Let's Integrate Everything!

(ubiquitous|anywhere|pervasive) computing
The "Internet of Things"
Web 2.0?

Let's Synchronize...
- your browser
- ...

Because that is the smart in smartphone

... with your phone!

The Google Way
Manage your phone from your browser
- Locate it
- Wipe it (in case of emergency)
- Install apps!

The Google Way
Sure this is safe
You can only remote install apps from Play
Google Bouncer will protect you

Apps are **inactive** after installation
- A user must start them once explicitly
- Only then can we start on boot, intercept messages, ...

2. Application
App is installed via remote-install

We need only **one** user interaction:
1. Open the app directly (via app-icon)
2. Click a custom URI (myapp://open.me)

1. Bypassing Bouncer
Already done in the past, multiple times
Bouncer evolves though

Assume that Bouncer catches malicious code
Why not upload a vulnerable app instead?

~~You can find us in Google Play!~~

~~Live~~ Demo
We are on good terms with the banks...
...so let's break something more fun instead

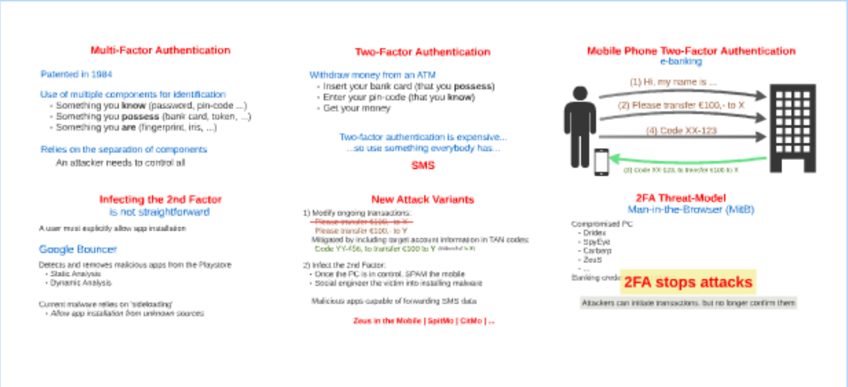
Google Authenticator!

A few hours after talking to Nick - head of Android platform security - ...our app and developer account got banned...

Elevate MitB to MitMo
And intercept SMS messages
Assuming control over the browser, we need to:
1. Bypass Bouncer
2. Steer the user into activating the app
Required only once
3. Intercept SMS... and profit!

3. Intercept Messages
Control over the phone
Install a SMS receiver, for each incoming SMS:
1. store it
2. upload request to our malicious server
Control over the browser
1. Log into e-banking environment
2. Initiate transaction
3. Confirm with intercepted TAN

The Fix
Google
1. Always require on-phone confirmation for app installs
2. Do not allow app activation through clicked URIs
3. Disable the remote install feature
4. Perhaps look at our hiding tricks?
The user
1. Watch out for unknown app installs
2. Use a separate account for Android
The Google Authenticator user
Use a non-android phone for your backup phone number



Multi-Factor Authentication

Patented in 1984

Use of multiple components for identification
- Something you **know** (password, pin-code ...)
- Something you **possess** (bank card, token, ...)
- Something you **are** (fingerprint, iris, ...)

Relies on the separation of components
An attacker needs to control all

Two-Factor Authentication

Withdraw money from an ATM
- Insert your bank card (that you **possess**)
- Enter your pin-code (that you **know**)
- Get your money

Two-factor authentication is expensive...
...so use something everybody has...

SMS

Mobile Phone Two-Factor Authentication
e-banking

(1) Hi, my name is ...
(2) Please transfer €100,- to X
(4) Code XX-123
(3) Code XX-123, to transfer €100 to X

Infecting the 2nd Factor
is not straightforward

A user must explicitly allow app installation

Google Bouncer

Detects and removes malicious apps from the Playstore
- Static Analysis
- Dynamic Analysis

Current malware relies on 'sideloading'
- *Allow app installation from unknown sources*

New Attack Variants

1) Modify ongoing transactions:
~~Please transfer €100,- to X~~
Please transfer €100,- to Y
Mitigated by including target account information in TAN codes:
Code YY-456, to transfer €100 to Y

2) Infect the 2nd Factor:
- Once the PC is in control, SPAM the mobile
- Social engineer the victim into installing malware

Malicious apps capable of forwarding SMS data

Zeus in the Mobile | SpitMo | CitMo | ...

2FA Threat-Model
Man-in-the-Browser (MitB)

Compromised PC
- Dridex
- SpyEye
- Carberp
- ZeuS
- ...

Banking crede

2FA stops attacks

Attackers can initiate transactions, but no longer confirm them

How Google Killed Two-Factor Authentication

(and the reactions)

http://www.few.vu.nl/~vvdveen/bandroid.html

Conclusions

We think that this is a serious issue

Hard to convince 'experts'

Mixed reactions from Google, but we have their attention

iOS and Windows Phone have similar remote install features

...but no API to read SMS messages YET

Easy, version-independent fix: explicit activation

Mobile-phone based 2FA seems doomed

financial institutions will come to the same conclusion

The Media is (mostly) clueless

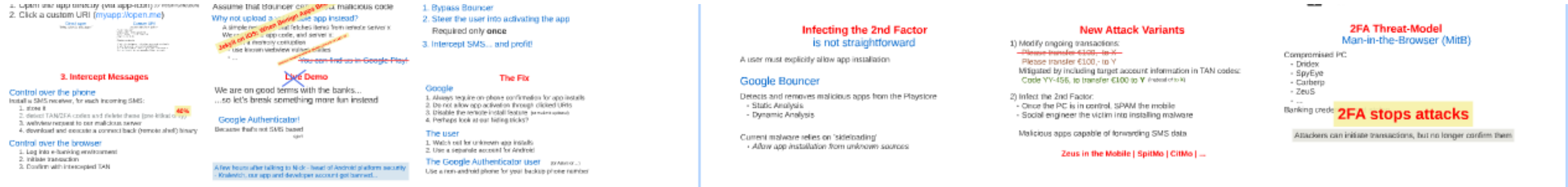
Radhesh Krishnan
Herbert Bos

Victor van der Veen

VU University Amsterdam

System and Network Security Group

Andrubis | TraceDroid (app analysis)
PathArmor @ CCS '15 (context-sensitive CFI)



2. Click a custom URI (myapp://open.me)

Assume that Bouncer catches malicious code

Why not upload a benign looking app instead?

~~You can find us in Google Play!~~

1. Bypass Bouncer
2. Steer the user into activating the app

Required only **once**

3. Intercept SMS... and profit!

3. Intercept Messages

Control over the phone

Install a SMS receiver, for each incoming SMS:

1. store it
2. detect TAN/2FA codes and delete these
3. webview request to our malicious server
4. download and execute a connect back (remote shell) binary

Control over the browser

1. Log into e-banking environment
2. Initiate transaction
3. Confirm with intercepted TAN

~~Live~~ Demo

We are on good terms with the banks...
...so let's break something more fun instead

Google Authenticator!

