NeoCatena Networks Inc.

>> Next Generation RFID Security >>

Security Risks in RFID Applications

Lukas Grunwald
Co-Founder and CTO

Agenda

- Generic Attacks
- Transition to RFID Systems
- Breaking Encrypted RFID Tags
- Reader-Emulation, Soft-Tags
- Unsecure Designs
- Conclusion

What is RFID?

- Radio Frequency Identification (RFID)
  - Wireless transmission of information between transponder and reader without visibility
  - Bidirectional transfer (read and write)
  - Transponder (tag) can be attached, embedded or implanted
  - Automatic correlation between object and saved data

Generic Terms

- RFID is often used as generic term for complete infrastructures.
  - A transponder (aka RFID-chip, -tag, -label, wireless label or simple chip)
  - A reader (in fact most of them can write to the tag too)
  - Some middleware (aka Edge Server), sometimes on an embedded system on the reader, which connects the reader to a server
  - Some communications infrastructure
  - Some databases storing the tag information (optional)
  - Integration with server farms, data warehouses, services and supporting systems

Transponders

- There are different kinds of transponders:
  - Only transmitting a unique ID (serial-number)
    - Passive
    - Identification
    - Tracking (toll-systems)
    - Clear text communication

Transponders

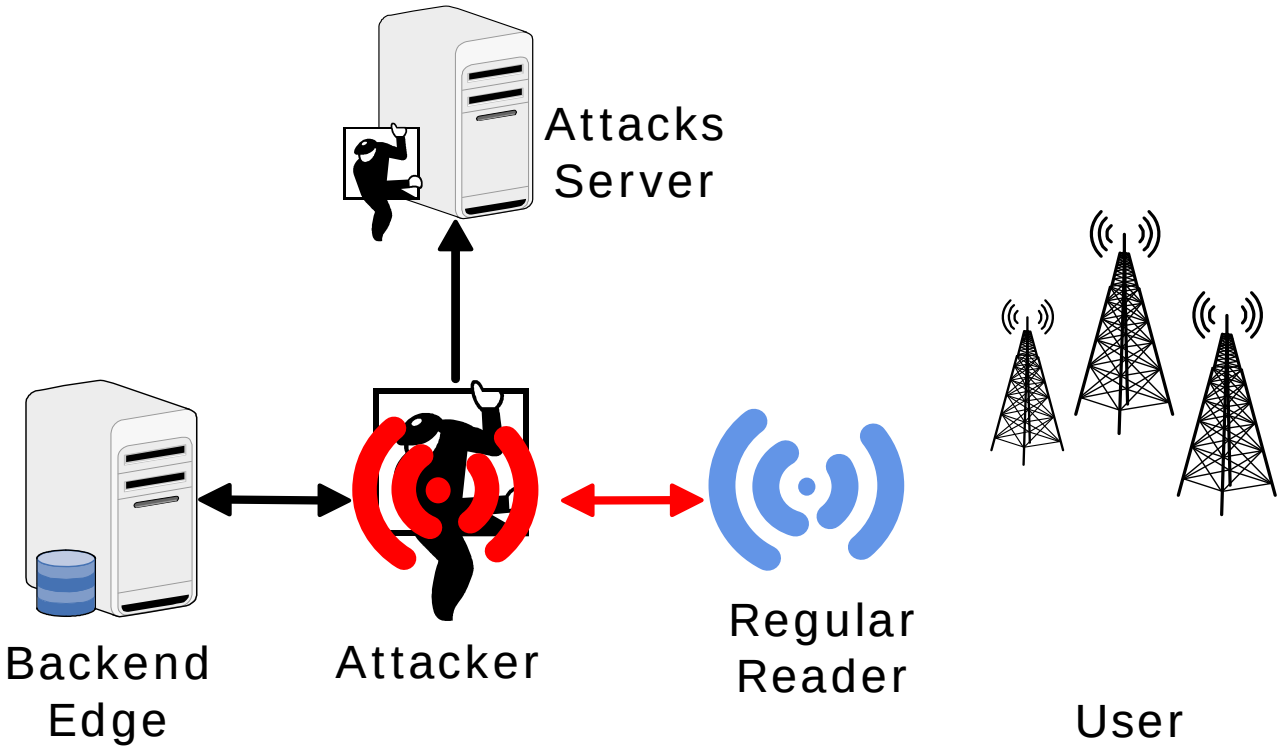
- There are different types of transponders:
  - Storage of Data / Metadata R/W WORM
    - Most passive, some active
    - EPC
    - Smart Labels
    - Most use clear text communication, some use encrypted communication

