2009 International Conference on Availability, Reliability and Security

# Vulnerability Analysis and Attacks on NFC-enabled Mobile Phones

Collin Mulliner
Fraunhofer Institute for Secure Information Technology (SIT)
collin.mulliner@sit.fraunhofer.de

## Abstract

*Near Field Communication (NFC)-enabled mobile phones and services are starting to appear in the field, yet no attempt was made to analyze the security of NFC-enabled mobile phones. The situation is critical because NFC is mostly used in the area of payment and ticketing. This paper presents our approach to security testing of NFC-enabled mobile phones. Our approach takes into account not only the NFC-subsystem but also software components that can be controlled through the NFC-interface. Through our testing approach, we were able to identify a number of previously unknown vulnerabilities, some of which can be exploited for spoofing of tag content, an NFC-based worm, and for Denial-of-Service attacks. We further show that our findings can be applied to real world NFC-services.*

***Keywords:*** *NFC, Mobile Phones, Vulnerability Analysis, Fuzzing, Phishing, Spoofing.*

## 1. Introduction

Near Field Communication (NFC)-enabled mobile phones and services are starting to appear in the field. According to market research[1], 30% of all sold mobile phones in 2011 will be NFC-enabled. Also, NFC-services heavily rely on the mobile phone network, therefore mobile phone service providers are pushing NFC-enabled phones since these represent an additional source of revenue.

The NFC technology is almost exclusively used for mobile payment and ticketing. Therefore, it is necessary to develop tools and techniques to assess and improve the security of NFC-devices and services.

In this paper, we present our novel approach to the vulnerability analysis of NFC-enabled mobile phones. To the best of our knowledge, no attempt has been made before to analyze or test NFC-enabled mobile phones for vulnerabilities.

We focused our analysis on mobile phones interacting with services that use passive NFC-tags since this is the common setup found in the field today. We acknowledge the fact that future NFC-based payment services will likely be based on smart cards built into every NFC-phone. We however anticipate large scale use of passive tags as soon as NFC-enabled phones hit the critical market penetration.

Vulnerability analysis was partially carried out using fuzzing. Fuzzing was chosen since we did not have access to the source code of the mobile phone. In this work, we present possibilities for fuzzing the NDEF implementation of an NFC-device.

So far, we found several vulnerabilities that allow attacks ranging from spoofing of tag content and interception of tag reading events to Denial-of-Service attacks.

We further present the results of our short survey of NFC-based services. The results verify that the issues discovered through our analysis present a potential threat.

The contributions of this paper are the following:

- We show the steps necessary to perform vulnerability analysis of an NFC-enabled mobile phone, taking into account software components that are not part of the NFC-subsystem but can also be controlled through it.
- We developed a set of tools for security testing of NFC-enabled mobile phones and NFC-services.
- We introduce multiple novel attacks against NFC-enabled mobile phones and services, including a proof-of-concept NFC-based worm.

The rest of this paper is organized as follows: Section 2 provides a brief introduction to Near Field Communication. In Section 3, we discuss related work. In Section 4, we show how we analyzed an NFC-enabled mobile phone and present the issues we found.

1 http://www.abiresearch.com/abiprdisplay.jsp?pressid=719

978-0-7695-3564-7/09 $25.00 © 2009 IEEE
DOI 10.1109/ARES.2009.46

In Section 5, we present our novel attacks against NFC-enabled mobile phones. Section 6 shows that our attacks can be applied to currently deployed NFC-services and in Section 7 we briefly conclude.

