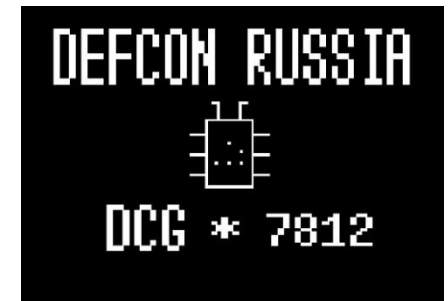


Мобильный банкинг: Кража по воздуху

Дмитрий Евдокимов
Директор исследовательского центра
Digital Security

#whoami

- Исследователь информационной безопасности в **Digital Security Research Group**
- Редактор рубрик в журнале **Xakep**
- Один из организаторов конференций **DEFCON Russia** и **ZeroNights**
- Специализируюсь на поиске **уязвимостей** в бинарных приложениях без исходного кода
- Анализ **мобильных** приложений для Android, iOS, WindowsPhone
- Докладчик на **конференциях** в Польше, Франции, Германии, ОАЭ



Исследования



Важна ли безопасность мобильного банкинга?

Распространенное мнение/миф: НЕТ

- Там ходит мало денег

Реальная ситуация: ДА

- ЦБ РФ не делает различий между ДБО и МБ
 - Рекомендации по безопасной разработке платежных систем
- Злоумышленнику неважно, ходят ли там вообще деньги
 - Главное, что ПО работает со счетом, где лежат деньги
- Достаточно одного взлома ключевого клиента банка
 - Ущерб репутации и переход клиента в другой банк
- ДБО и МБ очень связаны
 - Информацию/уязвимости в одном можно использовать для атаки на другое

Начальные цифры



79



61



59

Условия

Устройство:

- Устройство пользователя не имеет ни jailbreak, ни root-доступа
- Устройство имеет все последние обновления системы
- Устройство не заражено никакими вредоносными программами

Проверка:

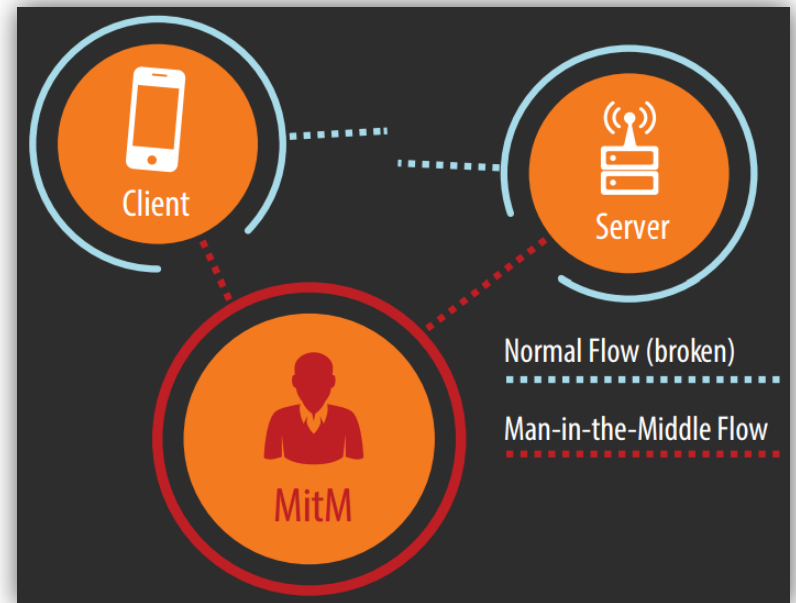
- Независимый анализ
- Проверка атаки «человек посередине» (Man-in-the-Middle, MitM)
 - Некорректная проверка SSL-сертификата
 - Использование SSL Pinning
- Динамический анализ
 - На стадии аутентификации
 - Не надо быть клиентом банка!
 - Повторить может любой желающий!

