с одинаковым именем.
- *

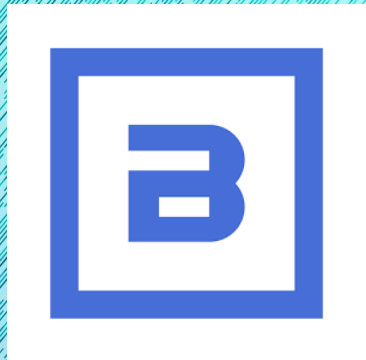
```
for (int i = 0; i < numEntries; ++i) {  
    * ZipEntry newEntry = new ZipEntry(hdrBuf,  
      bufferedStream);  
    * String entryName = newEntry.getName();  
    * if (entries.put(entryName, newEntry) != null) {  
      * throw new ZipException("Duplicate entry name:  
        " + entryName);  
    * }  
    * }  
  * }
```
- * Если в Android были установлены приложения, эксплуатирующие уязвимость, то они не удалятся, но смогут обновиться до следующих версий легальных приложений

Результаты исследования 1

- ✓ Обновление ОС Android, исправляющее bug 8219321, также предотвращает представленный способ эксплуатации данной уязвимости.
- ✓ Множество Android-устройств еще не обновилось / не выпущено обновление их производителем / прекращена поддержка производителем.
- ✓ Обновление для Samsung Galaxy S3 (прошивка I9300XXEMG4) вышло только в июле 2013, на <http://samsung-updates.com/device/?id=GT-I9300> появилось только в октябре 2013 для России (SER).
- ✓ Найденный способ эксплуатации работает не на всех устройствах: на уязвимой прошивке Samsung Galaxy S3 работает, на уязвимой прошивке HTC One X не работает.

Результаты исследования 2

- ✓ Информация о новом способе эксплуатации передана Google Security Team и Bluebox Security в августе 2013.



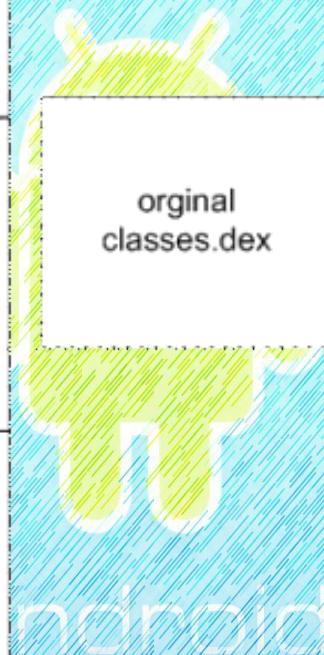
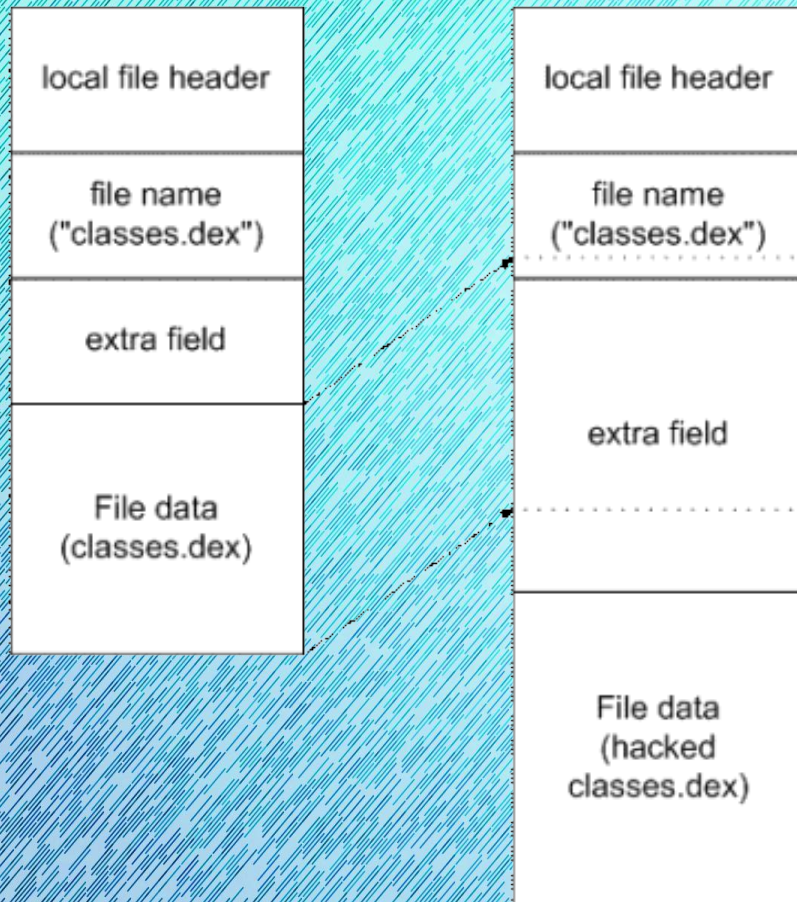
- ✓ Bluebox Security Scanner обновлен с версии 1.6 до версии 1.7 в сентябре 2013, до этого не обнаруживал приложения, использующие новый способ эксплуатации уязвимости.