Because that's not SMS based

A few hours after talking to Nick - head of Android platform security - Kralevich, our app and developer account got banned...

The Fix

Google

1. Always require on-phone confirmation for app installs
2. Do not allow app activation through clicked URIs
3. Disable the remote install feature
4. Perhaps look at our listing tricks?

The user

1. Watch out for unknown app installs
2. Use a separate account for Android

The Google Authenticator user

Use a non-android phone for your backup phone number

Infecting the 2nd Factor is not straightforward

A user must explicitly allow app installation

Google Bouncer

Detects and removes malicious apps from the Playstore
- Static Analysis
- Dynamic Analysis

Current malware relies on 'sideloading'
- *Allow app installation from unknown sources*

New Attack Variants

1) Modify ongoing transactions:

~~Please transfer €100,- to X~~

Please transfer €100,- to Y

Mitigated by including target account information in TAN codes:

Code YY-456, to transfer €100 to Y

2) Infect the 2nd Factor:
- Once the PC is in control, SPAM the mobile
- Social engineer the victim into installing malware

Malicious apps capable of forwarding SMS data

Zeus in the Mobile | SpitMo | CitMo | ...

2FA Threat-Model

Man-in-the-Browser (MitB)

Compromised PC
- Dridex
- SpyEye
- Carberp
- ZeuS
- ...

2FA stops attacks

Attackers can initiate transactions, but no longer confirm them

How Google Killed Two-Factor Authentication

(and the reactions)

http://www.few.vu.nl/~vvdveen/bandroid.html

Conclusions

We think that this is serious

Hard to convince 'experts'

Mixed reactions from Google, but we have their attention

iOS and Windows Phone have similar remote install features

...but no API to read SMS messages **YET**

Easy, version-independent fix: explicit activation

Mobile-phone based 2FA seems doomed

financial institutions will come to the same conclusion

The Media is (mostly) clueless

Radhesh Krishnan
Herbert Bos

Victor van der Veen

VU University Amsterdam

System and Network Security Group

Andrubis | TraceDroid (app analysis)
PathArmor @ CCS '15 (context-sensitive CFI)

www.few.vu.nl/~vvdveen/banc

We think that this is a serious bug

Hard to convince 'experts'

Mixed reactions from Google, but we have their attention

iOS and Windows Phone have similar remote install features

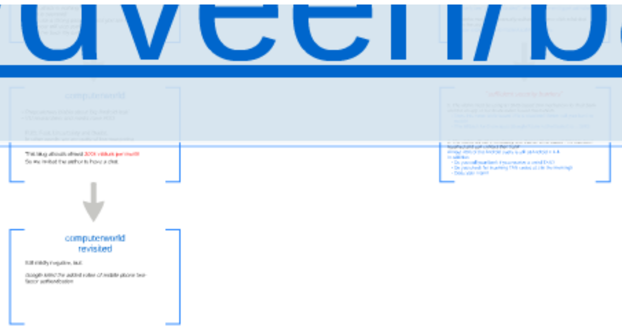
...but no API to read SMS messages **YET**

Easy, version-independent fix: explicit activation

Mobile-phone based 2FA seems doomed

financial institutions will come to the same conclusion

The Media is (mostly) clueless



Radhesh Krishnan
Herbert Bos
Victor van der Veen
VU University Amsterdam

System and Network Security Group

Andrubis | TraceDroid (app analysis)

PathArmor @ CCS '15 (context-sensitive CFI)

Herbert Bos

Victor van der Veen

VU University Amsterdam

System and Network Security Group

Andrubis | TraceDroid (app analysis)

PathArmor @ CCS '15 (context-sensitive CFI)

1. Open the app directly (via app-icon)
2. Click a custom URI (myapp://open.me)

3. Intercept Messages

Control over the phone

Install a SMS receiver, for each incoming SMS:

1. store it
2. detect TAN/2FA codes and delete these (pre-kitkat only) 46%
3. webview request to our malicious server
4. download and execute a connect back (remote shell) binary

Control over the browser

1. Log into e-banking environment
2. Initiate transaction
3. Confirm with intercepted TAN

Assume that Bouncer catches malicious code

Why not upload a vulnerable app instead?

A simple native app that fetches items from remote server x

We control app code, and server x:

- a memory corruption
- use known webview vulnerabilities
- ...

Jekyll on iOS: When Benign Apps Become Evil

~~You can find us in Google Play!~~

~~Live~~ Demo

We are on good terms with the banks...
...so let's break something more fun instead

Google Authenticator!

Because that's not SMS based

A few hours after talking to Nick - head of Android platform security - Kralevich, our app and developer account got banned...

1. Bypass Bouncer
2. Steer the user into activating the app
   Required only **once**
3. Intercept SMS... and profit!

The Fix

Google

1. Always require on-phone confirmation for app installs
2. Do not allow app activation through clicked URIs
3. Disable the remote install feature
4. Perhaps look at our hiding tricks?

The user

1. Watch out for unknown app installs
2. Use a separate account for Android

The Google Authenticator user

Use a non-android phone for your backup phone number

Infecting the 2nd Factor
is not straightforward

A user must explicitly allow app installation

Google Bouncer

Detects and removes malicious apps from the Playstore

- Static Analysis
- Dynamic Analysis

Current malware relies on 'sideloading'

- *Allow app installation from unknown sources*

New Attack Variants

1) Modify ongoing transactions:

~~Please transfer €100,- to X~~
Please transfer €100,- to Y
Mitigated by including target account information in TAN codes:
Code YY-456, to transfer €100 to Y

2) Infect the 2nd Factor:

- Once the PC is in control, SPAM the mobile
- Social engineer the victim into installing malware

Malicious apps capable of forwarding SMS data

Zeus in the Mobile | SpitMo | CitMo | ...

2FA Threat-Model
Man-in-the-Browser (MitB)

Compromised PC

- Dridex
- SpyEye
- Carberp
- ZeuS
- ...

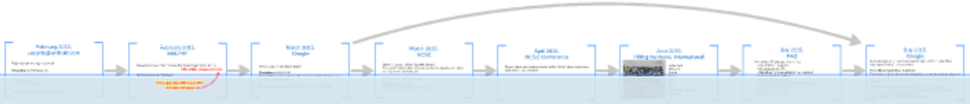
Banking crede

2FA stops attacks

Attackers can initiate transactions, but no longer confirm them

How Google Killed Two-Factor Authentication

(and the reactions)



http://www.few.vu.nl/~vvdveen/bandroid.html

Conclusions

We think that this is serious

Hard to convince 'experts'

Mixed reactions from Google, but we have their attention

iOS and Windows Phone have similar remote install features

...but no API to read SMS messages **YET**

Easy, version-independent fix: explicit activation

Mobile-phone based 2FA seems doomed

financial institutions will come to the same conclusion

The Media is (mostly) clueless

Radhesh Krishnan
Herbert Bos

Victor van der Veen

VU University Amsterdam

System and Network Security Group

Andrubis | TraceDroid (app analysis)
PathArmor @ CCS '15 (context-sensitive CFI)

Multi-Factor Authentication

Patented in 1984

Use of multiple components for identification
- Something you **know** (password, pin-code ...)
- Something you **possess** (bank card, token, ...)
- Something you **are** (fingerprint, iris, ...)

