Practical Attacks on NFC Enabled Cell Phones

Roel Verdult
Institute for Computing and Information Sciences
Radboud University Nijmegen, The Netherlands.
rverdult@cs.ru.nl

François Kooman
SURFnet B.V., The Netherlands.
Francois.Kooman@surfnet.nl

***Abstract*—NFC mobile phones can communicate with other phones, devices, or RFID tags. Those tags are often embedded into smart posters that offer the ability to exchange small files, photos and contact details. The Nokia 6212 Classic is currently the most popular NFC phone. It allows users to easily exchange digital objects using the NFC interface. To do so, two phones should be within the proximity coupling distance of 5 cm. This paper shows the NFC feature that invokes a Bluetooth connection without user consent can be abused to surreptitiously install malicious software on an NFC phone. This results in a serious vulnerability, when, for instance "smart posters" start acting "smarter", install malicious applications and start spreading viruses.**

I. Introduction

The Near Field Communication (NFC) technology is an extension of several Radio Frequency IDentification (RFID) proximity communication standards [9], [10], [13]. It basically combines the RFID standards and describes them with some additional features in two new standards [11], [12]. The two main new features added in the standards are peer-to-peer connections between two active NFC devices (NFCIP) and the emulation of a passive proximity RFID tag. The NFC technology mainly focus on contact-less smartcards that operate at a frequency of 13.56 MHz.

With the introduction of several pervasive devices, NFC is one of most promising techniques for connecting two devices at proximity range. A commercial example are the Smart Posters [20] which contain an embedded NFC tag. Such a poster provides interesting information to active NFC devices that are in proximity range of the embedded NFC tag. It is very user-friendly to provide digitalized information without extensive user interaction. However this has a major drawback from a security point of view, since the user has less control over the automatically triggered events. Although NFC is a promising technique, it is designed for small and lightweight transactions. For larger objects, distance and transfer rates the widely deployed Bluetooth protocol [2] is more applicable.

Content sharing and NFC Bluetooth pairing are features of the Nokia 6212 phone that combine both techniques. Content sharing provides a way to quickly share a contact, note, file or other object from one phone to another. NFC Bluetooth pairing makes it easy for users to let their phone pair with a Bluetooth NFC-enabled device like a headset.

A. Related work

There are two known attacks [16], [23] that use malicious code-injection worms on RFID and NFC systems. They both make use of a maliciously prepared passive tag that is read by the system and causes a buffer-overflow or SQL-injection. These attacks are limited, since the user has to initiate a transaction to read out the tag and parse the data stored on it. Although this limits the probability of a successful attack scenario, NFC is often used in the context where it invites users to touch a tag followed by an confirmation to execute a proposed action. The protocols and techniques that are Nokia proprietary and kept secret to provide security-through-obscurity. There are numerous examples in the literature [6]–[8], [24], [25] showing that once the secrecy of an protocol or cipher is lost, so is its security.

B. Our contribution

This paper analyzes the security vulnerabilities of the NFC features embedded into mobile phones. It demonstrates practical attacks on the latest firmware of the latest Nokia phone that has extensive NFC capabilities, the Nokia 6212 Classic. The attacks focus on Nokia's proposed proprietary "content sharing" and "NFC Bluetooth pairing" capabilities of the phone.

First, this paper shows the NFC communication protocol between the Nokia NFC phone and an NFC tag [21]. This message can be slightly modified and send using any NFC device in passive tag emulator mode. This could trick a user that tries to read a passive NDEF tag into communicating with a malicious NDEF tag emulator. Secondly we analyze the communication between two Nokia NFC phones when one phone tries to send an object to the other. We demonstrate the ability to impersonate a regular NFC device as an initiating (or target) phone. By modifying and sending a recorded NFC communication, we can activate an incoming Bluetooth channel on the target phone.

After tricking the user into touching the malicious NDEF tag and invoking the Bluetooth channel we demonstrate that it is possible to install applications on the phone without user consent. Finally we show that it is possible to escalate the application privileges and register the application in the manufacturer or operator domain of the Nokia 6212 phone. Applications running in these domains have unlimited access within the Java Mobile Edition Application Programmers Interface (Java ME API). They do not require any user interaction during the execution of restricted operation as defined in the Java ME developers manual [14]. In principle it would be possible to use these vulnerabilities to create a worm that spreads itself by touching other NFC phones. The NFC device used for demonstration contains comparable hardware that is embedded into the Nokia 6212 phone.

