Berlin Institute of Technology
FG Security in Telecommunications

Hacking NFC and NDEF: why I go and look at it again

NinjaCon / B-Sides Vienna 2011

Collin Mulliner, June 18th 2011, Vienna, Austria
collin@sec.t-labs.tu-berlin.de

History of this Talk :-)

- Started looking at NFC phone security in 2007/2008
- Found a bunch of bugs in THE NFC phone of that time
- Traveled to **Vienna** to look at NFC services in the field
- *2008 – 2010 (end of) NFC looked totally dead to me*
- NFC got interesting again (Google Nexus S)
- NinjaCon CFP … oh right this was in **Vienna**
- Here I am!

About me

- Collin Mulliner
  - PhD student at Technische Universität Berlin and T-Labs
- Group: Security in Telecommunications (SecT)
- Reseach area:
  - Mobile and smartphone security ;-)
- Contact:

  http://www.mulliner.org/blog/

  http://www.mulliner.org/collin/academic/

  Twitter: @collinrm

Agenda

- Introduction
- NFC phones
- My NFC/NDEF Security Tools (some new stuff)
- Nokia NFC phones
- Android/Google Nexus S (new stuff)
- Analysis of Field Test NFC Services (Vienna)
- Notes from the lab
- Conclusions

NFC just become popular!

- I looked at NFC in 2007/2008
  - Just research stuff then
  - Nokia S40 based

- Now 2010/2011:
  - VISA has NFC-based payment
    - iPhone hardware add-on
  - Android phones with NFC: Nexus S
    - Soon other (Samsung) Android phones
  - Google Wallet (NFC-based payment)
    - Not rolled out yet!

Near Field Communication (NFC)

- Bidirectional proximity coupling technology
  - Based on ISO 14443
- NFC device modes
  - RFID Reader/Writer
    - Proximity Coupling Device (PCD)
  - Card Emulation
    - Proximity Inductive Coupling Card (PICC)
  - NFCIP the Peer-to-Peer mode (ISO 18092)
    - Bidirectional communication between NFC devices
- RFID in your phone

NFC Tech

- Frequency: 13.56 Mhz
- Communication range: ~4cm
- Data transfer rate: 106, 216, 412 kbit/s
- Supported tags (by the standard):
  - ISO 14443 A/B
  - NXP Mifare Ultralight, Classic/Standard 1k/4k, DESFire
  - Sony FeliCa
  - Innovision Topaz, Jewel tag

NFC “general” Security

- No link level security (wireless not encrypted)
  - Eavesdropping (sniffing)
  - Man-in-the-middle
  - Data: Modification, Corruption, Insertion [8]

- Tamper with NFC/RFID tags
  - Modify original tag
  - Replace with malicious tag
  - … sounds easier than it is, more on this later ...

NFC Usage Concept

- Touch tag with your mobile phone
  - Phone reads tag ⇨ performs action
    (this is the stuff that is used right now!)

NFC Usage Concept cont.

- The NFC P2P mode (NFCID)
  - Only used for games and/or file transfer :-(

- NFC card emulation
  - Should be **the big thing** for NFC after all NFC is build for payment
  - Haven't seen real stuff using this

- Special stuff...
  - The “VISA” iPhone NFC adapter should be deployed somewhere...

NFC Data Exchange Format (NDEF)

- Container format to store NFC-data in RFID tags
  - Independent from tag type (mostly)
- Defines a number of NFC specific types
  - URI, TextRecord, SmartPoster, …
- Standardized by the NFC Forum [2]
  - Specs are public
  - Available for free

NDEF Record and Message

- The Record is the smallest entity in NDEF
  - Each Record carries a Type
  - Multiple Records from an *NDEF Message*

- Most important NDEF Types
  - URI Record (the thing that will trigger some action)
    - HTTP, TEL, SMS, …
  - Text Record
    - String of characters ;-)
    - Includes language identifier

The NFC SmartPoster

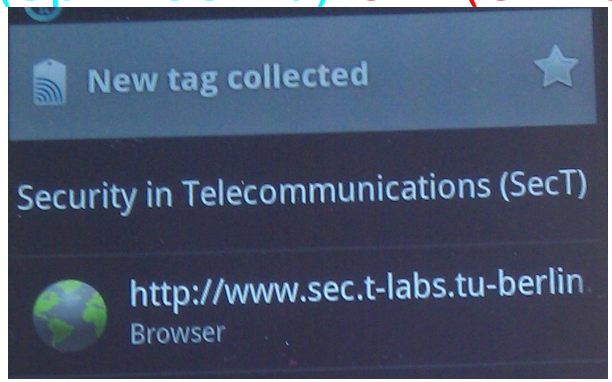
- URI with a title!
  - Title is a “descriptive” text
  - And an optional icon (not implemented anywhere!)
- Defines additional subtypes
  - Recommended action 'act' (what to do with the URI)
    - Execute, save, edit
    - Not implemented anywhere
  - Size and type of object the URI points to
- **This is one of the proclaimed key use cases of NFC!**

SmartPoster: example

Title: Security in Telecommunications (SecT)
URI : http://www.sec.t-labs.tu-berlin.de

```
03 58 D1 02 53 53 70 91 01 23 55 00 68 74 74 70
3A 2F 2F 77 77 77 2E 73 65 63 2E 74 2D 6C 61 62
73 2E 74 75 2D 62 65 72 6C 69 6E 2E 64 65 51 01
28 54 02 65 6E 53 65 63 75 72 69 74 79 20 69 6E
20 54 65 6C 65 63 6F 6D 6D 75 6E 69 63 61 74 69
6F 6E 73 20 28 53 65 63 54 29 FE
```

NDEF SmartPoster (Sp = 53 70) URI (U = 55) Text (T = 54)



NFC Mobile Phones

- Phones you can buy:
  - Nexus S (smartphone, Android)
  - Nokia 6212 classic (feature phone, S40)
  - Nokia 6131 NFC (feature phone, S40)

Quick Spec Nokia S40 NFC Phones

- GSM/UMTS feature phone with Bluetooth, GPRS, MicroSD, camera, J2ME/MIDP2.0 and of course NFC
- Interesting JSRs: 87 (Bluetooth), 257 (NFC)
- NFC support for:
  - SmartPoster, URI, Tel, SMS, vCal, vCard
  - Some Nokia extensions
  - Tags:
    - ISO 14443 A, NXP Mifare, Sony FeliCa (non secure parts only), Topaz and Jewel tag (read only)
- The 6212 classic supports “file transfer” … more later

Quick Spec Nexus S

- Android 2.3.x phone from Google (build by Samsung)
- Out of the box only NFC/RFID reading
  - HW supports writing many 3rd party apps support writing
- NFC hw: NXP PN65N
  → PN544 + secureMX
- NFC antenna in battery cover
  - red arrow
- More details at [19]
- Android NFC API [16]

Inside an NFC Phone

- Reader is active if phone is “active”
  - active = screen unlocked
- If NFC aware app is running
  - App handles interaction with NFC hardware
- If non-NFC app is running the phone OS takes care and …
  - Reads tags automatically if in range
  - Tag data (NDEF data) is parsed
  - If “known” data is found it is pushed to “registered” app
    - URI tags containing HTTP URLs are handed to the browser (not fully automated)

Attacking NFC Phones

- Nokia 6131 NFC
  - very old
- Nokia 6212 Classic
  - old
- Google Nexus S
  - new!

Attack Targets

- The mobile phone / smartphone
  - Crash system and/or app
  - Hijack phone (install malicious app)
  - Application bugs and design issues (fraud!)