Relies on the separation of components

An attacker needs to control all

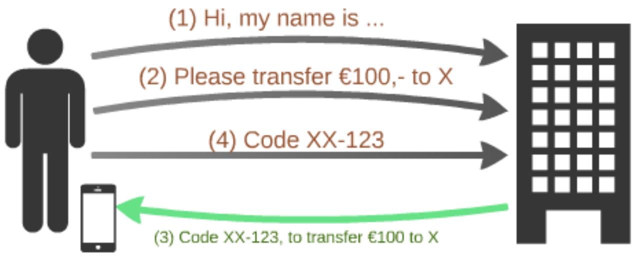
Two-Factor Authentication

Withdraw money from an ATM
- Insert your bank card (that you **possess**)
- Enter your pin-code (that you **know**)
- Get your money

Two-factor authentication is expensive...
...so use something everybody has...

SMS

Mobile Phone Two-Factor Authentication
e-banking


(1) Hi, my name is ...
(2) Please transfer €100,- to X
(4) Code XX-123
(3) Code XX-123, to transfer €100 to X

Infecting the 2nd Factor
is not straightforward

A user must explicitly allow app installation

Google Bouncer

Detects and removes malicious apps from the Playstore
- Static Analysis
- Dynamic Analysis

Current malware relies on 'sideloading'
- *Allow app installation from unknown sources*

New Attack Variants

1) Modify ongoing transactions:
~~Please transfer €100,- to X~~
Please transfer €100,- to Y
Mitigated by including target account information in TAN codes:
Code YY-456, to transfer €100 to **Y** (instead of to X)

2) Infect the 2nd Factor:
- Once the PC is in control, SPAM the mobile
- Social engineer the victim into installing malware

Malicious apps capable of forwarding SMS data

Zeus in the Mobile | SpitMo | CitMo | ...

2FA Threat-Model
Man-in-the-Browser (MitB)

Compromised PC
- Dridex
- SpyEye
- Carberp
- ZeuS
- ...

Banking credentials get stolen

2FA stops attacks

Attackers can initiate transactions, but no longer confirm them

Two-Factor Authentication

the reactions)

Multi-Factor Authentication

Patented in 1984

Use of multiple components for identification

- Something you **know** (password, pin-code ...)
- Something you **possess** (bank card, token, ...)
- Something you **are** (fingerprint, iris, ...)

Relies on the separation of components

An attacker needs to control all

Two-Factor Authentication

Withdraw money from an ATM

- Insert your bank card (that you **possess**)
- Enter your pin-code (that you **know**)
- Get your money

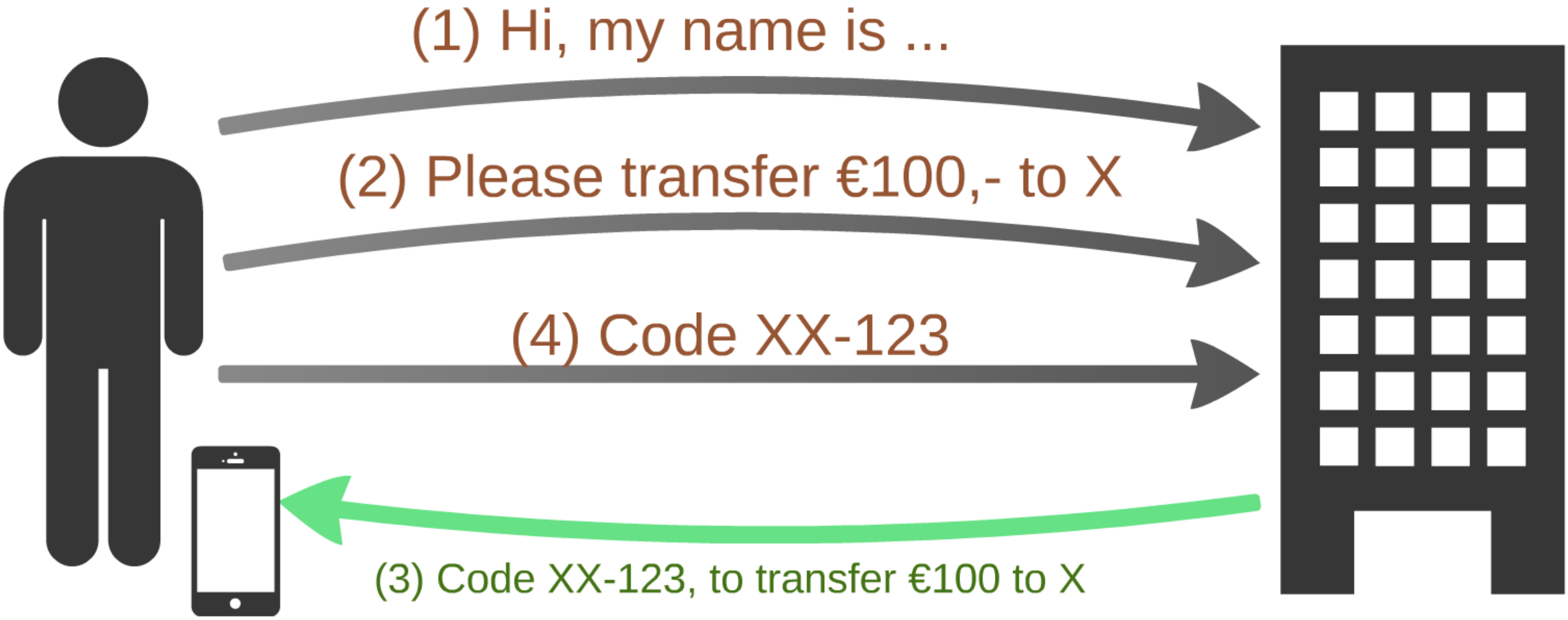
Two-factor authentication is expensive...
...so use something everybody has...

SMS

New Attack Variants

Mobile Phone Two-Factor Authentication

e-banking



(1) Hi, my name is ...

(2) Please transfer €100,- to X

(4) Code XX-123

(3) Code XX-123, to transfer €100 to X

2FA Threat-Model

(3) Code XX-123, to transfer €100 to X

2FA Threat-Model

Man-in-the-Browser (MitB)

Compromised PC

- Dridex
- SpyEye
- Carberp
- ZeuS
- ...

Banking credentials get stolen

2FA stops attacks

Attackers can initiate transactions, but no longer confirm them

SMS

New Attack Variants

1) Modify ongoing transactions:

~~Please transfer €100,- to X~~

Please transfer €100,- to Y

Mitigated by including target account information in TAN codes:

Code YY-456, to transfer €100 to **Y** (instead of to **X**)

2) Infect the 2nd Factor:

- Once the PC is in control, SPAM the mobile
- Social engineer the victim into installing malware

Malicious apps capable of forwarding SMS data

Zeus in the Mobile | SpitMo | CitMo | ...