Transponders

- There are different types of transponders:
  - Act as Smart Card Interface
    - Most active, some passive
    - Biometric Passport (ICAO - MRTD)
    - Access Control System (Mifare DESFire)
    - Encryption, authentication, encrypted communication

Known Attacks against RFID Systems

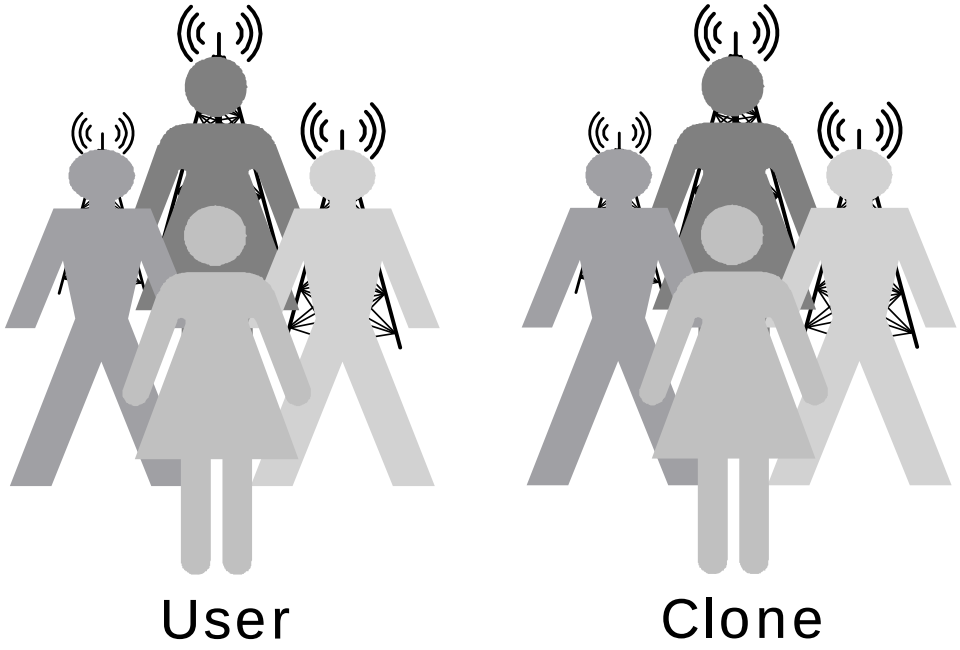
Man in the Middle



Man-in-the-Middle

- Sniffing of the communication between transponder and reader
  - Faking the communication between peers
  - Obtain UID, user data and meta data
  - Basis for subsequent attacks
  - Replay / relay attacks to fool access control systems