- The Services / Applications
  - Attack the service tags and back-end infrastructure
  - Mostly designed to protect service provider not customer

The Mifare Classic Tag

- Very common 13.56 Mhz RFID tag type
  - Used by all NFC services I've seen so far
- Two tag types
  - Mifare 1k ⇨ 720 bytes payload
  - Mifare 4k ⇨ 3408 bytes payload
- Per sector configurable R/W mode
  - Two 48bit keys control read and write access

My NFC Security Toolkit

- Tag reader/writer
  - Stationary and mobile (for field analysis)
- NDEF parsing and construction library
  - Analyze tag data collected in the field
  - Test NFC mobile phones (fuzzing)
- Tag security tester
  - Check read/write mode of tags in the field

Tag Reading/Writing/Dumping (ndef_mifare)

- Librfid-based tool for USB RFID reader/writer
  - Read, write, dump, format (NDEF), and wipe tags
  - ⇨ ndef_mifare.c (v0.3 released at NinjaCon)

- MIDP2.0/JSR-257 and Nokia extensions-based
  - Bluetooth interface for control by PDA/laptop
  - Raw dump of Mifare Classic tags
  - ⇨ BtNfcAdapter and BtNfcAdapterRAW(.jar)

- All tools available in source under GPLv2 [1]

ndef_mifare v0.3

```
root@bsod:/home/collin/projects/nfc/NFC/software/librfid/new/librfid/utils# ./nd
ef_mifare -h
opening reader handle OpenPCD, CM5x21
No OpenPCD found
ndef_mifare v0.3: Copyright Collin Mulliner http://www.mulliner.org/nfc/
 License: GPLv2
syntax: ndef_mifare <options> [file]

 -h     --help         Print this help message
 -r     --read [file]  Read a mifare card/tag
 -w     --write [file] Write a mifare card/tag
 -4     --mf4k         R/W Mifare 4K card/tag
 -m     --ign4kmad     Ignore second MAD in Mifare 4k
 -B     --key-b        Use B key
 -f     --format [sec] NDEF format card/tag, start at sector: sec
 -c     --clear        Clear/wipe data area
 -d     --dump [file]  Dump entire tag (inc. trailers)
```

Python NDEF Library

- Construct and parse
  - NDEF Records and Messages
  - High-level NDEF Records: Text, URI
  - High-level Messages: SmartPoster
  - Nokia Bluetooth Imaging Tag (non standard)
  - RMV ConTag (application specific)

- *Fuzzing ready ;-)*
  - Set field length independent from field content

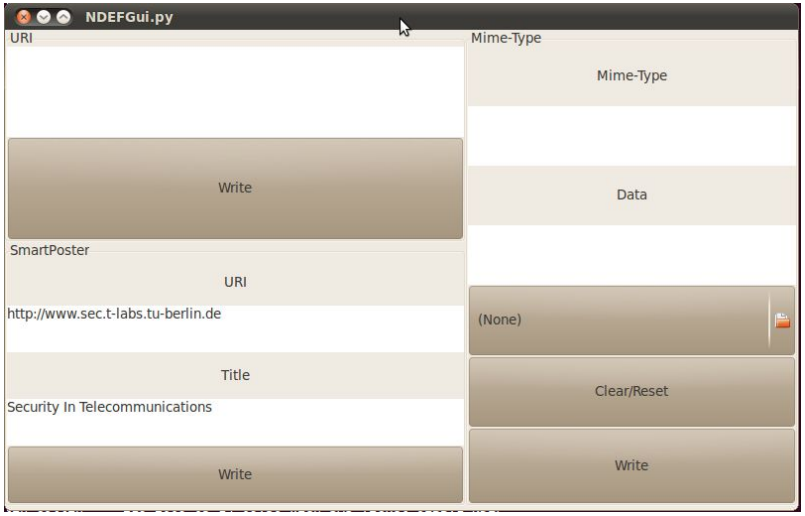
- Slightly updated version here at NinjaCon

Python NDEF Library cont.

- All functions accept an NDEF Message or NDEF Record in binary or hex as input
- Both binary and hex are supported as output
- Output easily writable with any RFID writer
- No library dependencies
  - Works really great on my Linux tablet
- Available in source under GPLv2 [1]

NDEF Quick Tools (some parts new!)

- NDEFGuiF.py
  - GUI app to build Uri, SmartPoster, and MIME tags
  - Writes output to file to be written using *ndef_mifare*



- QuickSP.py
  - Quick SmartPoster generator, write using *ndef_mifare*

```
quickSP.py /tmp/demo1.mf http://www.mulliner.org MUlliNER.ORG
```

Mifare Sector Trailer Tool

- Field tool to analyze R/W state of **Mifare** tags
  - Inspect individual sector trailer
  - Write individual or all sector trailer(s)
    - Set R/W mode and keys
  - Brute force and ”word list” *crack* sector key
    - Check for weak keys; speed ~10keys/s
    - (Proof-of-concept, very unlikely to break anything real!)
- Available in source under GPLv2 [1]
  - ⇨ MfStt(.jar)
- *Some features are available in the Android app: NFC Tag Info*

Nokia S40 - NFC Phone Analysis

- What parts of the standard are un-/supported
  - SmartPoster action 'act' is ignored :-(
  - Implementation issues?

- What about the components that are controllable by NFC?
  - Web browser just fetches anything pointed to by URL

Nokia 6131 NFC URI Spoofing

- Abuse SmartPoster to hide real URI
  - GUI mixes informational text and control data
  - ⇨ Trick user into performing harmful operation

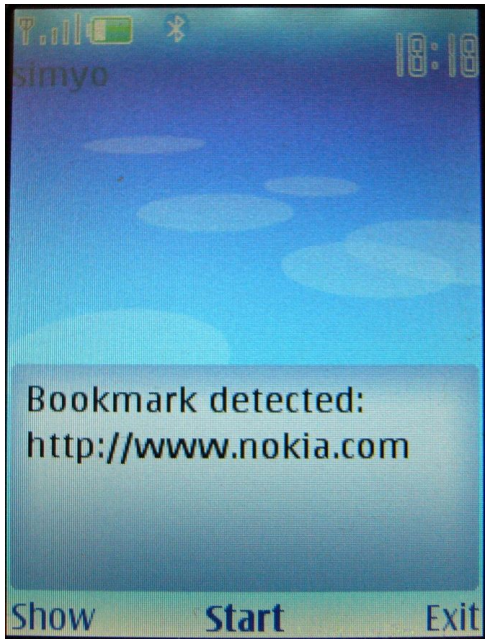
- Vulnerable components
  - Web browser (http, https, ftp, ...)
  - Phone dialer (initiate phone call)
  - Short Messaging (send SMS)

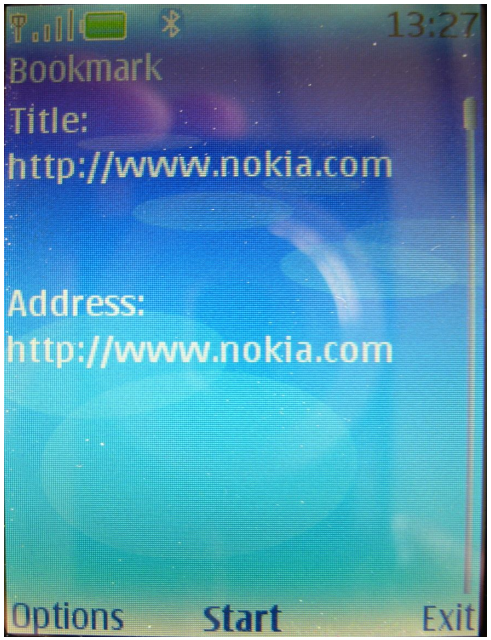
SmartPoster URL Spoofing

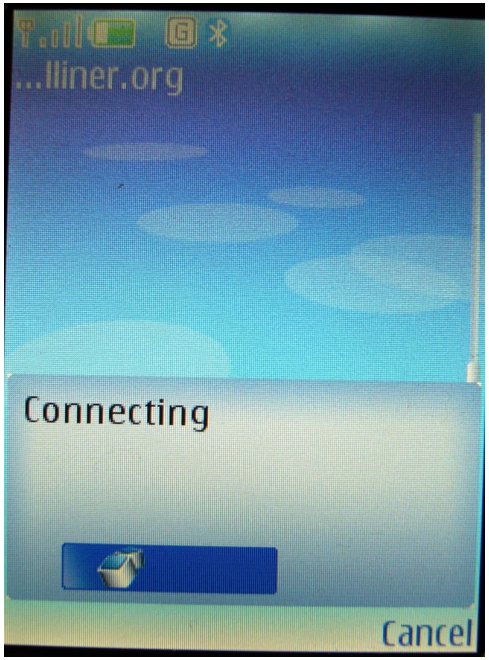
- Fake innocent locking URL stored in SmartPoster title
  - Actual URL is stored in URI record
- User can't easily determine the real URL he is going to load after reading an NDEF tag
- Title needs padding in order to hide real URI
  - Pad with either space or \r
  - End with a . (dot) in order to show the padding

Web Browser Example

- URI is “http://mulliner.org/blog/”
  - Title is:
    “http://www.nokia.com\r\r\rAddress:\rhttp://www.nokia.com\r**...**\r.”