Infecting the 2nd Factor
is not straightforward

A user must explicitly allow app installation

Google Bouncer

Detects and removes malicious apps from the Playstore

- Static Analysis
- Dynamic Analysis

Current malware relies on 'sideloading'

- *Allow app installation from unknown sources*

Let's Integrate Everything!

{ubiquitous|anywhere|pervasive} computing
The "Internet of Things"
Web 2.0?

Let's Synchronize...
- your browser
- your e-mail
- your contacts
- ...

... with your phone!

Because that is the *smart* in smartphone

The Google Way

Manage your phone from your browser

- Locate it
- Wipe it (in case of emergency)
- **Install apps!**

Permissions are shown in your browser only
No phone interaction

The Google Way

Sure this is safe

You can only remote install apps from Play.
Google Bouncer will protect you

Apps are **inactive** after installation
- A user must start them once explicitly
- Only then can we start on boot, intercept messages, ...

2. Apptivation

App is installed via remote-install

We need only **one** user interaction:
1. Open the app directly (via app-icon) (or install notification)
2. Click a custom URI (myapp://open.me)

1. Bypassing Bouncer

Already done in the past, multiple times
Bouncer *evolves* though

Assume that Bouncer can detect malicious code
Why not upload a vulnerable app instead?

A simple news app that fetches items from remote server x
We control the app code, and server x:
- exploit a memory corruption
- use known webview vulnerabilities
- ...

~~You can find us in Google Play!~~

Jekyll on iOS: When Benign Apps Become Evil (Usenix Sec '13)

Matthias Neugschwandtner, Martina Lindorfer @ LEET '13

Elevate MitB to MitMo

And intercept SMS messages

Assuming control over the browser, we need to:

1. Bypass Bouncer
2. Steer the user into activating the app

   Required only **once**

3. Intercept SMS... and profit!

3. Intercept Messages

Control over the phone

Install a SMS receiver, for each incoming SMS:
1. store it
2. detect TAN/2FA codes and delete these (pre-kitkat only)
3. webview request to our malicious server
4. download and execute a connect back (remote shell) binary

40%

Control over the browser

1. Log into e-banking environment
2. Initiate transaction
3. Confirm with intercepted TAN

~~Live~~ Demo

We are on good terms with the banks...
...so let's break something more fun instead

Google Authenticator!

Because that's not SMS based
right?

A few hours after talking to Nick - head of Android platform security - Kralevich, our app and developer account got banned...

The Fix

Google

1. Always require on-phone confirmation for app installs
2. Do not allow app activation through clicked URIs
3. Disable the remote install feature (or make it optional)
4. Perhaps look at our hiding tricks?

The user

1. Watch out for unknown app installs
2. Use a separate account for Android

The Google Authenticator user (or Azure or ...)

Use a non-android phone for your backup phone number

How Google Killed

(and t

Let's Integrate Everything!

{ubiquitous|anywhere|pervasive} computing
The "Internet of Things"
Web 2.0?

Let's Synchronize...

- your browser
- your e-mail
- your contacts
- ...

... with your phone!

Because that is the *smart* in smartphone

2. Apptivation

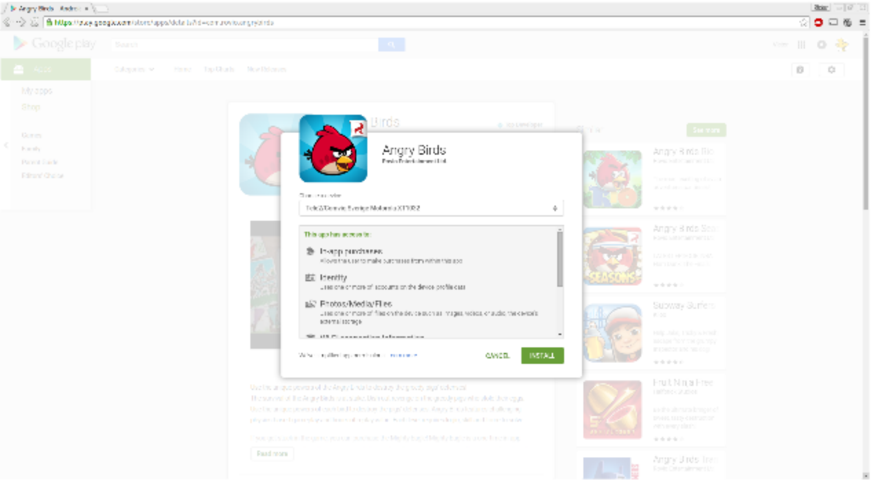
The Google Way

Manage your phone from your browser

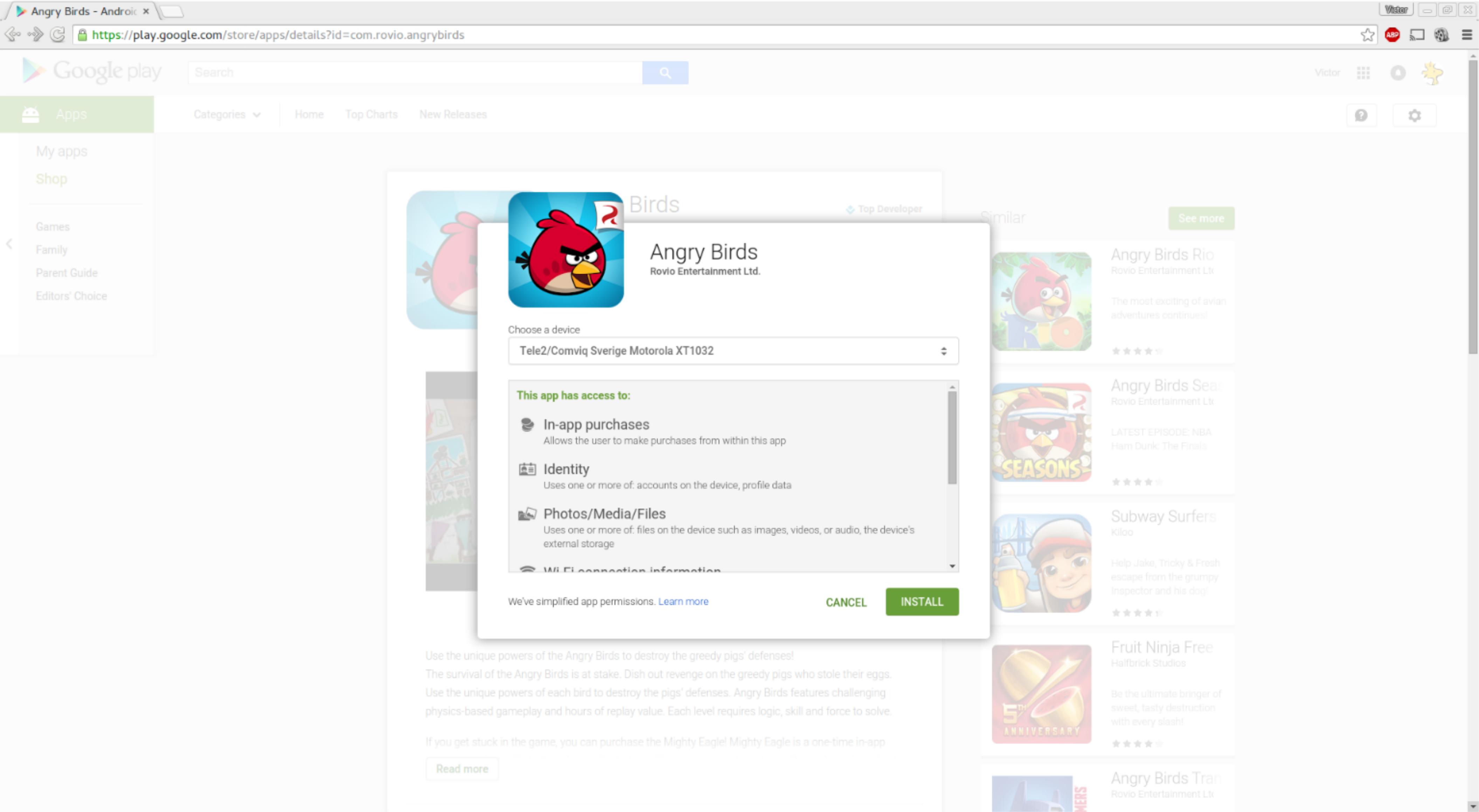
- Locate it
- Wipe it (in case of emergency)
- **Install apps!**