## 2. Near Field Communication

Near Field Communication (NFC) is a bidirectional proximity coupling technology based on the ISO14443 and the FeliCa RFID standards. NFC operates in the 13,56 Mhz spectrum and supports data transfer rates of 106, 216, and 424 kbit/s. The typical communication range of an NFC-device lies between 2 and 4 centimeters. There are three modes of operation for an NFC-device. In *reader/writer* mode an NFC-device acts as a proximity coupling device (PCD). In this mode the NFC-device can read and write data stored on NFC compliant passive transponders. In *card emulation* mode an NFC-device acts as proximity inductive coupling card (PICC). In the *peer-to-peer* mode two NFC devices can carry out bidirectional communication to transfer arbitrary data. Currently deployed services mainly utilize the *reader/write* (PCD) mode, therefore we focused our research around this mode.

### 2.1. An NFC Mobile Phone

An NFC-enabled mobile phone integrates an NFC-chip such as the NXP PN532[2] and possible a smart card into an ordinary mobile phone.

The NFC-subsystem, if not switched off, is active as long as the mobile phone is ready to accept user input. The NFC-subsystem constantly scans for NFC-tags. As soon as a tag is detected the phone notifies the NFC-aware application that is running. In this case the application has full control and can decide what to do. If a non-NFC-aware or no application is running the operating system reads the tag. If the tag contains data in a supported format the data is passed over to the application that is registered for handling it.

### 2.2. NFC Data Exchange Format

The NFC Data Exchange Format (NDEF) [4] is a container format used for storing data independent of the tag type. NDEF data is organized in records and messages (a message comprises a set of one or more records). Besides the actual data a record also contains type information about the data, such as a mime-type. NDEF further specifies data types such as the URI [7], Text [6], and Smart Poster [5] record.

A URI can be a simple HTTP URL or more phone centric URI such as `tel` (initiate a voice call) or `sms` (send a predefined short message). A text record contains a human readable string together with a language identifier. The Smart Poster consists of multiple records. In the simplest case these are an `SP` (Smart Poster), a URI, and a Text record. Additional text or image records are possible while there can only be one URI record. This is since the sole purpose of the Smart Poster is to provide additional information about a URI to the human user.

## 3. Related Work

In [1], Haselsteiner and Breitfuß show multiple attacks against NFC-based systems that rely on the lack of link level security of the NFC technology. The attacks range from eavesdropping to data modification, insertion, corruption, and Man-in-the-Middle attacks.

Rieback et al. show in [3] that RFID systems (NFC is based on RFID) can be attacked like traditional computer systems. They show that RFID-based SQL injection attacks as well as self-replicating RFID viruses are possible. Our NFC-worm uses the same transport medium (a RFID/NFC-tag) but infects NFC-enabled mobile phones rather than RFID middleware.

In [2], Madlmayr et al. give an overview of security measures for NFC use cases and devices. They describe the possibility for NFC-based phishing attacks by simply modifying or replacing NFC-tags. Our work presented in this paper shows attacks that additional abuse vulnerabilities existing in NFC-devices.

## 4. Analyzing an NFC Mobile Phone

We analyzed the most common NFC-enabled mobile phone, the Nokia 6131 NFC[3]. The phone only supports third party applications created for the Java2 Mobile Edition[4] (J2ME) platform. The user interface of the phone consists of a keypad and a small color display.

We identified NDEF formatted data as the primary input to an NFC-enabled mobile phone acting as tag reader, therefore we focus our analysis on reading, parsing, and displaying of NDEF formatted data. In order to narrow down our analysis we first determined which parts of the NDEF standard are supported by our device. The device seems to support the URI, Text, and the Smart Poster format with the exception of the Action and Image attributes.

Another part of our work was to analyze the NFC J2ME SDK. We present one result of this analysis in Section 5.2.

An important point of our analysis is to test components that are not part of the actual NFC-subsystem but can be controlled through it, for example the web browser and the mobile phone subsystem. These

2 http://www.nxp.com/products/identification/nfc/

3 http://www.forum.nokia.com/devices/6131_NFC

4 http://java.sun.com/javame/

components might have specific issues that are exploitable only, or possibly much easier, through the NFC-interface.