Survives brief inspection by user.

Man-in-the-middle Proxy

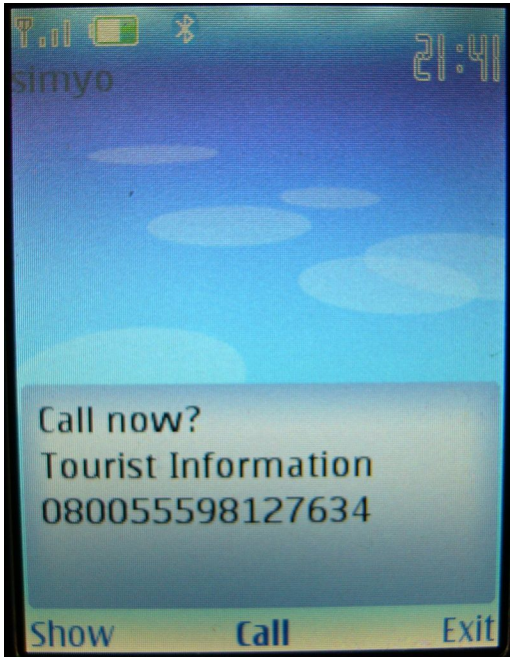
- Based on CGIProxy2.1 by James Marshall
  - Added WML handling and traffic logging
- Steal credentials (phishing...)
- Inject malicious content
- Works because:
  - Current URL is not displayed by web browser

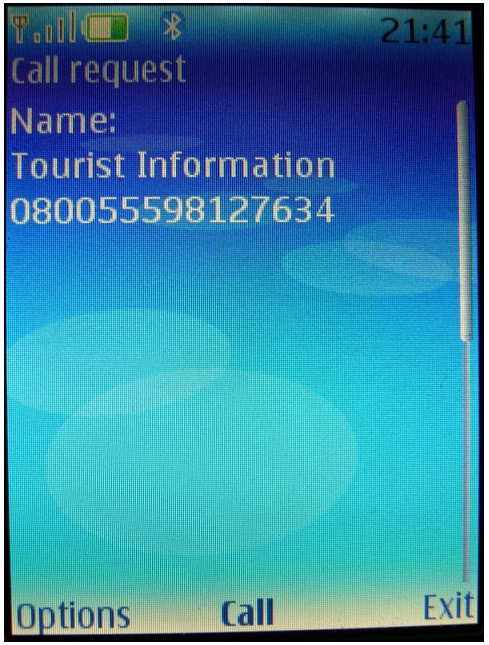
Example:

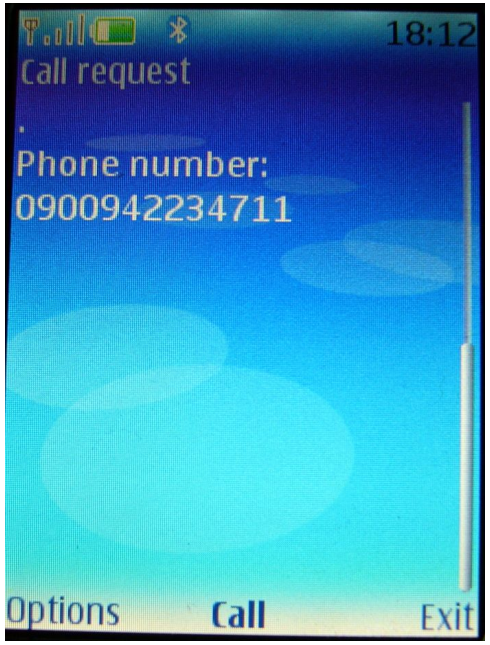
Title: https://mshop.store.com/
URI: http://attacker.com/proxy.cgi/https/mshop.store.com/

Phone Call Request Example

- URI is “tel:0900942234711”
  - Title is: “Tourist Information\r080055598127634\r\r\r\r\r\r\r.”



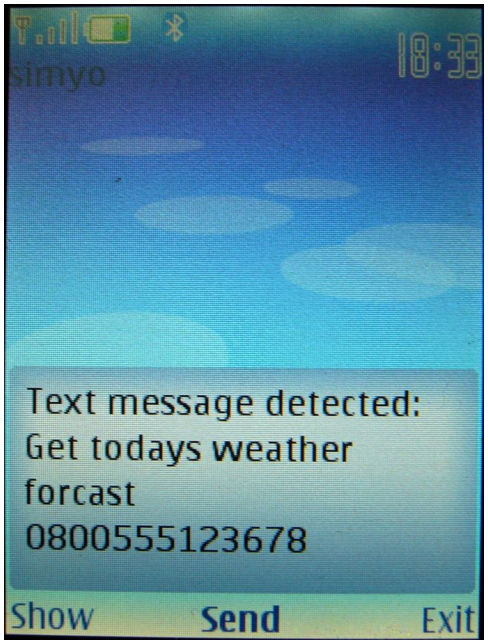


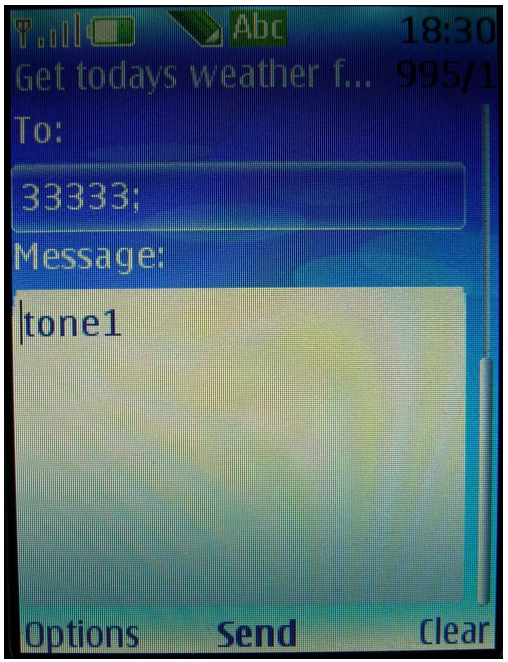


Survives brief inspection by user.

SMS Example

- URI “sms:33333?body=tone1”
  - Title is:"Get todays weather forecast\r0800555123678"





Attack from the Spec?

- Page 6: Smart Poster Record Type Definition (SPR 1.1) SmartPoster_RTD_1.0_2006-07-24

3.3.2 The Title Record

The Title record is an instance of a Text RTD Record [TEXT]. There MAY be an arbitrary number of title records in the Smart Poster. However, there MUST NOT be two or more records with the same language identifier.

The Title record SHOULD be shown to the user.