Permissions are shown in your browser only

No phone interaction



1. Bypassing Bouncer



Use the unique powers of the Angry Birds to destroy the greedy pigs' defenses!
The survival of the Angry Birds is at stake. Dish out revenge on the greedy pigs who stole their eggs. Use the unique powers of each bird to destroy the pigs' defenses. Angry Birds features challenging physics-based gameplay and hours of replay value. Each level requires logic, skill and force to solve.

If you get stuck in the game, you can purchase the Mighty Eagle! Mighty Eagle is a one-time in-app

Angry Birds
Rovio Entertainment Ltd.

Choose a device

Tele2/Comviq Sverige Motorola XT1032

This app has access to:

In-app purchases
Allows the user to make purchases from within this app

Identity
Uses one or more of: accounts on the device, profile data

Photos/Media/Files
Uses one or more of: files on the device such as images, videos, or audio, the device's external storage

Wi-Fi connection information

We've simplified app permissions. Learn more

The Google Way

Sure this is safe

You can only remote install apps from Play.
Google Bouncer will protect you

Apps are **inactive** after installation

- A user must start them once explicitly
- Only then can we start on boot, intercept messages, ...

Elevate MitB to MitMo

Elevate MitB to MitMo

And intercept SMS messages

Assuming control over the browser, we need to:

1. Bypass Bouncer
2. Steer the user into activating the app

   Required only **once**

3. Intercept SMS... and profit!

The Fix

1. Bypassing Bouncer

Already done in the past, multiple times

Bouncer *evolves* though

Assume that Bouncer can detect malicious code

Why not upload a vulnerable app instead?

A simple news app that fetches items from remote server x

We control the app code, and server x:

- craft a memory corruption
- use known webview vulnerabilities
- ...

Jekyll on iOS: When Benign Apps Become Evil (Usenix Sec '13)

Matthias Neugschwandtner, Martina Lindorfer @ LEET '13

~~You can find us in Google Play!~~

~~Live~~ Demo

2. Apptivation

App is installed via remote-install

We need only **one** user interaction:

1. Open the app directly (via app-icon) (or install notification)
2. Click a custom URI (myapp://open.me)

Direct open

"Hey, what is this app?"

Custom URI

Abuse synchronization

- Send a mail to self
- Replace links in Google Documents
- Post a URL to the user's Facebook wall
- Replace 'recent tabs'
- ...
- **Replace bookmarks**

1. We control the browser: replace bookmarks, retain functionality
2. Bookmarks now link to our controlled web server

3a. (old Chrome) The loaded webpage triggers an intent redirect
3b. (new Chrome) The webpage redirects after a user touch

3. Intercept Messages

custom URI (my

Direct open

"Hey, what is this app?"

Hey, what is this app?

pp://open.me)

Custom URI

Abuse synchronization

- Send a mail to self
- Replace links in Google Documents
- Post a URL to the user's Facebook wall
- Replace 'recent tabs'
- ...
- **Replace bookmarks**

1. We control the browser: replace bookmarks, retain functionality
2. Bookmarks now link to our controlled web server

3a. (old Chrome) The loaded webpage triggers an intent redirect
3b. (new Chrome) The webpage redirects after a user touch

3. Intercept Messages

Control over the phone

Install a SMS receiver, for each incoming SMS:

1. store it
2. detect TAN/2FA codes and delete these (pre-kitkat only)
3. webview request to our malicious server
4. download and execute a connect back (remote shell) binary

40%

Control over the browser

1. Log into e-banking environment
2. Initiate transaction
3. Confirm with intercepted TAN

- use known webview vulnerabilities
- ...

~~You can find us in Google Play!~~

~~Live Demo~~

We are on good terms with the banks...
...so let's break something more fun instead

Google Authenticator!

Because that's not SMS based

right?

A few hours after talking to Nick - head of Android platform security - Kralevich, our app and developer account got banned...

ng more fun instead

The Fix

Google

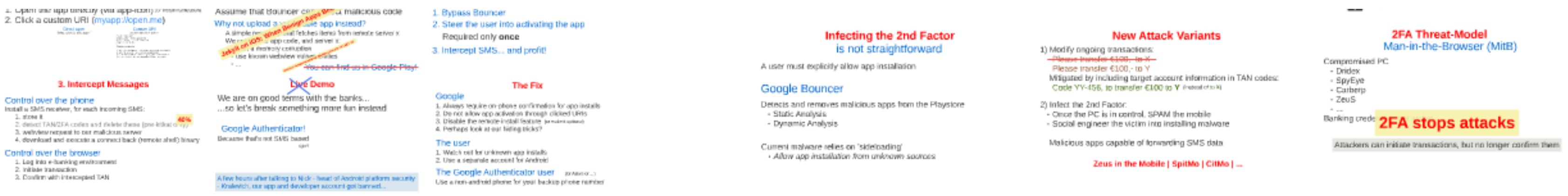
1. Always require on-phone confirmation for app installs
2. Do not allow app activation through clicked URIs
3. Disable the remote install feature (or make it optional)
4. Perhaps look at our hiding tricks?

The user

1. Watch out for unknown app installs
2. Use a separate account for Android

The Google Authenticator user (or Azure or ...)

Use a non-android phone for your backup phone number



1. Open the app directly (via app-icon)
2. Click a custom URI (myapp://open.me)

Assume that Bouncer catches malicious code

Why not upload a [illegible] app instead?