The paper is structured as follows. Section II introduces NFC techniques and protocol details. Section III provides more information about the NDEF specification and appliances can be found. The usability and differences of an NFC and Bluetooth connection and their relevance to the practical attacks are described in section IV. Details about the content sharing feature of the Nokia 6212 NFC phone is described in section V. Section VI presents eavesdropped traces which were made during a content sharing transmission. Section VII introduces and explains the layout of an NDEF Bluetooth Pairing tag. Section VIII shows the practical attacks that were executed on an NFC phone when it came in proximity of a regular NFC reader. In the last section we evaluate the impact of our attacks and a few possible countermeasures that could be taken into account to prevent the described attack scenarios.

II. NFC techniques and protocol

One of the main new features of NFC is the emulation of passive tags (*target* in NFC terminology). Emulation means here that an NFC device acts like a passive target and does *not* generate a radio frequency (RF) field. It communicates like a original passive tag to the RFID reader (*initiator* in NFC terminology). This makes the technology backwards compatible with already deployed RFID proximity readers. Passive NFC tags are often used for the storing small messages, trigger an application event or to redirect a user to online content. The technique used for this is called NFC Data Exchange Format (NDEF) and is thoroughly explained in the next section.

NFCIP is a special operation mode of NFC, which is defined in the ISO/IEC 18092 standard [11]. It provides a peer-to-peer communication channel between initiator and target. In passive mode, only the initiator is responsible for generating the RF field. In active mode both the initiator and target need to generate their own RF field and activate it intermittently. NFCIP works in master-slave mode, where the initiator starts with sending some data to which the target has to respond. Situation 1 in Figure 1 shows the class reader/tag relationship where situation 2 shows the NFCIP situation.

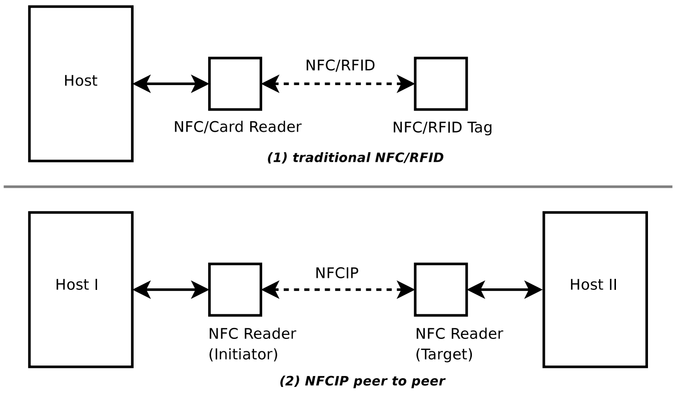

Fig. 1: NFCIP Communication

For our research we are using the NFC device from Advanced Card Systems (ACS)[1]. This reader contains a chip that is manufactured by NXP Semiconductors, one of the founders of NFC Forum[2]. The NFC controller chip [22] from NXP supports most of the extensive NFC features and is backwards compatible with several proprietary RFID protocols using the same frequency. To use all the features of these PN53x chips, it requires to use a low level command interface to the chip. For this we used the publicly available open-source library libnfc[3]. With this library it is possible to execute all low-level commands required to invoke the presented attacks.

III. NFC Data Exchange Format

NFC Data Exchange Format (NDEF) format standardizes how to store data on a smartcard that is compatible with one of the NFC Forum tags. These tags can be used to store for example bookmarks, business cards, alarm clock settings, Smart Posters information, Call or SMS Requests and several other objects. The following NDEF message represents a Smart Poster that links to the libnfc project website. When it gets touched by an NFC phone, the user is asked to open the browser and visit *http://libnfc.org*.

```
0321D1021C537091 01095402656E4C69 |.!...Sp...T.enLi|
626E666351010B55 036C69626E66632E |bnfcQ..U.libnfc.|
6F7267FE00000000 0000000000000000 |org.............|
```