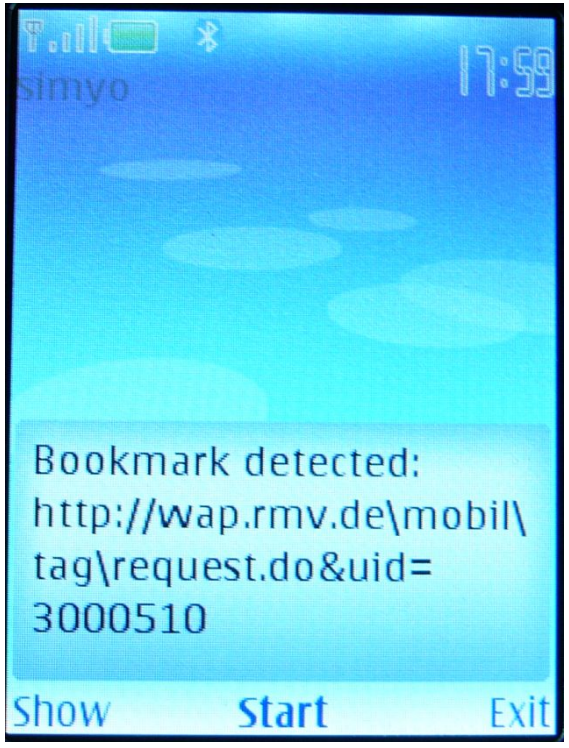
NOTE TO IMPLEMENTERS: The implementer should be aware of the fact that by putting malicious information to the Title record and thus misrepresenting the service, it might be possible to fool the user into thinking that the tag contents might be something else entirely. This is a so-called *phishing* technique. For example, if the Title record contains the text "http://www.internetbanking.com", and the URI record the text "http://myevilsite.com", the user might be fooled into giving his banking information, if the Title record is the only one that is shown to the user.

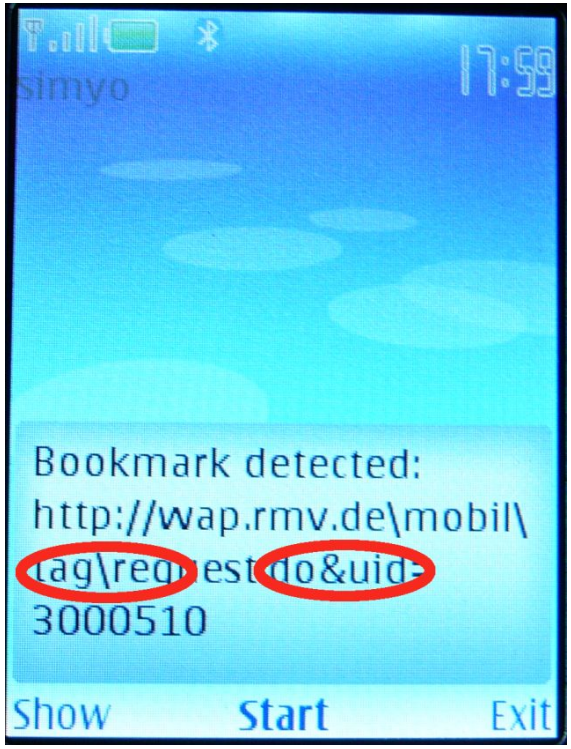
More on URL Spoofing

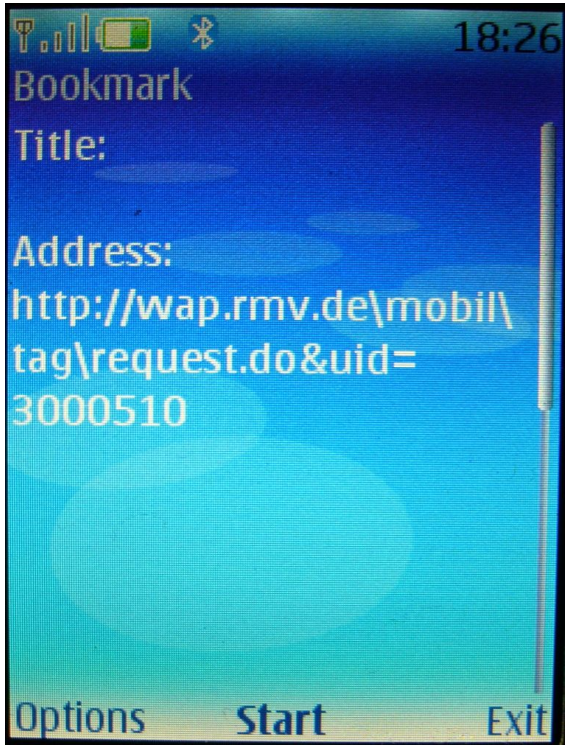
- Use classic @ method
  - Produces broken HTTP request but will work with a small redirector (HTTP 300 + new location)
  - Certain characters are not allowed in part before @
  - See *badproxy.py* example [1]

- Web browser display issue with long hostname
  - Partial hostname ⇨ user more easily fooled into loading malicious website

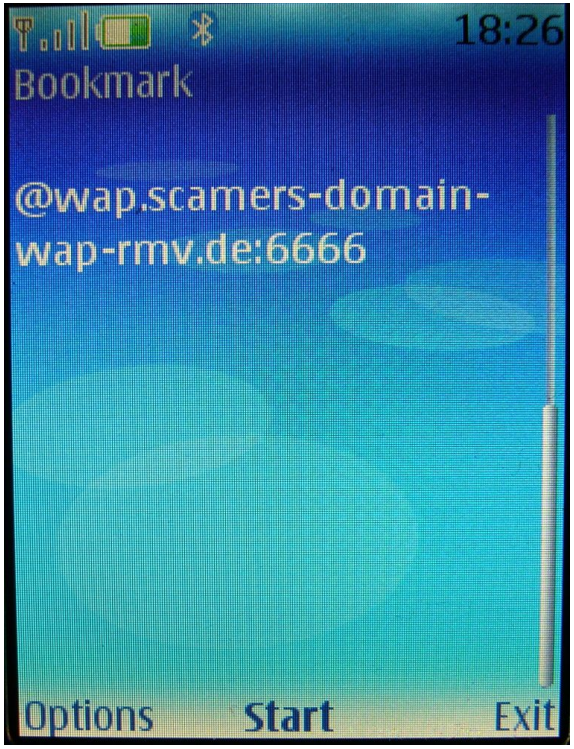
More on URL Spoofing cont.

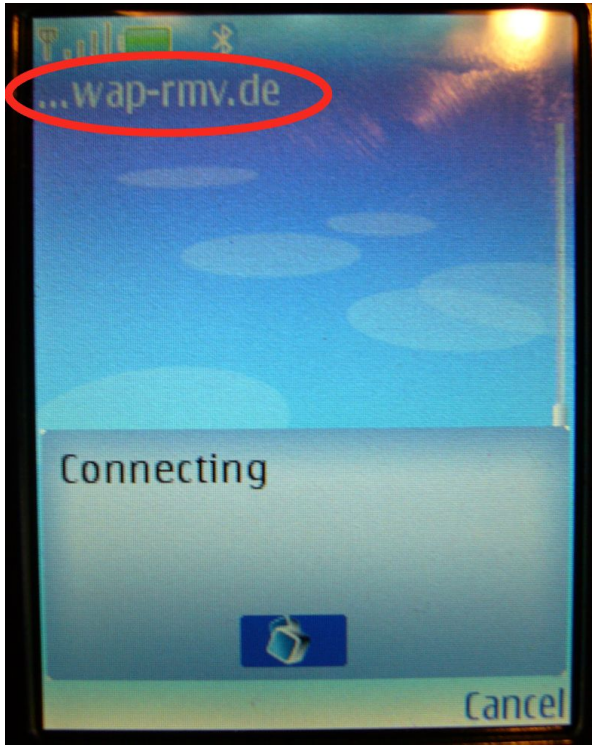






Partial Hostnames





Vendor Contacted (Nokia)

- Issues reported to Nokia in late March 2008
  - Very fast response
- Constant contact to Nokia since then
  - Added some more issues over time
- Nokia seems to take issues seriously!
  - Apparently they started fixing the bugs right away