Cloning

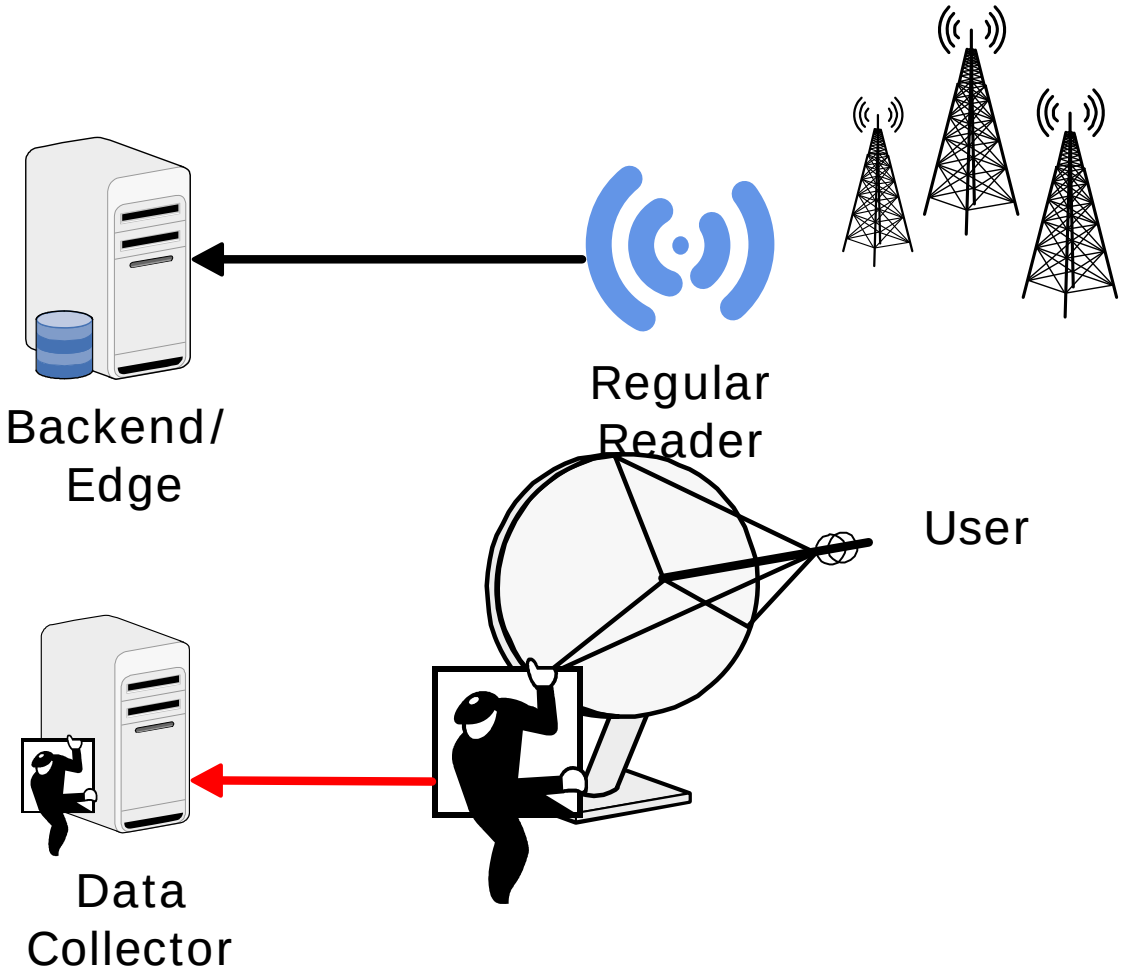


Cloning

- Duplicating or manipulating RFID tag data to create identical copies of RFID tags that will be accepted by an RFID application as valid
  - Gain illegal access to a restricted area
  - Inject counterfeit products into a digital supply chain
  - Change price tags at the Point of Sale (Cyber Shop Lifting)