A simple [illegible] that fetches items from remote server x

We [illegible] app code, and server x:

- [illegible] a memory corruption
- use known webview [illegible]
- ...

~~You can find us in Google Play!~~

1. Bypass Bouncer
2. Steer the user into activating the app

   Required only **once**

3. Intercept SMS... and profit!

3. Intercept Messages

Control over the phone

Install a SMS receiver, for each incoming SMS:

1. store it
2. detect TAN/2FA codes and delete these (pre-kitkat only)
3. webview request to our malicious server
4. download and execute a connect back (remote shell) binary

Control over the browser

1. Log into e-banking environment
2. Initiate transaction
3. Confirm with intercepted TAN

~~Live~~ Demo

We are on good terms with the banks...
...so let's break something more fun instead

Google Authenticator!

Because that's not SMS based

A few hours after talking to Nick - head of Android platform security - Kralevich, our app and developer account got banned...

The Fix

Google

1. Always require on-phone confirmation for app installs
2. Do not allow app activation through clicked URIs
3. Disable the remote install feature
4. Perhaps look at our listing tricks?

The user

1. Watch out for unknown app installs
2. Use a separate account for Android

The Google Authenticator user

Use a non-android phone for your backup phone number

Infecting the 2nd Factor
is not straightforward

A user must explicitly allow app installation

Google Bouncer

Detects and removes malicious apps from the Playstore

- Static Analysis
- Dynamic Analysis

Current malware relies on 'sideloading'

- *Allow app installation from unknown sources*

New Attack Variants

1) Modify ongoing transactions:

~~Please transfer €100,- to X~~

Please transfer €100,- to Y

Mitigated by including target account information in TAN codes:

Code YY-456, to transfer €100 to Y

2) Infect the 2nd Factor:

- Once the PC is in control, SPAM the mobile
- Social engineer the victim into installing malware

Malicious apps capable of forwarding SMS data

Zeus in the Mobile | SpitMo | CitMo | ...

2FA Threat-Model
Man-in-the-Browser (MitB)

Compromised PC

- Dridex
- SpyEye
- Carberp
- ZeuS
- ...

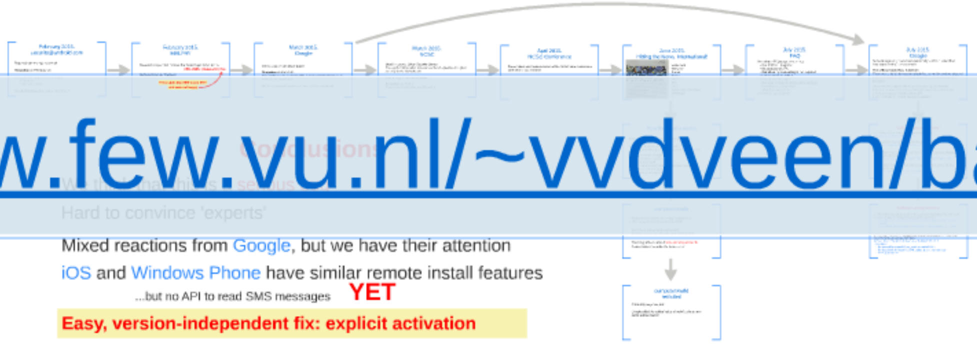
Banking crede[illegible]

2FA stops attacks

Attackers can initiate transactions, but no longer confirm them

How Google Killed Two-Factor Authentication

(and the reactions)



http://www.few.vu.nl/~vvdveen/bandroid.html

Conclusions

Hard to convince 'experts'

Mixed reactions from Google, but we have their attention

iOS and Windows Phone have similar remote install features

...but no API to read SMS messages **YET**

Easy, version-independent fix: explicit activation

Mobile-phone based 2FA seems doomed

financial institutions will come to the same conclusion

The Media is (mostly) clueless

Radhesh Krishnan
Herbert Bos

Victor van der Veen

VU University Amsterdam

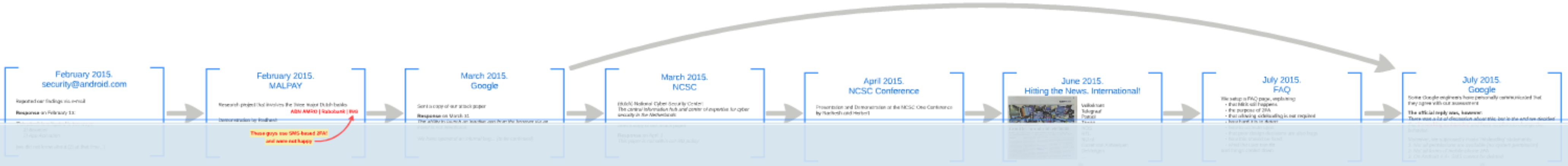
System and Network Security Group

Andrubis | TraceDroid (app analysis)

PathArmor @ CCS '15 (context-sensitive CFI)

oogle Killed Two-Factor Authe

(and the reactions)



w.few.vu.nl/~vvdveen/ba

Conclusions

We think that this is a serious bug

Hard to convince 'experts'

Mixed reactions from Google, but we have their attention

iOS and Windows Phone have similar remote install features

...but no API to read SMS messages **YET**

Easy, version-independent fix: explicit activation

Mobile-phone based 2FA seems doomed

financial institutions will come to the same conclusion

The Media is (mostly) clueless

How the media works
computerworld
This blog attracts almost 200k visitors per month
So we invited the author to have a chat
computerworld
revisited
still mildly negative, but:
Google killed the added value of mobile phone two-factor authentication
"sufficient security barriers"

Radhesh Krishnan

February 2015.
security@android.com

Reported our findings via e-mail

Response on February 13:

This attack is mitigated in two ways:

1) Bouncer

2) App Activation

(we did not know about (2) at that time...)

February 2015.
MALPAY

Research project that involves the three major Dutch banks

ABN AMRO | Rabobank | ING

Demonstration by Radhesh

These guys use SMS-based 2FA!
and were not happy

March 2015.
Google

Sent a copy of our attack paper

Response on March 31

The ability to launch an inactive app from the browser via an intent is not intentional.