Proof-of-Concept NDEF Worm (Nokia S40)

- Idea I had while playing with the push registry
  - Push registry allows registration for URI Record 'U'

- Basic idea: writable tags as transport for Worm
  - Use URI spoofing to hide the worm-install-URL
  - Silent MIDlet installation
    - No security warning when downloading a JAR file!
    - Auto install – user will only be asked before execution!
  - Spreads by writing URL pointing to itself to tag
  - Worm is activated by phone reading plain URI tag

- For full details see my old slides at: http://mulliner.org/nfc/

NDEF Fuzzing (Nokia S40)

- Quick sweep, just wanted to try it
- Setup
  - My NDEF library and NDEF writer tool
  - RFID reader/writer (I used a USB CardMan 5321)
  - Mifare 1k/4k tags
- Target: Nokia 6131 NFC
  - V05.12, 19-09-07, RM-216

Fuzzing Results

- NDEF Record
  - Payload length field (0xFFFFFFFF) crashes phone
- NDEF URI 'U' (well known type = 0x01)
  - “Tel:”<exactly 124 numbers> crashes phone
    - Shorter no. is accepted, longer no. produces an error
    - Best guess: off-by-one
  - Same result with “SMS:”
    - Same “phone” application handles both URIs?

Fuzzing Results cont.

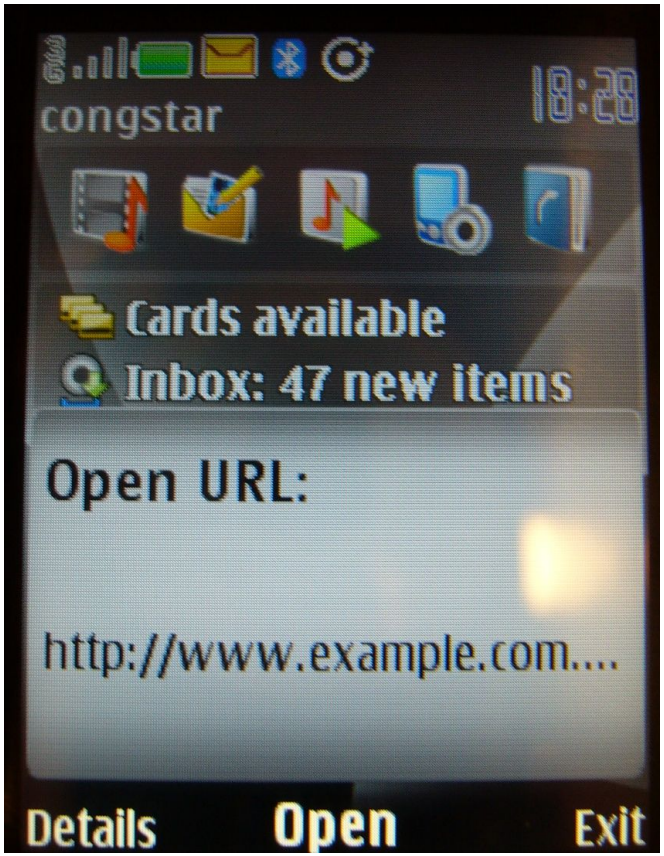
- Fuzzing using tags is hard work
  - Tag: on writer, to phone and back (no automation)
- Phone switches off after 4 crashes in a row
  - This is the S40 watchdog, nice thing that helps switching off phones
- Symbian Series 40 not very interesting
  - No known code injection technique
- This will be interesting for other phone OSes
  - Code injection via RFID/NFC anyone?

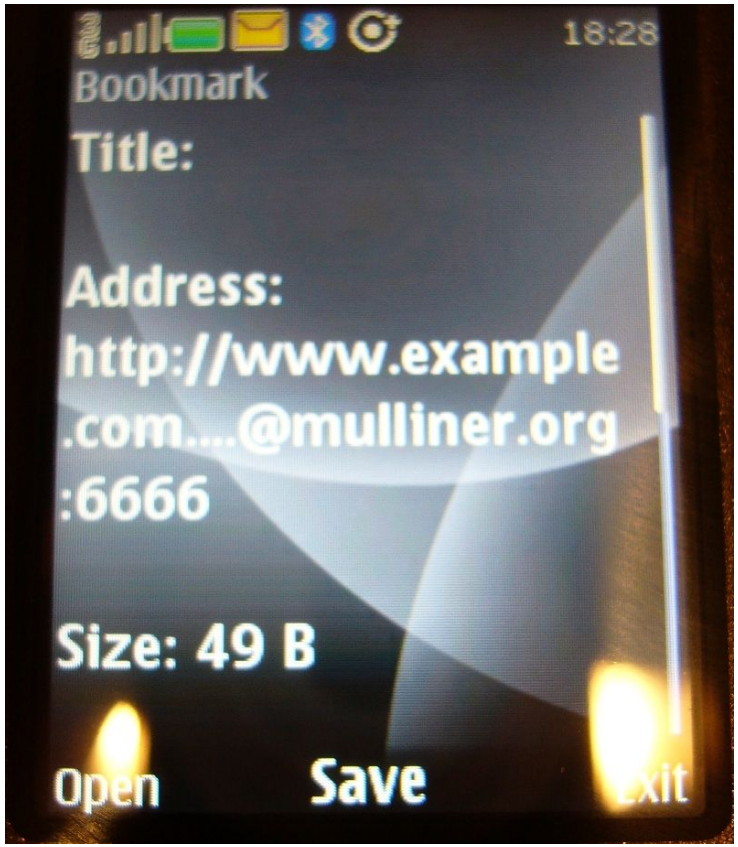
Nokia 6212 Classic

- Not vulnerable to most of the bugs I found in the Nokia 6131
- URL spoofing still possible
  - Space for URL display very limited, overlapping characters are replaced with “...”
  - Use good old @-trick
- Browser doesn't display URL or hostname
- Shows warning about unsigned MIDlets

Nokia 6212 Classic URL Spoofing

URI: http://www.exmaple.com.....@mulliner.org:6666 (broken http request so point to redirect proxy)





Paper:
Practical attacks on NFC enabled cell phones

- Authors: Roel Verdult and Francois Kooman
- Attack against the Nokia 6212 classic's file transfer feature
- Attack based on NFCIP (P2P mode) in combination with Bluetooth
- Allows to install J2ME/MIDP application without user's knowledge
- Paper published Feb. 2011 and is available at [17]

The Nexus S

- Not too many NFC apps / services yet
  - Google Wallet only announced
- This is what I did so far...

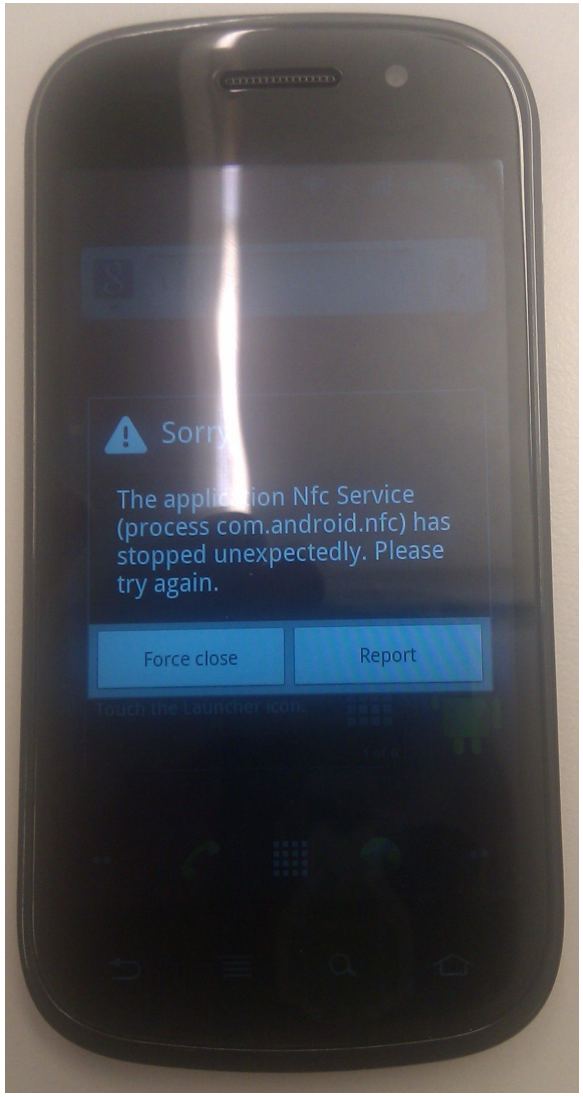
  1) NDEF fuzzing

  2) Auto launch URI fun

  3) Investigated SmartPosters
  - Only basics are implemented, again not 'act' :-(
  - URL spoofing?