Manipulation of Data

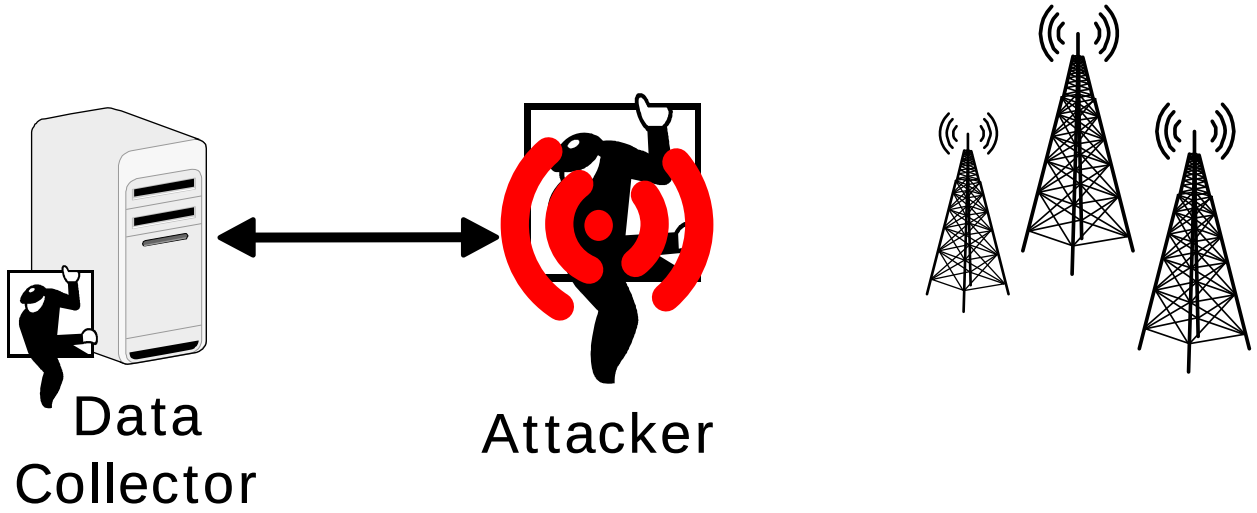
Passive Scanning



Passive Scanning

- Attacker sniffs the communication with his own antenna
- Energy for the tag is provided by legitimate reader
- Obtain
  - user-data
  - meta-data

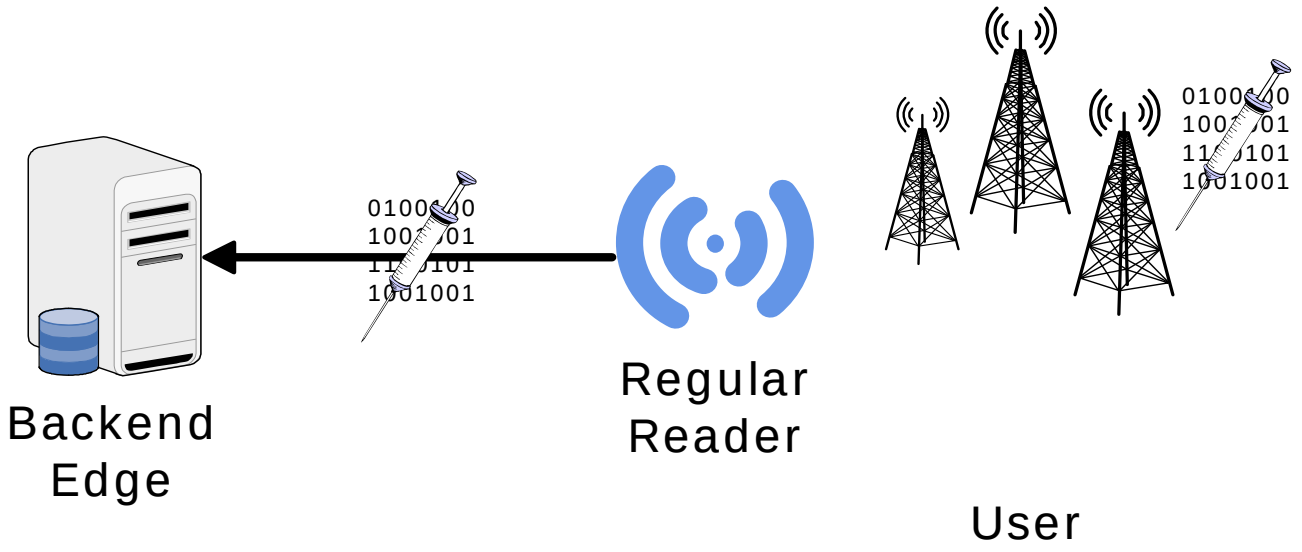
Active Scanning



Active Scanning

- Emulating a legitimate reader for unauthorized read/write operations
- Attacker uses own reader / antenna environment
- Energy for the tag is provided by attacker
  - Change of UID via manipulation of the Administrative Block
  - Forge identity
  - UID must be readable in clear text