The NDEF specifications [17] to decode this binary data set are publicly available on the NFC Forum website. The NDEF headers in this example use as Type Name Formats (TNF) the value `0x01`, which represents a “NFC Forum Well-known Type”. These well-known types are defined in the Technical Specification [17] from the NFC Forum.

```
0321 NDEF message TLV, Value length = 0x21
 D1 NDEF header, ME=1, MB=1, SR=1, TNF=0x01
 02 Record type length = 0x02
 1C Payload length = 0x1C
  5370 Record type = "Sp" (Smart poster)
  91 NDEF header, MB=1, SR=1, TNF=0x01
  01 Record type length = 0x01
  09 Payload length = 0x09
   54 Record type = "T" (Text)
   02 UTF-8, two-byte ISO language code
   656E Language code = "en" (English)
   4C69626E6663 Payload = "Libnfc"
 51 NDEF header, ME=1, SR=1, TNF=0x01
 01 Record type length = 0x01
 0B Payload length = 0x0B
  54 Record type = "U" (URI)
  03 URI Identifier code, prefix = "http://"
  6C69626E66632E6F7267FE Payload = "libnfc.org"
 FE TLV Terminator
```

This example shows that an NDEF message can contain multiple records of various types. A link to online content will most likely invoke a visit to the website. It is the responsibility of the NFC device that is processing the message to decide if there has to be any user interaction. When a combination of different record types are used it is unclear which action is presented to the user to accept. For most record types the Nokia 6212 NFC phone implemented this interaction by default. However, the NDEF attack shows that this interaction is clearly not specific enough for a user to assess all the risks that are involved.

IV. NFC and Bluetooth

The most common way to transmit an object from one phone to another is by sending it over a Bluetooth connection. When more phones supporting NFC technology this could change, but one should take into account that there are three major differences between NFC and Bluetooth, namely the transfer

[1] ACS ACR122, firmware ACR122U102, http://nfc-reader.com
[2] http://www.nfc-forum.org
[3] http://www.libnfc.org

rate, communication distance and initialization speed. The data transfer rate of NFC is substantially lower than a Bluetooth connection. This makes it more convenient to use Bluetooth for transferring bigger objects. This difference is not noticeable for smaller objects like contacts, text messages and business cards. The communication distance for two NFC devices is of about 5 centimeters, where a Bluetooth connection can take place over a distance of several meters. It can be inconvenient to keep two phones in proximity range of each other during the transfer of a big object. For smaller objects, the transfer finishes by merely a touch of one second, this has considerably less influence on the usability. The initialization phase of an NFC connection is finished in less than 10ms. This is considerable less than the several seconds a Bluetooth connection requires before it is ready. Advanced Bluetooth services only work after a successful pairing process. During this process the parties involved need to actively advertise and discover each other Bluetooth Media Access Control (MAC) address, followed by the entering of a mutually agreed PIN code. In contrast, the NFC initialization is handled automatically and does not require any user interaction.

The data that is required for a Bluetooth pairing is rather small, it only needs the MAC address and the PIN code. These two data fields can be combined in one NDEF message and transmitted using an NFC communication. This brings together the fast initialization between two NFC devices combined with the bigger range and speed of a Bluetooth connection.

V. Content Sharing

The Content Sharing feature can be invoked by browsing to a passive object on the phone. The object is passive when it only represents a dataset and does not contain any executable code. Select for instance a picture taken with the built-in camera, choose the "Share" option in the "Options" menu or use the NFC menu by choosing the "Share to device" option. This will make the phone search for another NFC capable device in its proximity and starts sending this object to the other device.

Content Sharing proceeds by establishing an NFCIP connection to share the actual data, or by enabling Bluetooth when the shared content exceeds a certain size limit. A simple "Business card", "Call request" or "Note" is transferred using the NFC communication. Bigger objects like a "Gallery item" containing a picture, or a larger "Note", is transferred over Bluetooth.

Content Sharing support is enabled by default and makes it more convenient for the initiator to share passive objects. For the target side, this is more troubling, it receives any incoming object without prior confirmation. Even when the option "by confirmation" is selected, the phone only notifies the user after completing the incoming transmission.

The Nokia 6212 "NFC Settings" allowed configurations:

Setting	**Options** (default = bold)
NFC	(**on** / off)
Content sharing	(quick / **by confirmation** / not available)

Content Sharing automatically enables Bluetooth on both the initiating and target phone if this was not active before. It does not matter if the phone has disabled the Bluetooth functionality, the phone is practically forced to enable it. Bluetooth devices usually have to be paired in order for any exchange of data to take place. An exception to this is the OBject EXchange (OBEX) service which can accept transfers directly. The authorization for sending a file is put at the application layer. When Content Sharing is initiated, the phone tries to find an NFCIP target in proximity range. After setting up the NFC connection the initiator triggers an OBEX configuration change on the target phone that allows data transfers from the Bluetooth device with the MAC address that belongs to the initiator.