- ✓ В августе-сентябре 2013 обновлен фильтр вредоносных приложений на Google Play Market, теперь в Маркет нельзя выкладывать приложения, содержащие эксплоит для bug 8219321.

- ✓ 2nd master key vulnerability (bug 9695860) практически нигде не исправлена, для Samsung Galaxy S 3 (SER - Russia) нет неуязвимой прошивки на <http://samsung-updates.com/device/?id=GT-I9300>.

Немного про 2nd master key vulnerability (bug 9695860)



Local file header

local file header signature bytes (0x04034b50)	4
version needed to extract	2 bytes
general purpose bit flag	2 bytes
compression method	2 bytes
last mod file time	2 bytes
last mod file date	2 bytes
crc-32	4 bytes
compressed size	4 bytes
uncompressed size	4 bytes
file name length	2 bytes
extra field length	2 bytes
file name (variable size)	
extra field (variable size)	
file data (variable size)	

Extra field length (0xFFFD): signed short in Java, unsigned short in C++

Исправление: All fields now unsigned

Дополнительные материалы

- ANDROID: ONE ROOT TO OWN THEM ALL / JEFF FORRISTAL / BLACKHAT USA 2013

<https://media.blackhat.com/us-13/US-13-Forristal-Android-One-Root-to-Own-Them-All-Slides.pdf>

- Android code signing

<http://nelenkov.blogspot.ru/2013/04/android-code-signing.html>

- Bluebox Security Scanner

<https://play.google.com/store/apps/details?id=com.bluebox.labs.onerootscanner>

- 2nd master key vulnerability

http://blog.sina.com.cn/s/blog_be6dacaee0101bksm.html

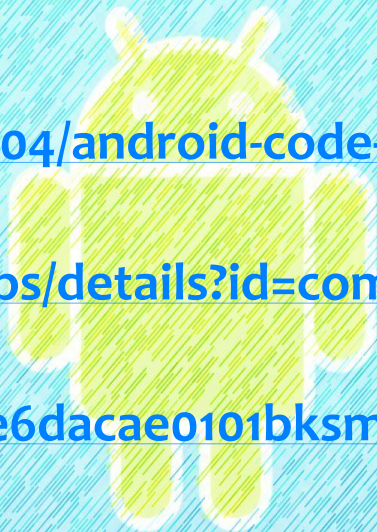
<http://www.saurik.com/id/18>

<http://nakedsecurity.sophos.com/2013/07/17/anatomy-of-another-android-hole-chinese-researchers-claim-new-code-verification-bypass/>

<http://www.h-online.com/open/news/item/Second-Android-signature-attack-disclosed-1918061.html>

- Universal fix для обеих уязвимостей (для установки требуются root-права и ОС Android >= 4.0)

<http://forum.xda-developers.com/showthread.php?t=2365294>





Спасибо за внимание!

Вопросы?

Contact: VMalyushin@mephi.ru