Can’t Touch This: Cloning Any Android HCE Contactless Card

Sławomir Jasek
slawomir.jasek@securing.pl
@slawekja

https://www.youtube.com/watch?v=tELZEPcgKkE

HackInTheBox Amsterdam, 13.04.2017
Cloning

1996: Dolly the sheep
Cloning

1996: Dolly the sheep
2001: cat „CopyCat“
2003: horse
Humans? Why no humans?

I. Commercial – no commercial interest in industries

II. Ethical/legal – beliefs, laws...

III. Technical - pets easy, primates very hard
Are we sure?
2017 – mobile contactless payment cards cloning?

I. Commercial

2017 – mobile contactless payment cards cloning?

I. Commercial

II. Ethical/legal

III. Technical

Technical countermeasures

- Tamper
- Tokenization
- Limited-use
- Cloud
- Secure
- Fraud
- Online
- Transactions
- Contactless
- Management
- Provisioning
- Detection
- Prevention

[Image: https://www.flickr.com/photos/un_photo/6872103103/]

[Website: www.wordclouds.com]
HCE TECHNOLOGY
Secure Element (since 2007)

SE communicates directly with NFC
• Apps and OS have no access to card data and to communication during payment.
SE dominance hierarchy clashes

Banks vs mobile operators, handset manufacturers, payment service providers...

Painful process
- special SIM required
- limited support

https://www.flickr.com/photos/jsouthorn/6616455243/
Google vs Isis Wallet

2011: Google Wallet with Galaxy Nexus embedded SE

Isis wallet (AT&T, Verizon, T-Mobile) - blocked Google Wallet for their devices.

Google: we will go our way - without SE.
Host Card Emulation

Android >=4.4, Blackberry OS, Windows Phone

No need for troublesome Secure Element, moved to „cloud”.

Software emulates contactless smart card.
How does it work?

Demo
INTEGRATION
How to embed it in mobile app?

- Own implementation
- External, „blackbox” library
  - Visa, Mastercard SDK
  - several other products
Vendors’ doc
Vendors’ doc

„Secure Element in the Cloud”

Mobile phone (Android 4.4+)

HCE Applet

Bank’s mobile app

HCE API

NFC controller

SDK

API
Sławomir Jasek

Enjoy appsec (dev, break, build...) since 2003.

Pentesting, consultancy, training - web, mobile, embedded...

Significant part of time for research.
HOW TO STEAL THE MONEY?
Right, so how to steal the money?

Steal the phone?
immediate report and cancel

U.S. card fraud by type, 2014

- Lost/stolen: 14%
- Counterfeit: 37%
- Online (card not present): 45%
- Other: 10%

Source: Aite Group, "EMV: Lessons Learned and the U.S. Outlook", June 2014.

https://twitter.com/therealbanksy/status/842853661407678464
Steal card data via NFC?

Credit card reader?

Let’s try!

The screen has to be on. In some cases unlock is required.

You won’t make online payments using it.

Creating magstripe track may be possible.

Tokenization

Random card numbers (tokens) replacing single static PAN

Limited "domain" use – only for contactless transactions
So, how are the EMV transactions executed?
The key

How to steal it?
Intercept in transfer?

Google Cloud Messaging

Mobile wallet server

"Secure Element in the cloud" server
Typically

Multiple servers, push included

Certificate pinning

Second layer encryption

"encryptedData":"AAABdxcgfXea9B050gH9/a1fcJz//UpQihZrvfdHwZboTo3kNN45M0eemFMrM1EM0BzixsDHTMFeUen19CKMjsbJT/IvZZGceL5KmQK971NoI5wo8Kh5qqF/hazsU2u0lyu5NxsE69QE62cffruh55DvX8f7/g="
Flaws?

- Improper pinning – accept all certs, use vulnerable lib...

Nasty bugs „deeply hidden” under the proprietary encryption layer.

- Difficult to exploit, need active access to transmission during provisioning.
The key

How to steal it?
  • Intercept in transfer?