We have openend an internal bug... (to be continued)

March 2015.
NCSC

(dutch) National Cyber Security Center:
The central information hub and center of expertise for cyber security in the Netherlands

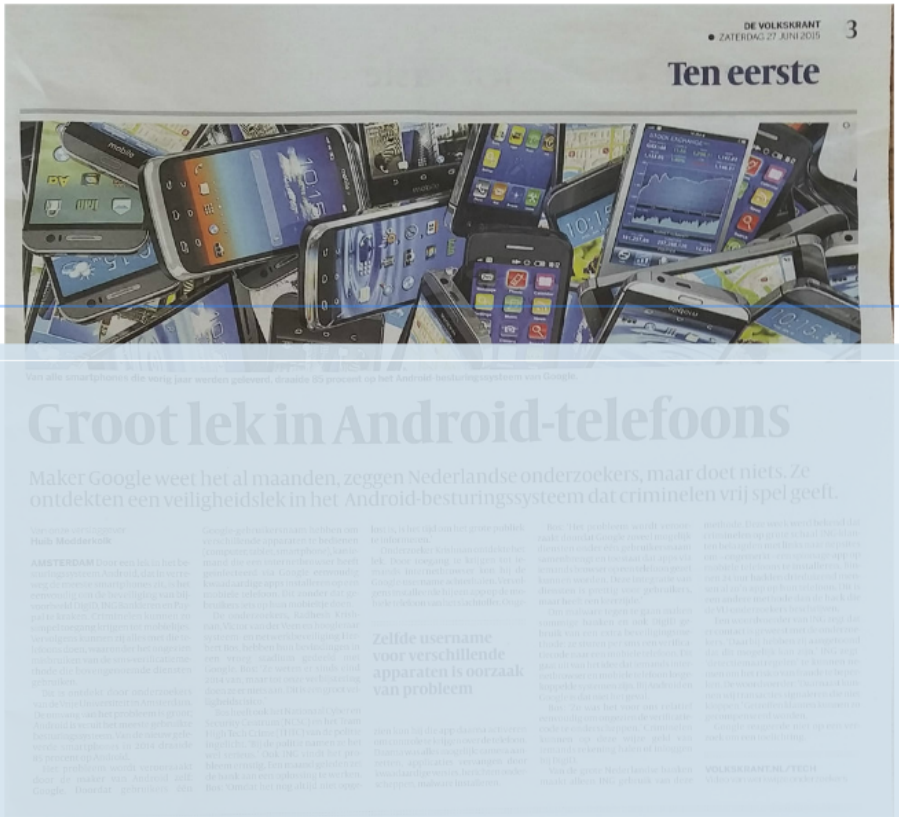
Sent a copy of our attack paper

Response on April 2
This paper is not within our RD policy

April 2015.
NCSC Conference

Presentation and Demonstration at the NCSC One Conference by Radhesh and Herbert

June 2015.
Hitting the News. International!



DE VOLKSKRANT 3
● ZATERDAG 27 JUNI 2015

Ten eerste

Van alle smartphones die vorig jaar werden geleverd, draaide 85 procent op het Android-besturingssysteem van Google.

Groot lek in Android-telefoons

Maker Google weet het al maanden, zeggen Nederlandse onderzoekers, maar doet niets. Ze ontdekten een veiligheidslek in het Android-besturingssysteem dat criminelen vrij spel geeft.

Van onze verslaggever
Huib Modderkolk

AMSTERDAM Door een lek in het besturingssysteem Android, dat in verreweg de meeste smartphones zit, is het eenvoudig om de beveiliging van bijvoorbeeld DigiD, ING Bankieren en Paypal te kraken. Criminelen kunnen zo simpel toegang krijgen tot mobieltjes. Vervolgens kunnen zij alles met die telefoons doen, waaronder het ongezien misbruiken van de sms-verificatiemethode die bovengenoemde diensten gebruiken.

Dit is ontdekt door onderzoekers van de Vrije Universiteit in Amsterdam. De omvang van het probleem is groot; Android is veruit het meest gebruikte besturingssysteem. Van de nieuw geleverde smartphones in 2014 draaide 85 procent op Android.

Het probleem wordt veroorzaakt door de maker van Android zelf: Google. Doordat gebruikers één Google-gebruikersnaam hebben om verschillende apparaten te bedienen (computer, tablet, smartphone), kan iemand die een internetbrowser heeft geïnfecteerd via Google eenvoudig kwaadaardige apps installeren op een mobiele telefoon. Dit zonder dat gebruikers iets op hun mobieltje doen.

De onderzoekers, Radhesh Krishnan, Victor van der Veen en hoogleraar systeem- en netwerkbeveiliging Herbert Bos, hebben hun bevindingen in een vroeg stadium gedeeld met Google. Bos: 'Ze weten er sinds eind 2014 van, maar tot onze verbijstering doen ze er niets aan. Dit is een groot veiligheidsrisico.'

Bos heeft ook het Nationaal Cyber en Security Centrum (NCSC) en het Team High Tech Crime (THTC) van de politie ingelicht. 'Bij de politie namen ze het wel serieus.' Ook ING vindt het probleem ernstig. Een maand geleden zei de bank aan een oplossing te werken. Bos: 'Omdat het nog altijd niet opgelost is, is het tijd om het grote publiek te informeren.'

Onderzoeker Krishnan ontdekte het lek. Door toegang te krijgen tot iemands internetbrowser kon hij de Google username achterhalen. Vervolgens installeerde hij een app op de mobiele telefoon van het slachtoffer. Ongezien kon hij die app daarna activeren om controle te krijgen over de telefoon. Daarna was alles mogelijk: camera aanzetten, applicaties vervangen door kwaadaardige versies, berichten onderscheppen, malware installeren.

Zelfde username voor verschillende apparaten is oorzaak van probleem

Bos: 'Het probleem wordt veroorzaakt doordat Google zoveel mogelijk diensten onder één gebruikersnaam samenbrengt en toestaat dat apps via iemands browser op een telefoon gezet kunnen worden. Deze integratie van diensten is prettig voor gebruikers, maar heeft een keerzijde.'

Om malware tegen te gaan maken sommige banken en ook DigiD gebruik van een extra beveiligingsmethode: ze sturen per sms een verificatiecode naar een mobiele telefoon. Dit gaat uit van het idee dat iemands internetbrowser en mobiele telefoon losgekoppelde systemen zijn. Bij Android en Google is dat niet het geval.

Bos: 'Zo was het voor ons relatief eenvoudig om ongezien de verificatiecode te onderscheppen.' Criminelen kunnen op deze wijze geld van iemands rekening halen of inloggen bij DigiD.

Van de grote Nederlandse banken maakt alleen ING gebruik van deze methode. Deze week werd bekend dat criminelen op grote schaal ING-klanten belaagden met links naar nepsites om – ongemerkt – een spionage-app op mobiele telefoons te installeren. Binnen 24 uur hadden drieduizend mensen al zo'n app op hun telefoon. Dit is een andere methode dan de hack die de VU-onderzoekers beschrijven.

Een woordvoerder van ING zegt dat er contact is geweest met de onderzoekers. 'Daarbij hebben zij aangetoond dat dit mogelijk kan zijn.' ING zegt 'detectiemaatregelen' te kunnen nemen om het risico van fraude te beperken. De woordvoerder: 'Daarnaast kunnen wij transacties signaleren die niet kloppen.' Getroffen klanten kunnen zo gecompenseerd worden.

Google reageerde niet op een verzoek om een toelichting.

VOLKSKRANT.NL/TECH
Video van [illegible] onderzoekers

Volkskrant
Telegraaf
Parool
Trouw
NOS
RTL
NU.nl
Gazet van Antwerpen
DeMorgen
...