Only passive objects can be shared from the phone. Sharing installed applications is not allowed and invokes the following error message: "This file is copyright protected". Even when it concerns a self-written application for which it was never specified that it should be "copyright protected".

VI. NFCIP Data Analysis

We captured the data exchanged between two NFC phones using two NFC reader devices. For logging and relaying the NFC communication between the two Nokia phones we published the nfcip-relay tool[4]. The high level overview of this setup is shown in Figure 2. NFC communication is compatible with the ISO/IEC 14443 Type A Proximity RFID standard [10]. Therefor we could verify the captured data at a network frame level using the Proxmark[5] RFID analyzer. To capture the frames we used the modified firmware written by Gerhard de Koning Gans [4], [5]. The data is wrapped in several network layers, for readability we focus on the captured data that was transferred on application level between the two NFCIP devices. This makes it easier to extract the actual commands we have to imitate later, when mimicking a phone.

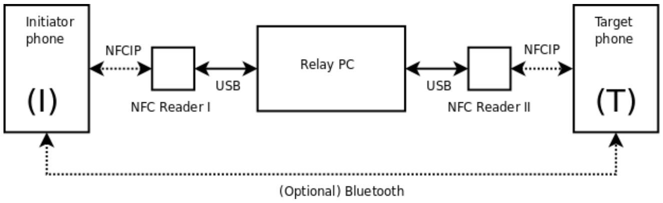

Fig. 2: NFCIP Relay Design

NFC Reader I is activated by the initiating phone, which means that the reader was placed in target mode. NFC Reader II activates the target phone, which means the reader is set to initiator mode. For the relay to work it should activate the target phone with the same parameters that were used by the phone to activate NFC Reader I. These exact parameters were used to activate the target phone with NFC Reader II. After this initialization the actual data is relayed between NFC Reader I and NFC Reader II.

```
1 I->T: 1620                               |. |
  T->I: 1630                               |.0|
2 I->T: c285                               |..|
  T->I: 8285                               |..|
3 I->T: 15e00091020a4872 10d1020461630101 |......Hr....ac..|
        31005c0d0d016e6f 6b69612e636f6d3a |1.\...nokia.com:|
        7368653101c00a49 6e69746961746f72 |she1...Initiator|
        00                                 |.|
  T->I: 15f00091020a4872 10d1020461630101 |......Hr....ac..|
        31005c0d0a016e6f 6b69612e636f6d3a |1.\...nokia.com:|
        7368653101400754 617267657400     |she1.@.Target.|
4 I->T: c18501                             |...|
```

[4] http://nfcip-java.googlecode.com

[5] http://www.proxmark.org

```
  T->I: 818501                            |...|
5 I->T: c1c51091020a4873 10d1020461630101 |......Hs....ac..|
        31005c0d0d016e6f 6b69612e636f6d3a |1.\...nokia.com:|
        7368653101c00a49 6e69746961746f72 |she1...Initiator|
        00                                |.|
  T->I: 81c51091020a4873 10d1020461630101 |......Hs....ac..|
        31005c0d0a016e6f 6b69612e636f6d3a |1.\...nokia.com:|
        7368653101400754 617267657400     |she1.@.Target.|
6 I->T: 15a001                            |...|
  T->I: 15b001                            |...|
7 I->T: 15e011d20c787465 78742f782d764361 |.....xtext/x-vCa|
        7264424547494e3a 56434152440d0a56 |rdBEGIN:VCARD..V|
        455253494f4e3a32 2e310d0a4e3b4348 |ERSION:2.1..N;CH|
        41525345543d5554 462d383b454e434f |ARSET=UTF-8;ENCO|
        44494e473d384249 543a446f653b4a6f |DING=8BIT:Doe;Jo|
        686e0d0a54454c3b 505245463b564f49 |hn..TEL;PREF;VOI|
        43453b454e434f44 494e473d38424954 |CE;ENCODING=8BIT|
        3a2b333132343132 33343536370d0a45 |:+31241234567..E|
        4e443a5643415244 0d0a             |ND:VCARD..|
  T->I: 818502                            |...|
8 I->T: 0040                              |.@|
  T->I: 0040                              |.@|
```