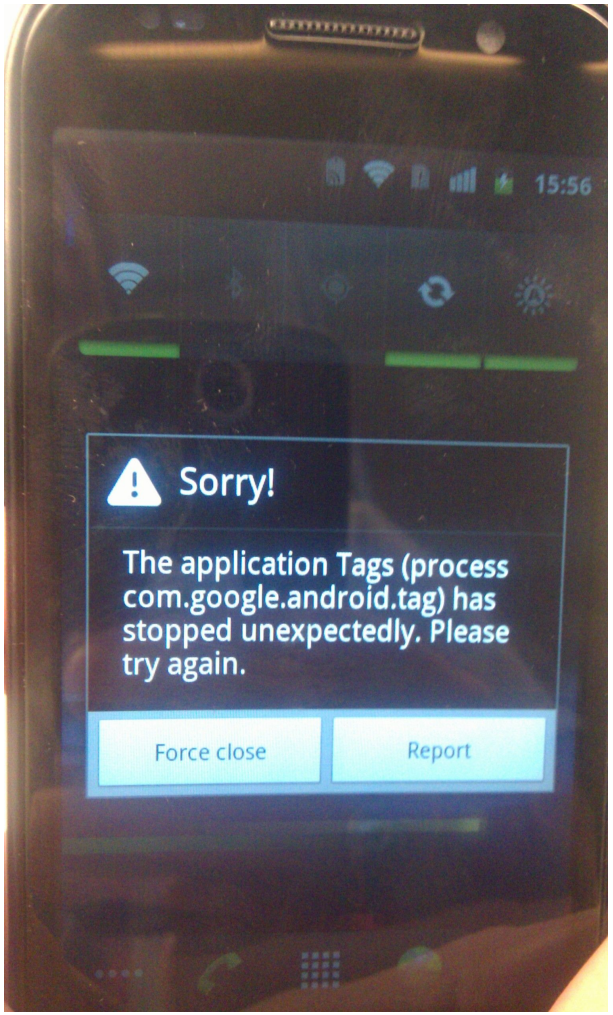
Nexus S

- I only took a brief look...
- NFC Tag TLV bug (from Nokia S40)
  - 0xFFFFFFFF as tag data length
  - Also crashes the Nexus S
    - Crashes the **com.android.nfc** service, low level NFC daemon
  - First thing I tried on the Nexus S :)
  - Nothing serious
    - Bad user experience



Nexus S cont.

- Playing with NDEF …
- Simple bug in NDEF parsing code...
  - Crashes the tag reader app **com.google.tag**
  - Record
    - length 0x0F
    - content “none”
  - Nothing serious
    - Bad user experience



Nexus S Bugs ... reported to Google

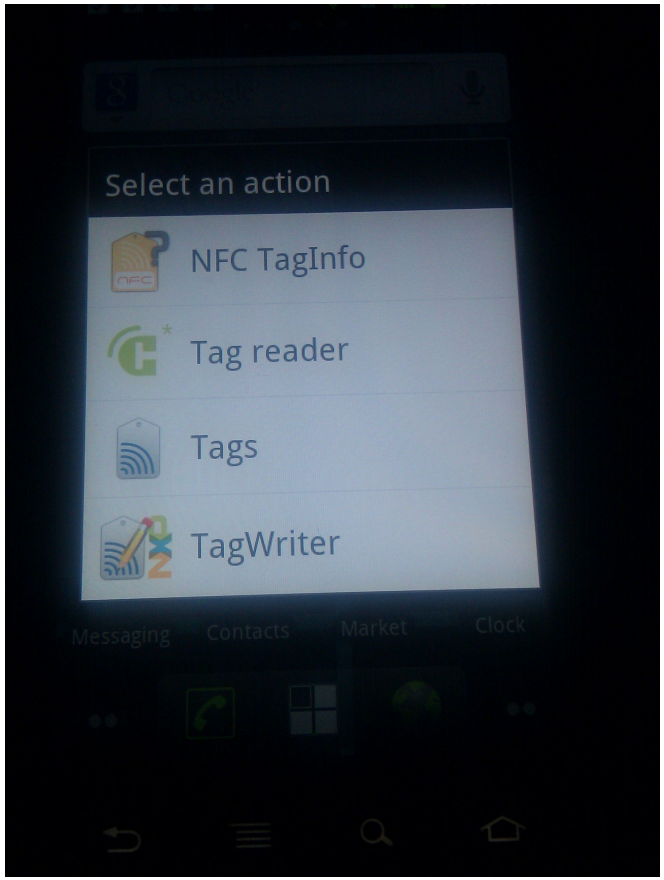
- Both should be fixed in the next release

3rd Party NFC Apps from the Market

- NXP TagWriter crashes (NDEF record parsing bug)
  - Sometimes crashes in a way that disables NFC on the Nexus S until a **other NFC app is started manually**
- NFC Tag Writer & Reader crashes (NDEF record parsing bug)
- There are tones and tones of other apps...
  - Go and try for yourself

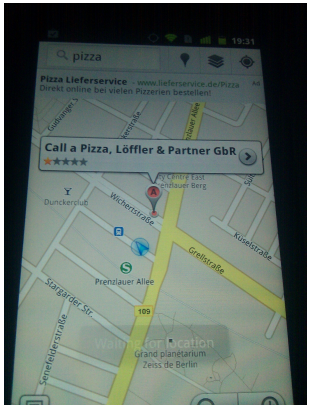
Multiple NFC Apps on Android

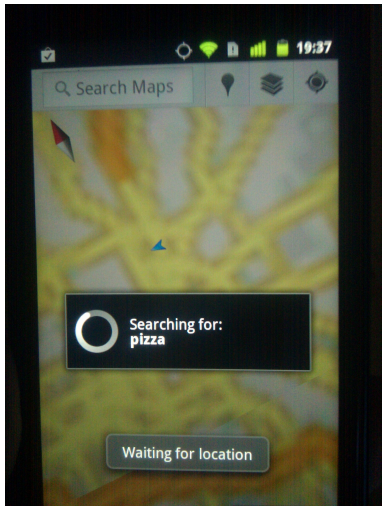
- URIs with “unknown” content get handled by the default app
- Multiple “default” handlers will cause user dialog



Potential fun with Android NFC/NDEF

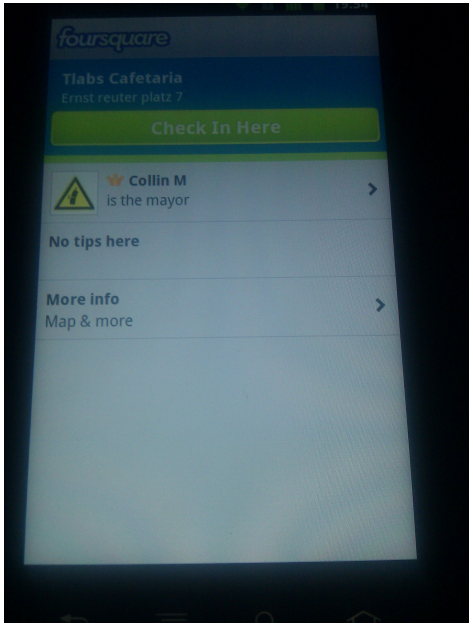
- **Automatic action on tag content (auto launch)**
- All URLs that contain **http://maps.google.TLD** are opened automatically in maps
- My pizza tag: http://maps.google.de/maps/place?q=pizza
  - Opens maps and searches for **pizza** at current location
- Also works with driving directions...
  - Copy & Past your maps URL to an NDEF tag





Foursquare ...