### 4.1. An NDEF Security Toolkit

For our analysis we have created a toolkit consisting of a flexible NDEF library and a tag reading and writing software for writing the created NDEF data structures to a tag. The tag reading/writing software is based on librfid[5]. All tests described in this paper were carried out using NXP Mifare Classic 1k and 4k RFID tags and a CardMan 5321 RFID reader/writer.

Our NDEF library allows to place arbitrary data into the various NDEF data types such as the URI and Text record. It further allows direct manipulation of length values of various parts of the NDEF format. The reader/writer software does not pre- and post- process any data that is written or read to and from a tag in order to be immune against malformed test data.

Results show that one can easily manipulate the information that is displayed to the user by inserting multiple consecutive whitespace characters into the Text record of a Smart Poster. (These characters cannot be inserted into a Smart Poster through the phone's user interface.) With this kind of manipulation one can prevent the phone from displaying security relevant information to the user or actually making it much harder for the user to locate this information. We will discuss this in more detail in Section 5 where we describe our attacks.

### 4.2. NDEF Fuzzing

Fuzzing is one of the primary testing methods for software for that no source code is available. Therefore, we also conducted a fuzzing trial against our target device. We only spent little time on fuzzing since even if we could find vulnerabilities, such as buffer overflows, there would be no way to exploit those for code injection as no known code injection techniques exist for the Series40 operating system running on our device.

We identified several fields of the NDEF format that can be utilized for fuzzing-based testing. These are the length fields of the NDEF record payload, type, and ID field. Further we tested the contents of the following fields: payload, type, and ID. We also created test cases for the specific NDEF record types URI and Text, since each consist of multiple fields, specifically the abbreviation in the URI record and the language identifier in the Text record.

The fuzzing process itself is very time consuming since there is almost no automation besides the actual fuzzing data generation. The process has four steps and works as follows:

- The fuzzer generates an NFC/NDEF tag image.
- The image is written to the tag.
- The phone reads the tag.
- The human observes the device for any special behavior such as a sudden reboot.

These four steps are repeated until the fuzzing generator's output space is exhausted or until testing is aborted.

**4.2.1. Fuzzing Results** We found a bug in the processing of the NDEF record payload length. The two length values `0xFFFFFFFE` and `0xFFFFFFFF` cause the phone to crash and reset. From our observations it seems that only the GUI system crashes since the user is not asked to enter the PIN of the inserted SIM card (this is normally required after a cold boot of the phone). After four crashes in a row the phone automatically powers down.

Further we found that the telephony subsystem that handles the URIs for `tel` and `sms` (voice call and short message, respectively) crashes when given a phone number that consists of exactly 124 characters (numbers 0 to 9). The crash and reset behaves like the one we observed before. We further believe this is an *off-by-one*[6] error since phone numbers with fewer characters are handled correctly and phone numbers with at least 125 characters cause the display of a message box showing an error message.

### 4.3. The Web Browser

One of the NFC Smart Poster's [5] main purpose is to make the web more accessible to mobile phone users by removing the time consuming and error-prone task of entering URLs like *http://www.example.com/ticket.php?id=12345* using the phone's keypad.

We identified several issues regarding the download of different kinds of files. The most problematic one is that JAR[7] (Java ARchive) files containing application code are downloaded and stored in the application directory as soon as the user activates the browser to open a URL pointing to a JAR file. This situation is critical because the application is executed after the download has completed, also the user still needs to confirm the execution. In comparison, the usual way of installing a J2ME application is to first download the corresponding JAD (Java Application Descriptor)

5 http://openmrtd.org/projects/librfid/

6 http://en.wikipedia.org/wiki/Off-by-one_error

7 http://developers.sun.com/mobility/learn/midp/lifecycle/

file. A JAD contains meta information about the application such as the name, the size, and the URL for the JAR file. After downloading a JAD file the mobile phone usually displays a security warning and provides the user the possibility to inspect the various details of the application. Only after completing these steps an application is installed. In Section 5.2 we show how this behavior can be abused.