This communication trace shows a recording of the NFCIP transmission when sharing an electronic business card. Messages 3 and 5 seem to be almost identical, it is unclear why Nokia uses this redundancy in their NFCIP transmission. Message 7 contains the actual content of the business card that appears to be encoded as a vCard[6]. An NFCIP replay with a regular NFC device of the initiator part would immediately work and result in the vCard being delivered to the target. When for instance a note is sent from the initiator to the target which exceeds the maximum length of a message, NFCIP chaining is used to send all the data. This means that the initiator sends a message where the first byte indicates (target specifier) that more data is coming after this message. The target responds with an empty message until all data from the initiator was received. In our case the data was split in blocks of maximum size of 236 bytes. Furthermore, we observed that all messages before message 7 are identical for both sending a note, vCard or image. Below we show messages from 7 onward where it activates the Bluetooth connection.

```
7 I->T: 15e01191020a4872 10d1020461630101 |......Hr....ac..|
        31005c0c25016e6f 6b69612e636f6d3a |1.\.%.nokia.com:|
        6274310000226566 eb815a0204000000 |bt1.."ef..Z.....|
        0000000000000000 00000000000a496e |..............In|
        69746961746f7200                  |itiator.|
  T->I: 818502                            |...|
8 I->T: 0040                              |.@|
  T->I: 81c52191020a4873 10d1020461630101 |..!...Hs....ac..|
        31005c0c22016e6f 6b69612e636f6d3a |1.\.".nokia.com:|
        6274310000226566 eea65a0204000000 |bt1.."ef..Z.....|
        0000000000000000 0000000000075461 |..............Ta|
        7267657400                        |rget.|
9 I->T: 15a002                            |...|
  T->I: 0040                              |.@|
```

Messages 7 and 8 are again similar to messages 3 and 5, but now a Bluetooth address is encoded in the message body. For the initiator this is `00226566eb81` and for the target it is `00226566eea6`. Again, every protocol run results in exactly the same data for every message. Replaying the above initiator messages results in an activation of Bluetooth on the target phone and acceptance of file transfers from the specified Bluetooth MAC address in message 7. We tried all configurations for the "Content sharing" option in the "NFC Settings" menu, but there was never a notification or confirmation reported back to the user during the Bluetooth activation process.

The phone uses the OBject EXchange (OBEX) protocol to exchange data over Bluetooth. OBEX is a protocol to efficiently transfer binary objects between devices. Running the protocol as before, which triggers the activation of Bluetooth, allows a data transfer using the OBEX protocol from the address specified in message 7. By using a phone as initiator to share a file to a PC that has an NFC reader attached we modify the Bluetooth hardware address of the target in the response message 7 to the hardware address of the PC's Bluetooth device. The tool `hcidump`[7] captures the Bluetooth data received at the PC. We observed that the phone tried to send a file using OBEX push. After configuring our PC to accept OBEX Push transfers from any device we were able to receive the file from the phone. We then turned this around and tried to send to the phone a file using OBEX push, which also worked flawlessly without requesting any user confirmation on the phone.

From a phone it is not possible to send applications (MIDlet suites) to another phone, this is possible from a PC as any file transfer is allowed when the correct Bluetooth hardware address was sent to the target in message 7. We observed that after pushing an application to the phone, it automatically gets installed and bookmarked in the phone menu. A possible threat is that this way it is possible to install an application on the phone which has some push registry entries activated that can launch the application on certain external events like a timer or by presenting a certain NFC tag in proximity distance of the phone. By default the permissions of the application do not allow any operation that may cause the user (financial) damage. The application won't be able to access the network or personal data without user acceptance.

VII. NDEF and Automatic Pairing

This section shows the use of NDEF messages [17]. The messages related to "Connection Handover" [19] are here of interest as they specify the use of Out of band (OOB) mechanisms to establish Bluetooth pairings. In particular appendix B.2 "Handover to a Bluetooth Carrier". This feature is provided with products like the Nokia Bluetooth BH-505 or BH-210[8] headset.