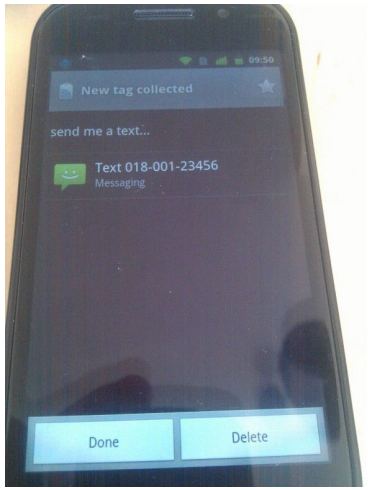
- has automatic URL handler... (taken from AndroidManifest.xml)
- http://m.foursquare.com/[user/venue/shout/checkin/checkins]
- Foursquare tag for your “venue”
  - http://m.foursquare.com/venue/*VenuID*
  - My favorite:
    http://m.foursquare.com/3610408
    - The “T-Labs Cafeteria” :-)

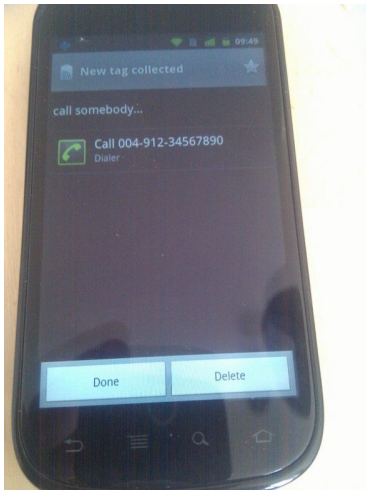


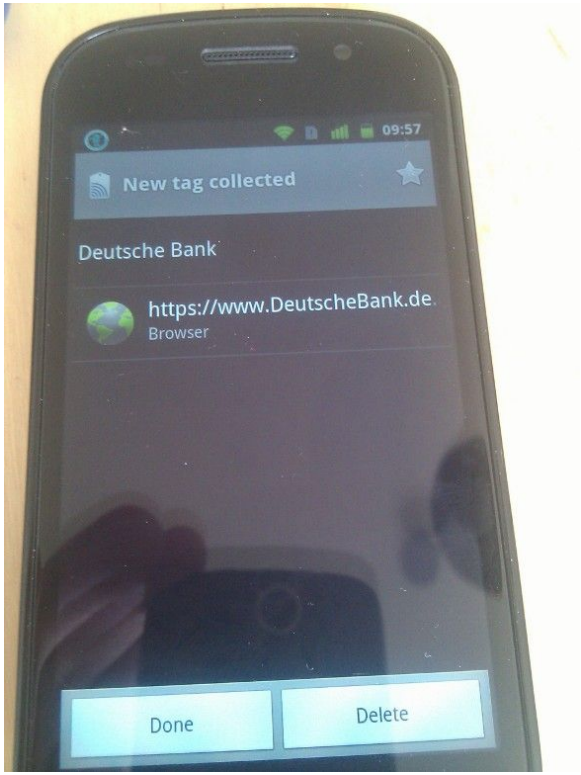
- Stuff like this could be the source for a lot of “fun”

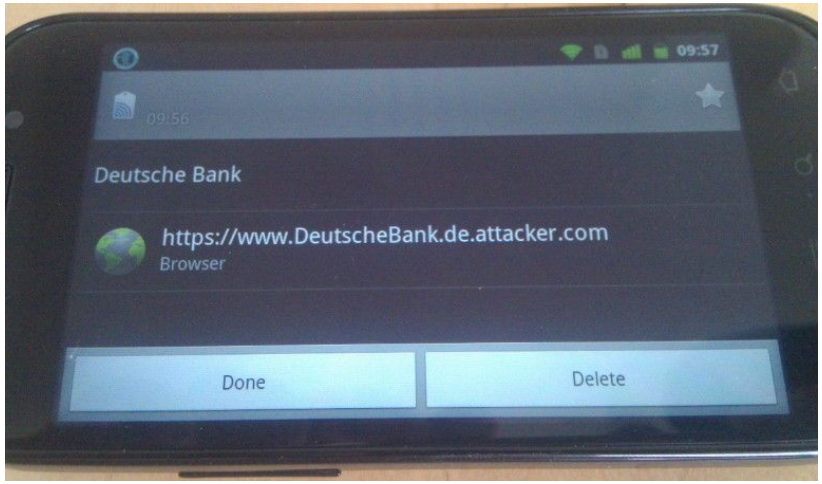
SmartPoster Spoofing and the Nexus S

- Nokia style SmartPoster spoofing?
  - TEL and SMS do not work...
  - HTTP works ... kind of ...









Android NFC/NDEF Forensics

- "Android tag app" records scanned tags in database
  - Database can be analyzed to see:
    - What kind of tags the user interacted with
    - Possible where he was (location tags)

```
/data/data/com.google.android.tag/databases/tags.db
```

- 3rd Party apps (e.g. NXP Tag Writer) also store tag data :-(

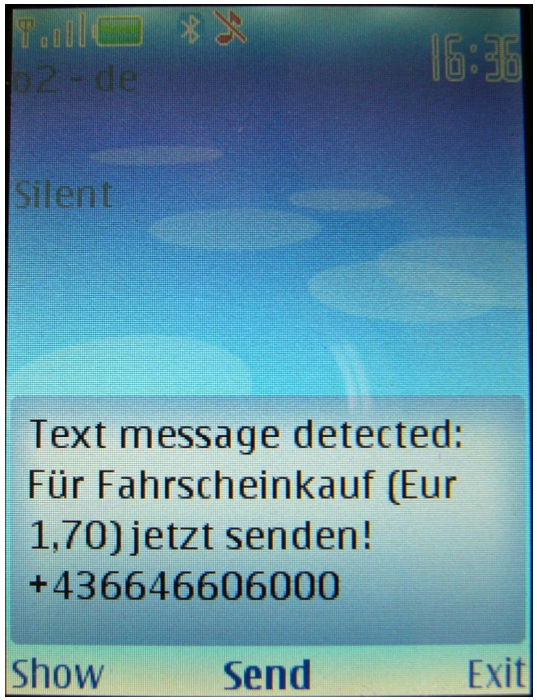
- Talk "NFC for n00bs" by The INTREPIDUS Group [18]

NFC Services

- Small survey to find vulnerable services
  - Places: Vienna Austria and Frankfurt/M. Germany
- Most services use default phone features
  - User doesn't need to install an extra application
- All services use Mifare Classic 1k for their tags
- Conducted survey with just the NFC phone
  - Data analysis on desktop of course

Wiener Linien

- NFC Ticketing for inner city Vienna Austria
  - SMS-based (request and receive ticket via SMS)



Wiener Linien cont.