Another problem we found is that the web browser does not show the full hostname of a web page being loaded if that hostname is longer than a certain number of characters. Our tests showed that only the last 10 characters of a hostname are displayed. This means a carefully crafted hostname can easily fool the user into believing he is visiting site A while his browser is loading site B.

## 5. Attacking NFC Mobile Phones

Through our analysis performed on the Smart Poster and URI handling and the results of our fuzzing tests, we derived a number of attacks against NFC-enabled mobile phones.

In the first part we show attacks based on the possibility of spoofing tag content. In the second part we present our proof-of-concept NFC-based worm that spreads using tags. In the final part we briefly discuss the possibility of Denial-of-Service attacks against NFC-enabled mobile phones and services.

### 5.1. Smart Poster URI Spoofing

In Section 4.1 we discovered a method to influence the display of Smart Poster data through inserting *space*, *tab*, and *new line* characters into the title record. The basic problem lies in the phone's GUI. The Smart Poster is displayed using a fixed sized message box. Inside the message box the title is displayed before the URI. Since the title can be of arbitrary length the URI may not be displayed if the title already uses the whole space provided by the message box. Further since the URI is not marked in any special way, parts of the title can be perceived as being the actual URI.

Although our NFC-phone offers the possibility to further inspect the contents of a Smart Poster it is unlikely that users actually verify the data displayed to them. Further the attacker can even manipulate the inspection-screen to fool the user. This is achieved through moving the actual URI to the second page of the inspection-screen. This is done by adding multiple *new line* characters and a *dot* to the end of the Smart Poster title, shown in Figures 1(b) and 2(b).

**5.1.1. Attacking The Web Browser** With the possibility to social engineer a victim into loading a malicious website, multiple attacks can be carried out. The most interesting one is a Man-in-the-Middle attack using a web-based proxy. Here the attacker can *steal credentials* or *inject malicious content*.

Attacks like this should work especially well since our test device does not display the URL of the displayed website. This behavior is common among mobile phone web browsers.

In order to try this attack we modified CGIproxy[8] to log all traffic passing through it and added WML [8] support to intercept WAP sites.

Figure 1 shows a simple spoofing attack. Figure 1(a) shows the original tag data as deployed by a service provider while Figure 1(b) shows a spoofed version of the same tag, both Smart Posters look exactly the same when displayed by our test device.

```
Title: Bank of Germany
URL:   https://www.bankofgermany.de
```

(a) Original Smart Poster

```
Title: Bank of Germany\rhttps://www.
         bankofgermany.de\r\r\r\r\r.
URL:   http://www.attacker.com
```

(b) Malicious Smart Poster

**Figure 1. URL Spoofing**

**5.1.2. Attacking The Mobile Telephony Service** The possibility of Smart Poster URI spoofing further allows for attacks against the mobile telephony subsystem using the `tel` and `sms` URIs. Here an attacker can craft a Smart Poster that displays phone number A to the user but, when activated, the phone dials phone number B.

Figure 2 shows an example of the telephony URI spoofing attack. Figure 2(a) shows a Smart Poster that would be deployed at a tourist information and Figure 2(b) shows the malicious version of the same poster. A user cannot distinguish one poster from another. The same attack is also possible for predefined short messages (SMS).

The possible impact of such a spoofing attack is high since the victim can be tricked into calling or sending a short message to a premium rate number (for example a 0900 number like shown in Figure 2(b)). These kinds of attacks are likely to happen since the attacker actually has a financial gain out of them.

