Визитка

АЛЕКСАНДР СВИРИДЕНКО, *программист-исследователь, компания «Доктор Веб»*

Детект уязвимости CVE-2014-8609
Sine ira et studio

В предыдущей статье [1] мы рассмотрели уязвимость CVE-2014-8609 (напомним, что для большинства устройств закрывающее ее обновление недоступно), которая позволяет полностью удалить все данные с устройства, а также создавать фейковые СМС

В начале 2015 года компания Samsung, стараясь обезопасить пользователей, начала выпускать патчи для своих довольно старых устройств. В том числе пришло исправление и на Samsung S3. Действительно, на данном устройстве после обновления код, разобранный нами в прошлой статье в качестве примера реализации уязвимости, уже не выполняется. Но проблема в том, что обновления устанавливают не все. Можно ли узнать, установлено интересующее нас обновление или нет?

После обновления прошивки нашего тестового устройства Samsung S3 версия его операционной системы не изменилась и осталась равна 4.3. Получается, что для детектирования уязвимости мы не можем ориентироваться на версию ОС, например, считать, что все устройства более ранних версий, чем Android 5.0, имеют уязвимость.

Каким образом можно определить наличие уязвимости CVE-2014-8609 на устройстве, естественно, не делая ничего вредного для пользователя?

Если посмотреть на изменения, которые были внесены для исправления этой проблемы (https://drw.sh/aknboe), то можно увидеть, что была добавлена строчка:

```
private static final String SHOULD_NOT_RESOLVE = ↵
        "SHOULDN'T RESOLVE!"
```

Этой переменной забивались поля Intent, чтобы ни у кого не было возможности их заменить. Но так решалась проблема в Android 5.0 . Не факт, что таким же способом закрыли уязвимость в старых устройствах. Теоретически, закрыть ее могли как угодно. Даже если и был выбран аналогичный способ, то строка могла быть совершенно другая. Для проверки извлекаем odex-файл из пропатченного Samsung S3 и декомпилируем его.

```
private void addAccount(String accountType) {

....

v8.setComponent(new ComponentName("SHOULDN\'T RESOLVE!", ↵
        "SHOULDN\'T RESOLVE!"));
v8.setAction("SHOULDN\'T RESOLVE!");
v8.addCategory("SHOULDN\'T RESOLVE!");
```

Видно, что инженеры из Samsung не стали ничего выдумывать и в качестве патча внесли точно такое же изменение. Нам повезло. И способ и строка оказались точно таким же, как и для 5-ых версий Android. Таким образом, для того чтобы определить, есть уязвимость или нет, можно просто осуществлять поиск интересующей нас строки в соответствующих ресурсах, так как стандартный обфускатор при компиляции приложения не шифрует строковые переменные. Достаточно открыть odex-файл и сделать по нему поиск (см. рис. 1).

Рисунок 1. Открытый для поиска odex-файл

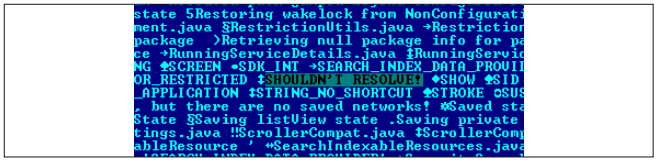
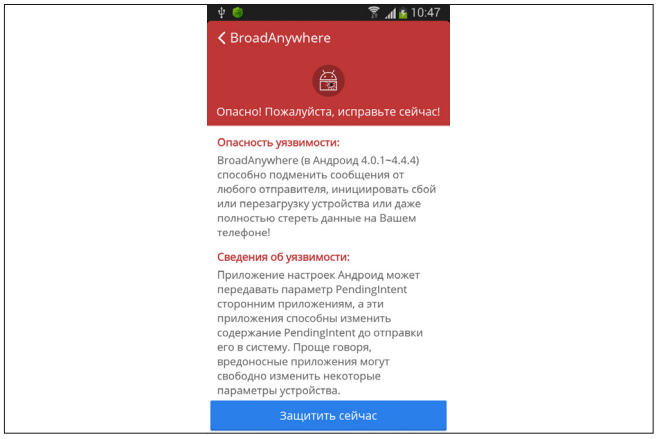


Рисунок 2. Предупреждение об уязвимости BroadAnywhere



Соответственно если мы видим данную строчку, то можем считать, что уязвимость закрыта (есть, правда, один нюанс, но о нем позже).

Насколько хорошо справляются с такой (достаточно простой) задачей сканеры уязвимостей? Поскольку критериев качества для большинства средств безопасности не существует, то пользователи достаточно часто используют бесплатные решения. ~~В связи с тем, что соответствующей строки бюджета не нашлось~~, без гнева и пристрастия возьмем два решения и посмотрим, как они работают.

В качестве первого примера поставим на пропатченный Samsung бесплатное решение, которое сейчас находится в российском топе (https://play.google.com/store/apps/collection/topselling_free).

И результат сразу неутешителен для тех, кто выберет данный продукт, – сканер показывает наличие уязвимости под именем BroadAnywhere (см. рис. 2).

В коде этого сканера за нахождение уязвимости отвечает следующий код:

```
public static boolean a() {
    int v0 = w.a();
    boolean v0_1 = 401 > v0 || v0 > 444 ? false : true;
    return v0_1;
}
```

Функция w.a() возвращает номер операционной системы, в котором вырезаны точки. То есть в Android 5.0.1 она вернет 501. Получается, что сканер показывает уязвимость на всех устройствах от версии 4.0.1 до 4.4.4. Это может свидетельствовать о технически низком уровне продукта. Проблема даже не в том, что анализ для версий выше 4.4.4 не производится, проблема в том, что подход к определению наличия уязвисмостей чисто формальный – по сути анализ вообще не производится, что при отсутствии специальных предварительных уведомлений от создателей продукта может ввести пользователей в заблуждение – понятно, что они не являются специалистами, способными понять, где выбранный ими продукт выполняет обещанный функционал, а где нет.

Теперь рассмотрим решение от уважаемой компании, о квалификации которой говорит то, что ее специалисты первыми сообщали об уязвимостях. Код их продукта очень сильно обфусцирован для противодействия злоумышленникам.

Пришлось смотреть, что происходит в памяти при работе программы. При запуске сразу обнаружилась строка "SHOULDN'T RESOLVE!". Способ детектирования верный, но дальнейший анализ показал неправильное определение пути к odex-файлу на Samsung S3 (там путь нетипичный). В поисках файла сканер просто перебирал пути, по которым эти odex-файлы обычно лежат в стоковом Android.

> //system/priv-app/Settings/arm/Settings.odex;
> //system/priv-app/Settings/arm64/Settings.odex;
> //system/priv-app/Settings/x86/Settings.odex.
> ...

На самом же деле Samsung разместил файл по адресу //System/app/SecSettings.odex. Пожелаем исправить эту ошибку в будущих версиях сканеров.

Вывод: пользователям предлагается достаточно много бесплатных решений безопасности. Но бесплатный сыр бывает только в мышеловке. Работа квалифицированных специалистов бесплатной быть не может, и полностью доверять бесплатным системам безопасности, по нашему мнению, опрометчиво. EOF

1. Свириденко А. Разбор уязвимости CVE-2014-8609, или Когда можно будет спать спокойно. // «Системный администратор», №1-2, 2015 г. – С. 54-55 (http://samag.ru/archive/article/2865).