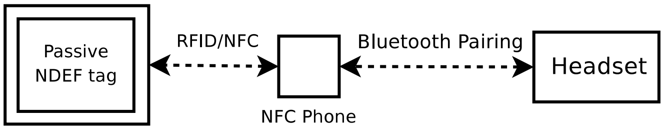

Fig. 3: NFC OOB Bluetooth Pairing

The product comes with a pairing tag that enables NFC phones to configure a paired bluetooth connection channel for diverse audio purposes. It simply requires a user to bring the NFC phone in proximity distance of the tag that is embedded into the headset. The phone shows a "connect to" confirmation before it connects to the headset. For our example we created an NDEF pairing tag with a "1234" PIN code, with the Bluetooth address `00:11:22:33:44:55` and "Poster" as description. Looking closer to the content of an NDEF tag we will find the following data.

```
0330D40C216E6F6B 69612E636F6D3A62 |.0..!nokia.com:b|
```

[6] http://www.w3.org/TR/vcard-rdf/

[7] http://www.bluez.org/

[8] http://europe.nokia.com/bh-210

```
0324 NDEF message TLV, Value length = 0x24
 D4 NDEF header, ME=1, MB=1, SR=1, TNF=0x04
 0C Record type (RT) length = 0x0C
 15 Payload length = 0x15
  6E6F6B69612E636F6D3A6274 RT = "nokia.com:bt"
  00 Configuration = 0x01 (PIN)
  001122334455 Bluetooth MAC = "00:11:22:33:44:55"
  Bluetooth Class of Device (CoD)
   20 Major Service Class = Audio
   04 Major Device Class = Audio/Video
   18 Minor Device Class = Headphones
  31323334 PIN Code = "1234"
  000000000000000000000000 PIN Code Padding
  06 Length of name = 0x06
  506F73746572 Name = "Poster"
 FE TLV Terminator
```

Fig. 4: NDEF pairing tag

```
7400001122334455 2004183132333400 |t..."3DU ..1234.|
0000000000000000 00000006506F7374 |............Post|
6572FE                            |er.|
```

This pairing tag uses a proprietary NDEF definition. This is recognizable by looking at the specified TNF value, which has in this case the value 0x04 that represents an NFC Forum external type defined in the Record Type Definition document [18]. Looking at the NDEF message in detail it shows a Record type of “nokia.com:bt”. This is a proprietary Bluetooth Pairing tag that is defined by Nokia. This tag was introduced at the release of the first public Nokia 6131 NFC phone, but is compatible and supported by current generation Nokia NFC phones.

After confirmation the phone automatically starts pairing using the supplied Bluetooth MAC Address and PIN Code. This even happens when the Bluetooth connection is disabled. The phone enables it without explicitly notifying the user of this activation. When the pairing succeeded it starts connecting to the headset interface available in the Bluetooth protocol stack of the headset.

It is possible to replace the Bluetooth MAC Address with an address that belongs to a PC. The phone pairs successfully but can not connect to the headphone interface. It will deactivate Bluetooth immediately after an unsuccessful headset setup. It is interesting to see that the pairing between the two addresses is stored in the phone as a successful and trusted channel, even when it failed to connect to the headset.

The Bluetooth specification [2] mentions an OOB mechanism as part of “Simple Pairing”. The detailed white paper about simple pairing [1] focuses more on the security and less on the actual implementation details. The paper mentions that the OOB mechanism either uses one-way or two-way authentication where cryptographic information can be exchanged using the NFC communication channel. Figure 4 shows that the advanced cryptographic features of the OOB mechanism supports are not required.

VIII. Attacking a cell phone

This section describes three attacks, one that uses Content Sharing, the other a NDEF pairing tag and a the last one a combination of both. The properties of these attacks are shown in figure 5.

Using the content sharing feature it is possible to upload a malicious application to the cell phone without any user consent. By sending a modified NFCIP communication trace of the content sharing feature, it is possible to activate Bluetooth on the target phone. An attacker could force this upon a victim just by standing next to him and holding the attacking NFC device in proximity range of his phone. This invokes the content sharing feature between NFC device, malicious PC and the phone. We use an OBEX transfer to upload a malicious application (MIDlet suite) to the mobile phone. The victim does not have to interact with his phone to accept any incoming connection, which makes this attack very practical. It could be performed just by holding the attacking NFC device next to the victims pocket that carries the phone.