8 http://www.jmarshall.com/tools/cgiproxy/

```
Title: Tourist Information
URL:   tel:08001234567
```

(a) Original Smart Poster

```
Title: Tourist Information\r080012345
        67\r\r\r\r\r\r\r.
URL:   tel:09009996668
```

(b) Malicious Smart Poster

**Figure 2. Telephony URI Spoofing**

### 5.2. A Proof-of-Concept NFC Worm

During our analysis of the NFC-enabled mobile phone and the available functionality of JSR-257 [9] (the standardized NFC Java API) we discovered a possibility to abuse the PushRegistry in order to intercept all URI NDEF messages, NDEF messages that only consist out of a URI record (see the Smart Poster description in Section 2.2 for a counter example). The PushRegistry provides a mechanism for applications to register themselves for handling specific data like, for example, images in the JPEG format. As soon as a tag is read, the PushRegistry will analyze the content of the tag and launch the application registered for the corresponding content. The problem is that the PushRegistry allows an application to register for the generic URI type *urn:nfc:wtk:U*. Therefore, the application is executed for all data types that are based on the URI schema, including the URI type itself. Certain types, such as the Smart Poster, cannot be handled by third party applications.

Being able to intercept all URI NDEF tags we designed and implemented a proof-of-concept NFC-worm that is activated every time a tag of this kind is read. Upon activation the worm tries to spread to the tag by writing a URL that points to a copy of itself (on the Internet) back to the tag. In order to stay undetected, the worm-URL is hidden in a Smart Poster that shows the original URL stored on the tag (using the Smart Poster URI spoofing attack described in 5.1). The next phone reading the tag will get infected by the worm when loading the URL. The user still needs to run the worm binary once, but since he simply needs to confirm the execution after the download, it is likely that the worm is executed.

Infected devices are marked with a cookie that is set while downloading the JAR file containing the worm. If a request to the worm-server contains the cookie, the server answers with a HTTP redirect to the URL originally stored on the tag used by the worm. This is done to keep the tag working and to hide the presence of the worm. The content of a Smart Poster containing the worm-URL is shown in Figure 3.

```
Title: http://example.com/ares09/\r\r\r\r\r\r.
URI:   http://nfcworm.net/worm.php?url=http%3A%2F%2F
         example.com/ares09/
```

**Figure 3. NFC-Worm Title and URI**

### 5.3. Denial-of-Service Attacks

Denial-of-Services attacks can be used for destroying the trust relationship between the customer and the service provider. If for example mobile phones crash and reboot every time they touch an NFC-tag, users likely will stop using the service in order to avoid the crash. Such an attack could be implemented by using a sticky paper tag containing a malformed NDEF message and placing it on top of an NFC-tag belonging to the service that is to be discredited. We describe some malformed NDEF messages that cause the phone to crash and reboot in Section 4.2.1.

Users will not be able to link the sticky tag to the crash and therefore will believe that the tag belonging to the service has crashed their mobile phones.

## 6. Security of NFC-based Services

We conducted a small survey of NFC-based services in order to verify the possible attacks we found. We wanted to determine if the components vulnerable to attacks are used by currently installed services. We only surveyed services that are open to the public and are not in test phase for selected users only.

In order to conduct the survey we have developed a second set of tools to inspect the tags deployed by service operators. The tools run on an NFC-enabled mobile phone in order to ease the survey process.

In the rest of this section we will briefly present the services we surveyed. For each service we will explain how it works and how it can be used in order to attack the users of the service.

All of the services described here use only the built-in functionality of our analyzed NFC-enabled mobile phone, no additional software was installed.

### 6.1. Wiener Linien

The Wiener Linien is the public transport system for the inner city of Vienna Austria. The NFC-service of the Wiener Linien is a simple e-ticketing system based on short messages (SMS). In order to use the service, a customer just has to hold his phone up against one of the NFC-tags placed next

9 http://jcp.org/en/jsr/detail?id=257

to the ticket vending machine in every subway station. The phone displays the message: `Text Message detected: Für Fahrscheinkauf (Eur 1.70) jetzt senden! +436646606000` (it basically says: to buy a ticket press send). If the user accepts and sends the request, he will receive his ticket as a short message. The fare is deducted from his phone bill or his paybox[10] account.