Stored in user-space – not hardware Secure Element.
  • Get it from the phone?
Mobile malware?

Most common:
- Overlay stealing data
- Intercept SMS
- ...
- Does not have access to card data (private folder of the app)
How to access the data?

root = GOD

https://en.wikipedia.org/wiki/The_Creation_of_Adam
Rooting possibilities?

The Best One Click Android Root Tool For ALL Android Devices
Quantity of Supported Models

104136
Malware with root?

10 million Android phones infected by all-powerful auto-rooting apps

First detected in November, Shedun/Humstron ‘GODLESS’ Mobile Malware Uses Multiple Exploits to Root Devices

Tordow Banking Trojan – A Grave Threat for Android Users

Those downloading Android apps from third-party stores are the one affected by this trojan.
Key stored on the device

Stored in user-space – not hardware Secure Element.

How to steal it?
• Intercept in transfer?
• Get it from the phone? Root malware!

But the key is encrypted... How to decrypt?
Decompile the app binary?
VIENTO REACTIVA FOCO DE INCENDIO EN PORTEZUELO
How does it work?
Encryption

Does not require user interaction (no PIN/pass).

Works also when phone is offline.

So, what can we do to clone the card?
Install the same app, copy data?

No, it does not work 😞

The key is tied to specific device
Encryption again

- Keypair
- Data stored on the phone
- APK hardcoded in app
- Device-specific characteristic

secured app ID
How about exactly same hardware device?

Same model, OS version, IMEI...
Works!

In most cases you need to copy also other user data (not just the payment app)

Not really practical attack on a mass scale...
How to make it to a different device?
Device characteristics?

- AndroidID
- DeviceID (IMEI)
- Phone number
- MAC address
- Manufacturer, Model
- Serial
- OS version, build
Device characteristics?

- AndroidID
- DeviceID (IMEI)
- Phone number
- MAC address
- Manufacturer, Model
- Serial
- OS version, build

May change in time
Device characteristics?

AndroidID
DeviceID (IMEI)
Phone number
MAC address
Manufacturer, Model
Serial
OS version, build

02:00:00:00:00:00 (privacy)

May change in time

Mostly inaccessible
Device characteristics?

- AndroidID
- DeviceID (IMEI)
- Phone number
- MAC address
- Manufacturer, Model
- Serial
- OS version, build

Most common:
- May change in time

Require special privileges, e.g.
- „Make phone calls, ..”
- Mostly inaccessible
- 02:00:00:00:00:00 (privacy)

Non-standard, mostly not used

May change in time
Xposed Framework

Change behavior of system and apps
Hooks into system calls.

Requires root.
Standard device

OS

DeviceID?

3882343...

Mobile app

Xposed - framework

+ module

changing ID

OS

Xposed

DeviceID?

6666666...

Mobile app
Xposed – helps to imitate original device

The key is tied to specific device
ROOT DETECTION
Root detection?

Having ultimate control you can always hide from detectors.

Detection checks for popular rooting ways

http://simpsons.wikia.com/wiki/The_Itchy_%26_Scratchy_Show
SafetyNet root detection

private static final String[][] a = {
    "/system/bin/su",
    "/system/xbin/su",
    "/system/bin/./su",
    "/system/bin/.su",
    "/system/xbin/./su" };
Live demo

https://www.flickr.com/photos/136682034@N03/26086288495/
Cloning script

tar  
chown  
restorecon
backup.sh

<redacted>
restore.sh

<redacted>
The secret ingredient

Exactly same OS and Google Services version
Real risk?

PoC was on a single, small amount transaction from the same network and physical location.

Google definitely has some FDS/behavioral analysis systems.
Finally, we can use the card on other device!

... but the keys are limited-use.

Only a few transactions < 25 EUR each?

Then the keys have to be replenished.

So, how does it work?
Keys replenish – most common: GCM combined

- Mobile wallet server
- Google Cloud Messaging
- "Secure Element in the cloud" server
This part we already have

Mobile wallet server

Not able to determine attacker vs original user
How to hijack GCM push?
Hijack GCM push

Copy relevant user data (/data/system/users, ...)