How the edia works

Follow Mond ne 29

We erately release tails...
.. 'journalist' (i.e.,) cared to act us

I , we were slaughte
s attack is nothing n
s is oversold!
t use a strong pass and you are s
ncer will stop you!
e me back my taxe

computerworld

- *Preposterous blabla about 'big Android leak'*
- *VU researchers and media raise FUD*

FUD: Fear, Uncertainty and Doubt.
In other words: we are guilty of fearmongering

This blog attracts almost 200k visitors per month
So we invited the author to have a chat

computerworld revisited

Still mildly negative, but:

Google killed the added value of mobile phone two-factor authentication

July 2015.
FAQ

We setup a FAQ page, explaining

- that MitB still happens
- the purpose of 2FA
- that allowing sideloading is not required
- how hard it is to detect
- how to activate apps
- that poor design decisions are also bugs
- how this should be fixed
- what the user can do

and things cooled down

July 2015.
Google

Some Google engineers have personally communicated that they agree with our assessment

The official reply was, however:
There was a lot of discussion about this, but in the end we decided that it's working as intended and have no plans to change the behavior.

Moreover, we supposedly made 'misleading' statements:
1. Not all permissions are available (no system permission)
2. Not all forms of mobile-phone 2FA
3. On Android 4.4+ SMS cannot be deleted

"suffic securi rriers"

1. A victim's PC or browse be comp d
Yes, that is the threat-mo 2FA

attacker must uplo nalicious ap ay
an still download c alicious - vul

victim must av icing that the s app was uploaded
robably mean e]
only see "*ap installed*", whic even trigger activation

victim must manually activat pp or click a link that
es the app
can explo nronization/usabili es

n/b

"sufficient security barriers"

5. The victim must be using an SMS-based 2FA mechanism for their bank and not an app or hardware-token based mechanism

- Does this mean SMS-based 2FA is obsolete? Better call your bank to switch!
- The fallback for those apps (Google/Azure Authenticator) is ... SMS

6. The victim will still immediately see that an SMS-based TAN has been received and can contact their bank

Almost 40% of the Android users is still at Android < 4.4.

In addition:

- Do you call your bank if you receive a weird TAN?
- Do you check for incoming TAN codes at 3 in the morning?
- Does your mom?

t (2) at that time...)

These guys use SMS-based 2FA!
and were not happy

intent is not intentional.

We have openend an internal bug... (to be continued)

Response on April 2
This paper is not within our RD policy

.few.vu.nl/~v

Conclusions

We think that this is a serious bug

Hard to convince 'experts'

Mixed reactions from Google, but we have their attention

iOS and Windows Phone have similar remote install features

...but no API to read SMS messages **YET**

Easy, version-independent fix: explicit activation

Mobile-phone based 2FA seems doomed

financial institutions will come to the same conclusion

The Media is (mostly) clueless

1. Open the app directly (via app-icon)
2. Click a custom URI (myapp://open.me)

3. Intercept Messages

Control over the phone

Install a SMS receiver, for each incoming SMS:

1. store it
2. detect TAN/2FA codes and delete these (pre-kitkat only) 46%
3. webview request to our malicious server
4. download and execute a connect back (remote shell) binary

Control over the browser

1. Log into e-banking environment
2. Initiate transaction
3. Confirm with intercepted TAN

Assume that Bouncer catches malicious code

Why not upload a [illegible] app instead?

A simple [illegible] that fetches items from remote server x

We [illegible] app code, and server x:

- [illegible] a memory corruption
- use known webview vulnerabilities
- ...

Jekyll on iOS: When Benign Apps Become Evil

~~You can find us in Google Play!~~

~~Live~~ Demo

We are on good terms with the banks...
...so let's break something more fun instead

Google Authenticator!
Because that's not SMS based

A few hours after talking to Nick - head of Android platform security - Kralevich, our app and developer account got banned...

1. Bypass Bouncer
2. Steer the user into activating the app
   Required only **once**
3. Intercept SMS... and profit!

The Fix

Google

1. Always require on-phone confirmation for app installs
2. Do not allow app activation through clicked URIs
3. Disable the remote install feature (or make it optional)
4. Perhaps look at our listing tricks?

The user

1. Watch out for unknown app installs
2. Use a separate account for Android

The Google Authenticator user (or Authy or ...)

Use a non-android phone for your backup phone number

Infecting the 2nd Factor is not straightforward

A user must explicitly allow app installation

Google Bouncer

Detects and removes malicious apps from the Playstore

- Static Analysis
- Dynamic Analysis

Current malware relies on 'sideloading'

- *Allow app installation from unknown sources*

New Attack Variants

1) Modify ongoing transactions:

~~Please transfer €100,- to X~~
Please transfer €100,- to Y
Mitigated by including target account information in TAN codes:
Code YY-456, to transfer €100 to Y (instead of to X)

2) Infect the 2nd Factor:

- Once the PC is in control, SPAM the mobile
- Social engineer the victim into installing malware

Malicious apps capable of forwarding SMS data

Zeus in the Mobile | SpitMo | CitMo | ...

2FA Threat-Model
Man-in-the-Browser (MitB)

Compromised PC

- Dridex
- SpyEye
- Carberp
- ZeuS
- ...

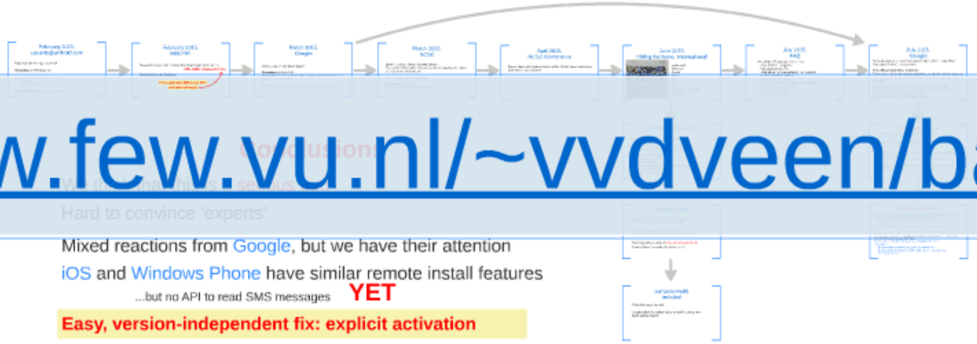
Banking crede[illegible]

2FA stops attacks

Attackers can initiate transactions, but no longer confirm them

How Google Killed Two-Factor Authentication

(and the reactions)



http://www.few.vu.nl/~vvdveen/bandroid.html

Conclusions

We think that this is a serious ...

Hard to convince 'experts'

Mixed reactions from Google, but we have their attention

iOS and Windows Phone have similar remote install features
...but no API to read SMS messages **YET**

Easy, version-independent fix: explicit activation

Mobile-phone based 2FA seems doomed
financial institutions will come to the same conclusion

The Media is (mostly) clueless

Radhesh Krishnan
Herbert Bos

Victor van der Veen

VU University Amsterdam

System and Network Security Group

Andrubis | TraceDroid (app analysis)
PathArmor @ CCS '15 (context-sensitive CFI)