The malicious application tries to read out the International Mobile Equipment Identity (IMEI) number of the mobile phone. If this application is not running in the manufacturer domain, the application has no right to access this phone specific information. To overcome this problem we combine it with the NDEF pairing attack.

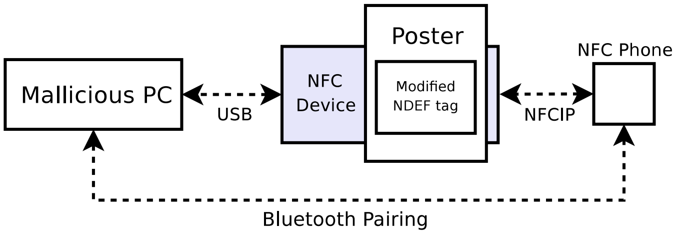

Fig. 6: Smart Bluetooth Poster

The NDEF pairing attack is demonstrated with a malicious smart poster that we call a “Smart Bluetooth Poster”. The setup is shown in figure 6, it looks like a regular smart poster, but it has an NFC device attached to the back instead of a working NDEF tag. An attacker could easily create such a malicious poster, but an existing commercial poster could be altered as well. A typical commercial smart poster contains an NDEF tag that represents a bookmark to the promoted product. It is fairly simple to break the original tag that is embedded into the poster with a tool like the rfidzapper [3].

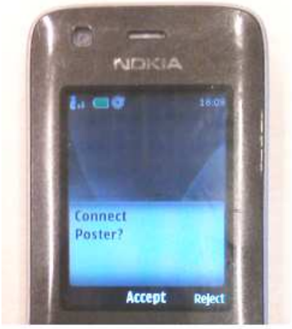

Fig. 7: NFC Phone asked to accept Bluetooth pairing

The NFC device that is attached to the back of the poster emulates a passive NDEF pairing tag. The user has no knowledge about the NFC device and his phone can not detect the difference between a genuine tag and our emulated one. If the user touches the NDEF pairing tag and accepts only one vague notification shown in figure 7 it starts pairing a Bluetooh connection. When the pairing is complete the phone disables the

Attack	**Requires user accept**	**Activates BT**	**Pairs BT**	**Accepts Upload**	**Change Access Rights**
Content Sharing	**no**	**yes**	**no**	once	no
NDEF Pairing	once	no	**yes**	many	**yes**
Combined attack	once	**yes**	**yes**	many	**yes**

Fig. 5: Practical NFC attacks and their properties

Bluetooth immediately. Luckily we have our Content Sharing attack to activate the Bluetooth again. The Bluetooth connection stays active for at least 20 seconds. It requires that the connection is originated from the MAC address that is embedded into the NFCIP message. But it does not require an OBEX Push connection. We were able to use our just paired connection to access all available Bluetooth services that are provided by the NFC phone.

One of these services is the proprietary Nokia PC Suite interface. This interface allows the user to backup the phone. Using the PC Suite interface, we were able to get read and write access to the complete phone memory. This includes the part where Nokia stores the access conditions per installed application. Using the gnokii [9] tool and the information described in the thesis of François Kooman [15] we were able to alter the access rights of the application and register it into the manufacturer or operator domain. The new configuration allowed our application for example to read out the IMEI number of the phone.

An application running in one of these domains does not require any confirmation at all, it can for example start automatically, access the GSM network and personal address book without the consent of the user. By changing the access conditions it enables a malicious attacker to spread a virus through a smart poster. In principle it would be possible to spread the virus from one NFC phone to another just by touching. In the manufacturer domain an application has the rights to control the Bluetooth and NFC interface without user approval.

IX. Conclusions

The attacks we presented are very practical and a serious threat for users to get infected by a virus and unwanted spyware application. Combining the NFC, Bluetooh and proprietary Nokia features of the Nokia 6212 Classic NFC phone we were able to upload and install an application into the manufacturer domain. In this domain the application has no access limitations and does not need any confirmation from the user to access advanced features like the GSM network or securely stored information like the private address book.

We encountered some vulnerabilities in the NFC related features of the Nokia 6212 phone. The phone should not trust a pairing that just failed to connect to a headset. The activity of the NFC interface should be more clear to the user, where it is connecting to and which access rights an Initiator of NDEF tag requires. The security risks of following a bookmark may differ from pairing a Bluetooth connection.