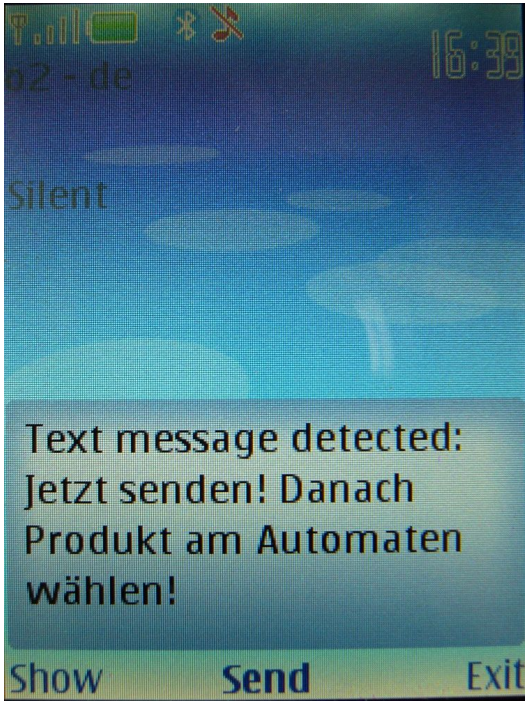
- Tags are read-only
  - Including unused sectors
- Tag attack (sticky tag, discussed later)
  - Use Nokia 6131 spoofing attack to replace actual phone number with “bad” (premium rate) number
- User will trust tag because it **worked** before
  - Maybe spoofing is not even required

Wiener Linien cont.

- Tags are read-only
  - Including unused sectors
- Tag attack (sticky tag, discussed later)
  - Use Nokia 6131 spoofing attack to replace actual phone number with “bad” (premium rate) number
  - Got a 3 Euro ring tone instead of your metro ticket?
- User will trust tag because it **worked** before
  - Maybe spoofing is not even required

Selecta Vending Machine

- Mobile phone payment via SMS (Vienna)
  - Payment via phone bill (SMS ties customer to machine and transaction)



Selecta Vending Machine cont.

- Tags are read-only (including unused sectors)
- Malicious tag attack, but...
- Can be abused to cash out anonymously
  - Make tags pointing to vending machine A and stick them on machine B, C, D, ...
  - Wait at machine A and pull out your free snack
  - (I haven't actually tried this, I swear!)

Vienna ÖBB Handy-Ticket

- Train e-ticketing system

Vienna ÖBB Handy-Ticket cont.

- Tags are read-only (including unused sectors)
- Tag points to website:
  - http://live.a1.net/oebbticket?start=Wien%20Mitte&n=2
- Malicious tag attack (man-in-the-middle via proxy)
  - Steal user credentials
  - User tracking (station is encoded into URL)
  - Inject trojan JAR (auto install bug in Nokia 6131 NFC)
- System seems to be inactive at the moment
  - Never activated or maybe tied to specific carrier (A1)?

RMV Handy Ticket (ConTags)

- Is the e-ticketing system of the Frankfurt area public transport system
- **Requires application install**
- NFC is a non essential part of the system
  - It just selects the train station for you
- Looks boring but has some interesting parts...

RMV ConTags

- Contain two NDEF Records
  - RMV custom record, contains:
    - TNF: 0x04 (urn:nfc:ext:)
    - Type: *rmv.de:hst*
    - Numeric Station ID
    - Station Name
    - Public key signature of custom and URI Record
  - URI Record pointing to time table for that station
    - Only URI is “seen” by the phone if Handy-Ticket app. is not installed

RMV Custom Record

- Record size is variable
  - Bytes 0,1,7 are fixed to 0x01,0x05,0x02
  - Bytes[2,3,4] ⇨ numeric Station ID
    - ID = B[2] * 0x10000 + B[3] * 8 + B[4]
  - Bytes[5,6] ⇨ some number (unknown)
  - Byte[8] is station name length, name follows
  - Byte [8 + station name length] is fixed to 0x03
  - Next byte is length of pub-key signature, sig. follows

RMV ConTag Example

- Tag is from: *Frankfurt/Main Konstablerwache*
  - Total Size: 214 bytes

- Custom Record (154 bytes payload)
  - Station ID: 3000510
  - Name: Konstablerwache

- URI Record (43 bytes payload)
  - http://wap.rmv.de/mobil/tag/request.do?id=3000510

Closer look at the ConTag

- Tags are not truly read-only
  - Read: KeyA (default NDEF key)
  - Write: KeyB (secret)
  - Attack ⇨ break secret B key and overwrite tag

- Tag data area is not locked
  - Unused sectors are left in manufacturer mode
  - Attack ⇨ change key (no updates possible: denial-of-service)

Tag Attacks

- Stick a “bad” tag on top of “good” tag
  - Use tinfoil for shielding off original tag
  - Use RFID-Zapper [8] to fry original tag
  - Sticky paper tag is ~1,20€ (in low quantities) [7]

- Replace original “good” tag with “bad” tag

- Hijack tag of service provider
  - Break write key and overwrite with malicious data
  - Ultimate user trust!

Attack Tags

⇦ Use tinfoil to shield off original tag.

NFC Phone / Service DoS

- Possible Goals
  - Discredit NFC-based service
  - User awareness (this stuff is still kinda insecure)
- Action
  - Write “problematic” content to sticky tags
  - Place sticky tags on top of service tags
- Result
  - Phone / app will crash
    - Users will stop using the service

Notes from the Lab

- No UID spoofing with the Nokia 6131 NFC
  - Can't set UID in Card Emulation mode
  - I know you all wished this was possible!

- Tags are not “formatted” by the phone when storing a new NDEF message
  - Only uses space needed by new message
  - Parts of old data are easily readable
  - ⇨ Wipe tags before passing them to strangers

NFC Phones for the RFID Guys

- JSR-257 and Nokia extensions allow relative low level access to various tag types
  - See my tools: BtNfcAdpaterRAW or MfStt
- Phone or Phone + PDA is much more portable than your USB/serial RFID reader and laptop
- Nexus S and future Android NFC phones
  - Will do all tasks, are able to run my NFC Python lib
- Easy field research without looking too suspicious

Conclusions

- NFC / NDEF is an interesting research field
- With Google pushing NFC will become widespread?!?!
- NFC phones are still buggy... haven't learned from previous research :-(
- Will be interesting what happens with NDEF (SmartPosters) once the payment stuff is used by “normal” people
  - NDEF stuff just calls for trouble :-)
- Folks: start NFC / NDEF hacking!!!1elf

Future Work ...

- “Port” tag low level reading tools to Android
  - Bigger user community
- There will be tones of Android NFC apps / services
  - Analyze and break those :)
- Analyze SecureElement-based services
  - Google Wallet and friends

Questions?

Thank you!

Follow me on Twitter: @collinrm

References

[1] http://www.mulliner.org/nfc/ (NFC Security Tools)
[2] http://www.nfc-forum.org (NFC-Forum)
[3] http://europe.nokia.com/A4307094 (Nokia 6131 NFC)
[4] http://www.rmv.de/coremedia/generator/RMV/Tarife/RMVHandyTicket
[5] http://www.forum.nokia.com/main/resources/technologies/nfc/ (Nokia NFC SDK)
[6] http://www.openpcd.org/openpicc.0.html (Sniffing RFID)
[7] http://www.quio.de/Karten/papieretiketten_13.56/papieretiketten_13.56.html (RFID Tag Shop)
[8] http://events.ccc.de/congress/2005/static/r/f/i/RFID-Zapper(EN)_77f3.html (RFID-Zapper)
[9] http://events.iaik.tugraz.at/RFIDSec06/Program/papers/002%20-%20Security%20in%20NFC.pdf
[10] http://prisms.cs.umass.edu/~kevinfu/papers/RFID-CC-manuscript.pdf
[11] http://doi.ieeecomputersociety.org/10.1109/ARES.2008.105
[12] http://rfidiot.org/ (Copying RFID Credit Cards – ChAP.py)
[13] http://www.cs.virginia.edu/~evans/pubs/usenix08/usenix08.pdf (Mifare CRYPTO1 broken)
[14] http://www.nfc.at/ (NFC in Austria)
[15] http://europe.nokia.com/A4991361 (Nokia 6212 Classic)
[16] http://developer.android.com/reference/android/nfc/package-summary.html
[17] http://www.computer.org/portal/web/csdl/doi/10.1109/NFC.2011.16
[18] http://intrepidusgroup.com/insight/category/nfc/
[19] http://www.madlmayr.at/blog/?p=139