M3 – Insufficient Transport Layer Protection



Атака MitM

- Подключение пользователя к поддельной Wi-Fi-точке доступа
- Подключение к поддельной базовой станции оператора
- Использование зараженного сетевого оборудования



При контролировании канала передачи данных между приложением мобильного банкинга и сервером злоумышленник может **украсть деньги** со счета клиента, то есть нанести прямой финансовый ущерб

Последствия MitM



Кража денежных
средств со счетов
клиента



Раскрытие
информации о счетах
клиента и его прошлых
операциях



Просмотр данных
о текущей операции
клиента

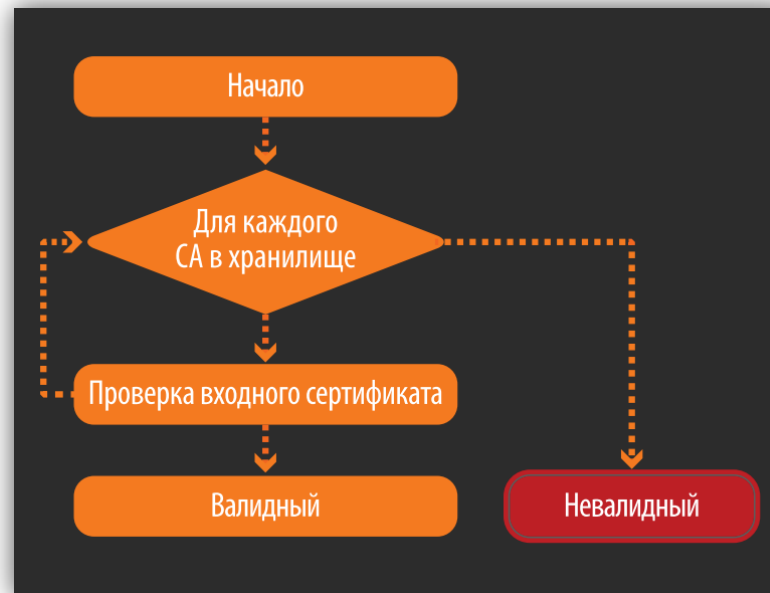


Отказ
в обслуживании
клиента

Виды каналов связи

- Открытые
- Защищенные нестандартными методами
 - ~ = открытые
- Защищенные стандартными методами
 - SSL/TLS
 - Все взаимодействие основывается на паре «открытый и закрытый ключ» сертификата сервера
 - Сертификат должен быть **действительным**

SSL-сертификаты на устройстве



 4.0.3	134
 4.2.2	140
 4.4.2	150

 IOS 5	183
 IOS 6	183
 IOS 7	211

Некорректное использование SSL

- Использование уязвимых фреймворков
- Отключение проверок (отладочное API)
- Некорректное переопределение стандартных обработчиков на собственные
- Неправильная конфигурация API-вызовов
- Слабые параметры шифрования
- Использование уязвимой версии библиотеки
- Неправильная обработка результатов вызовов
- Отсутствие проверки на имя хоста или использование неправильных регулярных выражений для проверки



Проблемы frameworks

- Очень-очень модно
- А что там под капотом? Что там с безопасностью?
- RCE Apache Cordova
 - Cross-Application Scripting – выполнение вредоносного JS в контексте Cordova-based приложения
 - <http://cordova.apache.org/announcements/2014/08/04/android-351.html>
- Titanium
 - RCE + отсутствие проверки сертификата
 - <http://www.appcelerator.com/blog/2012/11/the-titanium-sdk-and-certificate-validation/>
- ...

Проблемы с SSL

- Некорректное использование SSL
 - Невнимательность и ошибки разработчика
- Компрометация корневого сертификата
 - Установка вредоносного сертификата с помощью соц. инженерии
 - Инциденты с Bit9, DigiNator и Comodo
 - Сертификаты иностранных государств ;)

```
Certificate:
Data:
  Version: 3 (0x2)
  Serial Number: 49 (0x31)
  Signature Algorithm: sha1WithRSAEncryption
  Issuer: C=JP, O=Japanese Government, OU=ApplicationCA
  Validity
    Not Before: Dec 12 15:00:00 2007 GMT
    Not After : Dec 12 15:00:00 2017 GMT
  Subject: C=JP, O=Japanese Government, OU=ApplicationCA
  Subject: C=US, O=U.S. Government, OU=DoD, OU=PKI, CN=DoD Root CA 2
```

SSL Pinning

В качестве защиты от компрометации корневых системных сертификатов и специально встроенных пользовательских можно использовать подход SSL Pinning.

Pinning – это процесс ассоциации хоста с его ожидаемым X509-сертификатом или публичным ключом.