We strongly encourage manufacturers to drop their proprietary solutions for NFC pairing and migrate from old fashioned PIN code pairing to the advanced cryptographic solutions proposed by the Bluetooth Special Interest Group [1].

With respect to the responsible disclosure principle we disclosed our findings in advance to Nokia. We encouraged Nokia to look into these problems before releasing a next model NFC phone.

References

[1] Simple pairing whitepaper, 2006. Revision V10r00.

[2] Bluetooth Specification plus EDR, Master Table of Contents & Compliance Requirements, 2007. Version 2.1.

[3] J. Collins. Rfid-zapper shoots to kill. *RFID Journal*, 2006.

[4] Gerhard de Koning Gans. Analysis of the MIFARE Classic used in the OV-chipkaart project. Master's thesis, Radboud University Nijmegen, 2008.

[5] Gerhard de Koning Gans, Jaap-Henk Hoepman, and Flavio D. Garcia. A practical attack on the MIFARE Classic. In *8th Smart Card Research and Advanced Application Workshop (CARDIS 2008)*, volume 5189 of *Lecture Notes in Computer Science*, pages 267–282. Springer Verlag, 2008.

[6] Flavio D. Garcia, Gerhard de Koning Gans, Ruben Muijrers, Peter van Rossum, Roel Verdult, Ronny Wichers Schreur, and Bart Jacobs. Dismantling MIFARE Classic. In *13th European Symposium on Research in Computer Security (ESORICS 2008)*, volume 5283 of *Lecture Notes in Computer Science*, pages 97–114. Springer-Verlag, 2008.

[7] Flavio D. Garcia, Peter van Rossum, Roel Verdult, and Ronny Wichers Schreur. Wirelessly pickpocketing a MIFARE Classic card. In *30th IEEE Symposium on Security and Privacy (S&P 2009)*, pages 3–15. IEEE Computer Society, 2009.

[8] Flavio D. Garcia, Peter van Rossum, Roel Verdult, and Ronny Wichers Schreur. Dismantling SecureMemory, CryptoMemory and CryptoRF. In *17th ACM Conference on Computer and Communications Security (CCS 2010)*, pages 250–259. ACM/SIGSAC, 2010.

[9] Identification cards — contactless integrated circuit(s) cards — vicinity cards (ISO/IEC 15693), 2000.

[10] Identification cards — contactless integrated circuit cards — proximity cards (ISO/IEC 14443), 2001.

[11] Information technology — telecommunications and information exchange between systems — near field communication interface and protocol 1 (NFCIP-1) (ISO/IEC 18092), 2004.

[12] Information technology — telecommunications and information exchange between systems — near field communication interface and protocol 2 (NFCIP-2) (ISO/IEC 21481), 2005.

[13] Specification of implementation for integrated circuit(s) cards (JICSAP/JSA jis x 6319), 2005.

[14] Mobile Information Device Profile for Java 2 Micro Edition (MIDP), 2002. Version 2.0.

[15] François Kooman. Using Mobile Phones for Public Transport Payment. Master's thesis, Radboud University Nijmegen, Institute for Computing and Information Sciences, The Netherlands, 2009.

[16] C. Mulliner. Vulnerability Analysis and Attacks on NFC-enabled Mobile Phones. In *Proceedings of the 1st International Workshop on Sensor Security (IWSS) at ARES*, pages 695–700, Fukuoka, Japan, March 2009.

[17] Technical Specification, NFC Data Exchange Format (NDEF), 2006. NDEF 1.0.

[18] Technical Specification, NFC Record Type Definition (RTD), 2006. RTD 1.0.

[19] Technical specification, connection handover, 2010. Connection Handover 1.2.

[20] Technical Specification, Smart Poster Record Type Definition, 2006.

[21] Technical Specification, Type 4 Tag Operation, 2007.

[22] Near Field Communication (NFC) controller — Product data sheet, 2007. Rev. 3.2.

[23] Melanie R. Rieback, Bruno Crispo, and Andrew S. Tanenbaum. Rfid malware: truth vs. myth. *IEEE Security & Privacy*, 4(4):70–72, 2006.

[24] Roel Verdult. Proof of concept, cloning the OV-chip card. Technical report, Radboud University Nijmegen, 2008.

[25] Roel Verdult. Security analysis of RFID tags. Master's thesis, Radboud University Nijmegen, 2008.

[9] http://www.gnokii.org