DEMO: NFCGate - An NFC Relay Application for Android

[Extended Abstract]

Max Maass, Uwe Müller, Tom Schons, Daniel Wegemer and Matthias Schulz
Secure Mobile Networking Lab
Technische Universität Darmstadt
Darmstadt, Germany
{mmaass,umueller,tschons,dwegemer,mschulz}@seemoo.tu-darmstadt.de

ABSTRACT

Near Field Communication (NFC) is a technology widely used for security-critical applications like access control or payment systems. Many of these systems rely on the security assumption that the card has to be in close proximity to communicate with the reader. We developed NFCGate, an Android application capable of relaying NFC communication between card and reader using two rooted but otherwise unmodified Android phones. This enables us to increase the distance between card and reader, eavesdrop on, and even modify the exchanged data. The application should work for any system built on top of ISO 14443-3 that is not hardened against relay attacks, and was successfully tested with a popular contactless card payment system and an electronic passport document.

Categories and Subject Descriptors

K.4.4 [**Computing Milieux**]: Computers and SocietyElectronic Commerce[Security]; C.3 [**Computer Systems Organization**]: Special-Purpose and Application-based systems—*Smartcards*

General Terms

Security

Keywords

Near Field Communication, Relay Attack, Android

1. INTRODUCTION

Near Field Communication (NFC) technology is used for many applications, ranging from access control and payment systems to electronic passports. Due to the low range of the wireless signal, many systems operate under the assumption that successful communication between reader and token implies proximity. A relay attack breaks this assumption by placing a proxy reader near the token and an emulator near the reader. The reader will now communicate with the emulator, which will forward all messages to the proxy reader that will in turn communicate with the token and send back the response. The relay attack circumvents traditional authentication protocols, as the reader is still communicating with the correct token, which will be able to correctly perform any authentication protocol.

Previous works have implemented this relay attack using custom hardware [2], Nokia / Blackberry phones [1] and customized Android systems [3]. Another implementation [5] does not require a modified Android system, but only works for a subset of NFC protocols, which does not include the protocols used for many security-critical applications. It also does not emulate the unique identifier (UID) of the card, which is sometimes checked as part of the security model of deployed systems.

We propose NFCGate, an NFC relay application using a rooted stock Android phone, capable of forwarding any NFC protocol based on ISO 14443-3 and supported by Android. This allows evaluating the security of NFC systems against many types of attacks, including relay, replay, and modification by an active Man in the Middle (MitM).

2. METHODOLOGY

In order to implement an NFC relay on Android, two components need to be implemented: A *reader device* that communicates with the NFC token, and an *emulator device*, which communicates with the NFC reader (cf. Fig. 1). Android offers card reader functionality for many cards using the `android.nfc` API. Starting with Android 4.4, it also offers a *Host Card Emulation* (HCE) API, which allows applications to simulate an NFC tag for an NFC reader. However, the HCE API imposes some limitations:

- Applications need to register their Application ID (AID) in advance and only receive traffic after the reader has sent an ISO 14443-3 SELECT for that AID.
- The Unique Indentifier (UID) of the emulated tag is randomized and indicates that it is an emulated tag by setting the first byte to `0x08`.

In our work, we solved both of these problems using *Cydia Substrate* [4], a framework that allows applications to patch Android code (both native C and high-level Java) without recompiling and reflashing the system. Instead, code is injected into the *zygote* process, from which the NFC daemon is forked. Once it has been forked from *zygote*, Substrate allows developers to hook into and change certain functions.

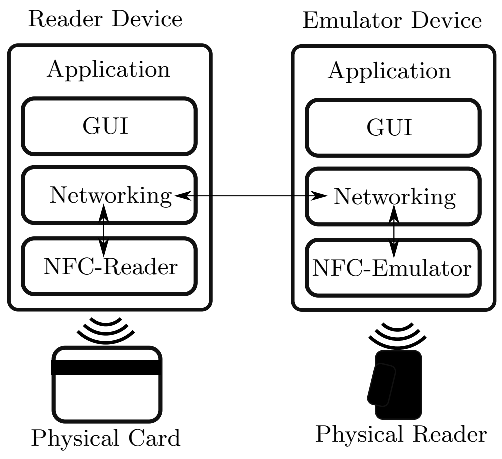

Figure 1: System schematic

AID detection.