- app1 использует SSL Pinning
 - Проверяет «вшитый» сертификат
- app2 не использует SSL Pinning
 - Обращается к системному хранилищу



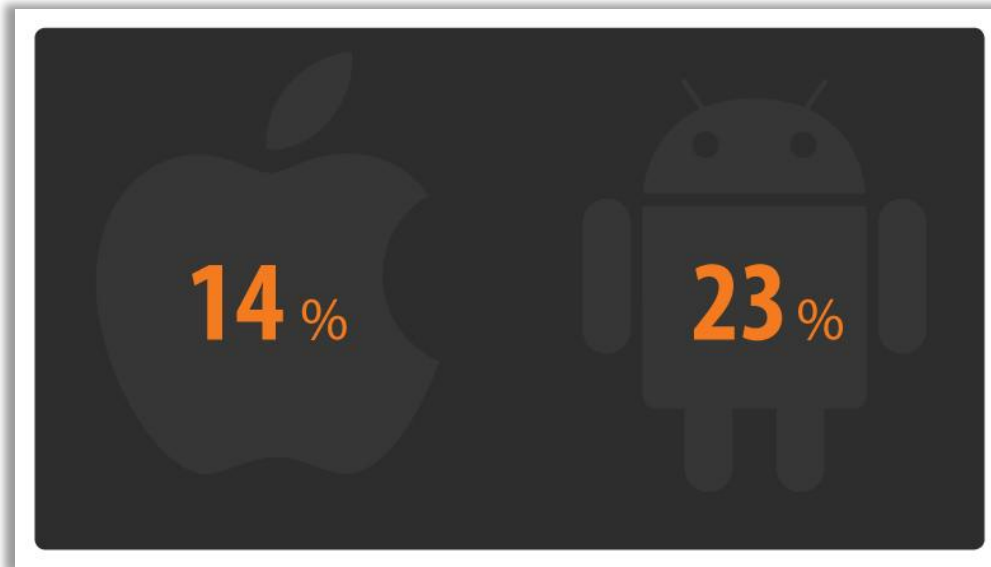
А как же многофакторная аутентификация?

Вход в систему	
Фактор знания Пароль/PIN-код/...	****
Вещественный фактор Аппаратный ключ/...	
Биометрический фактор Голос/Сетчатка глаза/Почерк/...	
Ситуативный фактор Местоположение/Движение/...	

- В классической системе ДБО второй фактор часто приходит в виде SMS на телефон
 - В случае мобильного банкинга второй фактор идет по тому же каналу, что и контролирует злоумышленник :(

Результаты

УЯЗВИМЫ



- 1 уязвимость → MitM = **кража** денег
- Сочетание других уязвимостей может также привести к **краже** денег со счетов клиентов!

Результаты: интересные находки

- Почти все приложения отправляют на сервер информацию об устройстве, порой и уникальные идентификаторы устройства
- Для входа в мобильный банк часто используются учетные данные с интернет-банкинга
- Атака “User enumeration” – получение списка действующих логинов
- Возможность блокировки аккаунтов пользователей
- Интересные ошибки на серверах
 - Отладочные трейсы
 - Раскрытие внутренней банковской информации (даже информация об **АБС**)
- Один разработчик (даже одна платформа), а уязвимости разные
- ...

Безопасная разработка



- iOS
 - Secure Coding Guide from Apple
 - <https://developer.apple.com/library/ios/documentation/Security/Conceptual/SecureCodingGuide/SecureCodingGuide.pdf>
 - IOS Developer Cheat Sheet
 - https://www.owasp.org/index.php/IOS_Developer_Cheat_Sheet
- Android
 - The CERT Oracle Secure Coding Standard for Java
 - <https://www.securecoding.cert.org/confluence/pages/viewpage.action?pageId=111509535>
 - Analysis of Android Applicability: CERT's Java Coding Guidelines
 - <https://www.securecoding.cert.org/confluence/display/java/Analysis+of+Android+Applicability%3A+CERT%27s+Java+Coding+Guidelines>
 - Security Tips from Android
 - <http://developer.android.com/training/articles/security-tips.html>

Рекомендации

- Использовать SDL (Security Development Lifecycle)
 - Также касается систем ДБО
- Для разработки такого критичного ПО нужны подготовленные тестовые среды
- Удалять отладочный код
- Тестировать на проникновение приложения и серверную сторону
- Грамотно использовать многофакторную аутентификацию
- Использовать SSL Pinning

Мобильный банкинг: Кража по воздуху

Спасибо за внимание!
Вопросы?

Digital Security в Москве: (495) 223-07-86
Digital Security в Санкт-Петербурге: (812) 703-15-47

d.evdokimov@dsec.ru
@evdokimovds