Both devices have same AndroidID, keys, subscriptions

Test push received by:
- sometimes both
- only one (mostly „cloned“)
- I can block original user
Having root access to victim’s phone

- Make few low-value transactions from another device
- Make multiple transactions (renew limited-use keys)
- But... there are usually limits on number of transactions
FLOOR LIMIT
The „floor limit”

Transactions > 25 EUR need authorization

Several options:

• Enter card PIN in terminal

BUT - how do you set up the PIN?

Mobile malware -> can sniff the PIN / trick user into entering it
The „floor limit”

Transactions > 25 EUR need authorization

Several options:

• Enter card PIN in terminal
• CDCVM
CDCVM
Consumer Device Cardholder Verification Method

Malware = steal the PIN

On-device cardholder verification (e.g. PIN)

EMV crypto

Cardholder verification

Consumer Device Verified
Having root access to victim’s phone

- Make few low-value transactions from another device
- Make multiple transactions (renew limited-use keys)
- Make transactions on higher amounts
CDCVM – not very common

< 20% apps support it

So what if application does not support CDCVM?
API method names (cannot be obfuscated)

`setCvmVerificationMode(CvmMode paramCvmMode);`

`setCvmVerified(boolean paramBoolean);`
const/4 v9, 0x1
invoke-interface {v8,v9}, L<redacted>;-&gt;setCvmVerified(Z)V
new-instance v9, L<redacted>/CvmMode;
sget-object v10, L<redacted>/VerifyingEntity;-&gt;MOBILE_APP:
  L<redacted>/VerifyingEntity;
sget-object v11, L<redacted>/VerifyingType;-&gt;PASSCODE:
  L<redacted>/VerifyingType;
invoke-direct {v9, v10, v11}, L<redacted>/CvmMode;-&gt;<init>
  (L<redacted>/VerifyingEntity;L<redacted>/VerifyingType;)V
invoke-interface {v8, v9}, L<redacted>;-&gt;setCvmVerificationMode
  (L<redacted>/CvmMode;)V
CDCVM in app which does not support it ;)
Results are inconsistent...

• Terminal did not ask for PIN
• Transaction was declined (but the card was incorrect anyway)

Definitely worth digging deeper
OTHER APPS
Other applications

Most banks think of/are during/after implementation.

We have physically proved cloning possible in 8 apps (and 7 libs).

Others we can estimate based on libs used (PoC requires account in bank).
The easiest one

- No root detection
- Simple device checks
- No GCM push for replenish

The hardest one

Checks multiple device characteristics

Native lib root detection

Good integrity checks and obfuscation

Had to use unrooted phone - same model, with cloned IMEI
WHAT CAN WE DO BETTER?
Don’t be the last one...
Check for more device characteristics?

- Device serial
- SIM serial/IMSI
- Display size?
- CPU?
- Sensors?

https://www.flickr.com/photos/volvob12b/11248541865/
Improve root detection

Craft your own
SafetyNet
  • will definitely improve
RootBeer
  • Open-source

https://github.com/scottyab/rootbeer/
Integrity checks, binary protections...

- Code obfuscation
- Install source, signing keys
- Tamper, debug detection
- Notifications, reporting
- Wipe on compromise

https://www.flickr.com/photos/carolynwill/1118743053
Backend - fraud management

Detect duplicated card use not enough

Device scoring - os version, patch level, bootloader unlock, installed soft

Malware handling

Behavioral analysis

https://www.flickr.com/photos/widnr/6545526341/
Future?

Devices will be more resilient, TPM?

More widespread mobile payments = more attention of fraudsters.

Hope for the best, but prepare - and verify - for the worst!

"with great power comes great responsibility"

http://www.techiestate.com/spiderman-android-game/
Lunch time!

https://www.flickr.com/photos/34739556@N04/5361451866/
MORE THAN SECURITY TESTING

Thank you! Questions?

Slawomir.Jasek@securing.pl    @slawekja