As Android only routes SELECTs for a specific AID to our application, we modified the two functions Android uses to detect which AID (if any) is contained within each received NFC command. *NFC_SetStaticRfCback()* is a part of *libnfc-nci*, a native C library which Android uses to interface with HCE-enabled NFC chips, whilst (*findSelectAid()*) is part of the Android OS and written in Java.

Substrate allowed us to inject code into both of these functions to return a wildcard AID—regardless of the input value—which we could then register for our application. As this potentially breaks other applications using HCE, we also added an inter-process communication (IPC) component, which allows us to dynamically enable the patch when it is needed, and preserve the traditional behaviour for all other times.

UID emulation.

By default, the Android HCE stack creates a randomized UID for each HCE session, and the API does not offer any options to influence this behaviour. However, *libnfc-nci* contains a function that allows uploading raw configuration streams to the NFC chip. This enables us to upload configuration data for all values required for the anticollision protocol defined in ISO 14443-3, which are then used by the chip. Using this method, we can clone the UID of a tag and present a falsified UID to the reader, thus bypassing security mechanisms based on the UID checks.

3. EVALUATION

We evaluated the system using a popular contactless payment system, deployed in many cafeterias, and an access control system. These systems use MiFare DESFire EV1 chips, which offer cryptographic features for authentication and data confidentiality. Both systems accepted our relayed communication and successfully performed critical actions like payments or opening doors to restricted areas by holding one device near the wallet of an authorized person. This also worked if the communication was routed over the internet. The setup also allowed us to modify data while it was relayed, in order to drop or replace certain commands or inject new commands. This functionality can be used to test the resilience of NFC systems against manipulated data.

The basic delay of communicating with an NFC tag using the Android APIs is around 15 ± 6 ms, depending on the length of command and reply. Relaying the NFC commands over the network introduces some additional delays, which lead to a total delay of 65 ± 38 ms if the devices are in the same wireless network, and even larger delays if the data traverses the public internet. These delays could be reduced by using different technologies (e.g. Bluetooth).

4. LIMITATIONS

At the moment, our application is compatible with devices running Android 4.4.X. Android 5 is currently not supported, as it is incompatible with *Cydia Substrate*. Our application is also affected by some limitations of the Android system: Android does not support *extended length APDUs*, so applications using this feature will not be compatible. The UID emulation functionality does not work with every NFC chip, but has been shown to be compatible with the Broadcom BCM20793 family, as used in the Nexus 4, 5 and 7 (newer iteration), among other devices. Finally, some readers are hardened against relay attacks, for example, by using distance bounding protocols or timing checks.

5. DEMONSTRATION

For the practical demonstration, we will provide a test setup of two Android devices and an NFC card reader. We will provide cards for demonstration purposes, but will also offer conference participants the chance to test their own NFC-enabled cards or tokens. The source code of the application and precompiled Android packages will be made available for conference participants and the general public. A preview version, including a video demonstration, is available at https://seemoo.tu-darmstadt.de/nfcgate.

6. CONCLUSION

In this extended abstract, we discussed NFCGate, an NFC relaying application for Android capable of relaying arbitrary NFC protocols built on ISO 14443-3. The application successfully relayed the communication of a popular contactless payment system based on the MiFare DESFire technology, using two rooted Android phones. Relaying allows for both inspection and modification of NFC traffic, enabling easier, more in-depth security analysis of NFC systems without the need for specialized hardware.

7. ACKNOWLEDGEMENTS

This work has been supported by the DFG SFB 1053 MAKI, by LOEWE CASED and by LOEWE NICER.

8. REFERENCES

[1] L. Francis, G. P. Hancke, K. Mayes, and K. Markantonakis. Practical Relay Attack on Contactless Transactions by Using NFC Mobile Phones. *IACR Cryptology ePrint Archive*, 2011:618, 2011.

[2] G. P. Hancke. Practical attacks on proximity identification systems. In *IEEE Symposium on Security and Privacy*, 2006.

[3] E. Lee. NFC Hacking Made Easy. *Def Con 20*, 2011.

[4] L. SaurikIT. Cydia Substrate. http://www.cydiasubstrate.com/.

[5] sinpowei. NFCSpy. https://code.google.com/p/nfcspy/.