A possible attack using this service would be replacing NFC-tags with malicious tags that display the same message and phone number using the Smart Poster spoofing attack described in Section 5.1.2, but will instead point to a premium rate phone number owned by the attacker. The victim will likely notice that he did not receive his ticket, but at this point in time the damage is already done. A careful attacker would only replace a single tag in every station in order to trick the user into believing that something just went wrong when buying the ticket and therefore might just try another unmodified NFC-tag.

### 6.2. Selecta Vending Machines

The Selecta company started installing soda and snack vending machines that offer mobile phone payment using the paybox service. The payment works as follows: each vending machine has a unique identifier in the form of `SNACK257` that is printed on the machine. A customer wishing to buy an item sends a short message containing this identifier to a phone number also printed on the machine. In the next step the machine displays that it is ready to dispense an item. After the customer selected an item the amount is charged to his paybox account. NFC-equipped vending machines feature a tag containing an SMS Smart Poster that contains the same data that is printed on the machine. The customer only needs to read the tag and send the message.

A possible attack on these vending machines could have the goal of buying snacks or soda using somebody else's paybox account. The attack would work as following. The attacker produces a number of fake tags (the vending machine tags are cheap paper tags) that contain the ID of vending machine A. These are mounted on vending machines B, C, and D. The attacker only needs to wait until vending machine A shows that it is ready for selecting an item. This attack has the important advantage that it is nearly untraceable since no premium rate phone number is needed. This attack would again utilize the Smart Poster spoofing vulnerability described in Section 5.1.2.

10 http://www.a1.net/privat/paybox

### 6.3. Vienna ÖBB Handy-Ticket

The ÖBB Handy-Ticket is a web-based train ticketing system in the city of Vienna Austria. Unfortunately the system was disabled so we could not determine how it actually works. We analyzed the NFC-tags deployed for this system. The tags contain a Smart Poster pointing to a website, the URLs look like this: `http://live.a1.net/oebbticket?start=Wien%20Mitte&n=2`.

A Man-in-the-Middle attack as described in Section 5.1.1 would be the most likely attack since here, an attacker could steal user credentials and/or inject malicious content such as a link to a piece of malware like our proof-of-concept NFC-worm (see Section 5.2)

## 7. Conclusions and Future Work

We presented a novel method to perform vulnerability analysis of NFC-enabled mobile phones through the application of fuzzing using NFC-tags. We not only analyzed the NFC-subsystem but also components that can be controlled through the NFC-interface, such as the web browser. The testing tools developed for our analysis are freely available from our website.

Through our analysis we found several security vulnerabilities in an NFC-enabled mobile phone of which some can be abused for phishing, an NFC-based worm, and Denial-of-Service attacks.

Future work includes improving the fuzzing process through automation. Also future NFC-devices are likely to be more complex, efficient methods have to be developed to analyze these.

## References


[1] E. Haselsteiner, K. Breitfuß. Security in Near Field Communication (NFC). In *Workshop on RFID Security*, 2006.

[2] G. Madlmayr, J. Langer, C. Kantner, J. Scharinger. NFC Devices: Security and Privacy. In *Third International Conference on Availability, Reliability and Security*, pages 642–647, 2008.

[3] M. R. Rieback, B. Crispo, A. S. Tanenbaum. Is Your Cat Infected with a Computer Virus? In *Fourth IEEE International Conference on Pervasive Computing and Communications PerCom*, pages 169–179, 2006.

[4] NFC Forum. NFC Data Exchange Format (NDEF). `http://www.nfc-forum.org`.

[5] NFC Forum. Smart Poster Record Type Definition. `http://www.nfc-forum.org`.

[6] NFC Forum. Text Record Type Definition. `http://www.nfc-forum.org`.

[7] NFC Forum. URI Record Type Definition. `http://www.nfc-forum.org`.

[8] WAP Forum. Wireless Application Protocol Wireless Markup Language Specification. `http://www.wapforum.org`.