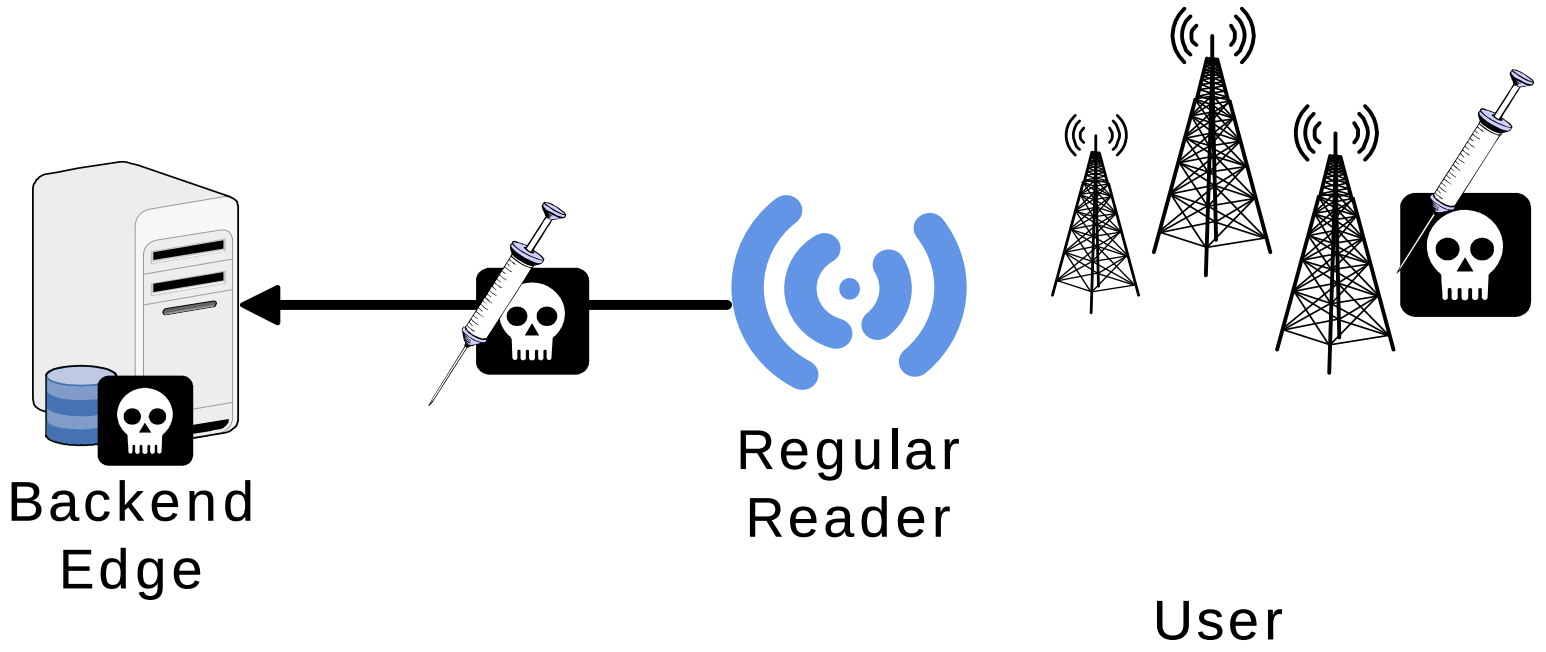
Code Injection



Code Injection

- Insertion of executable code fragments into tag data
  - SQL injection
  - Shell-Code
  - String format attack
  - Buffer overrun
- Attack edge servers, middleware and back-ends via manipulated data structures
- Non-spreading attack

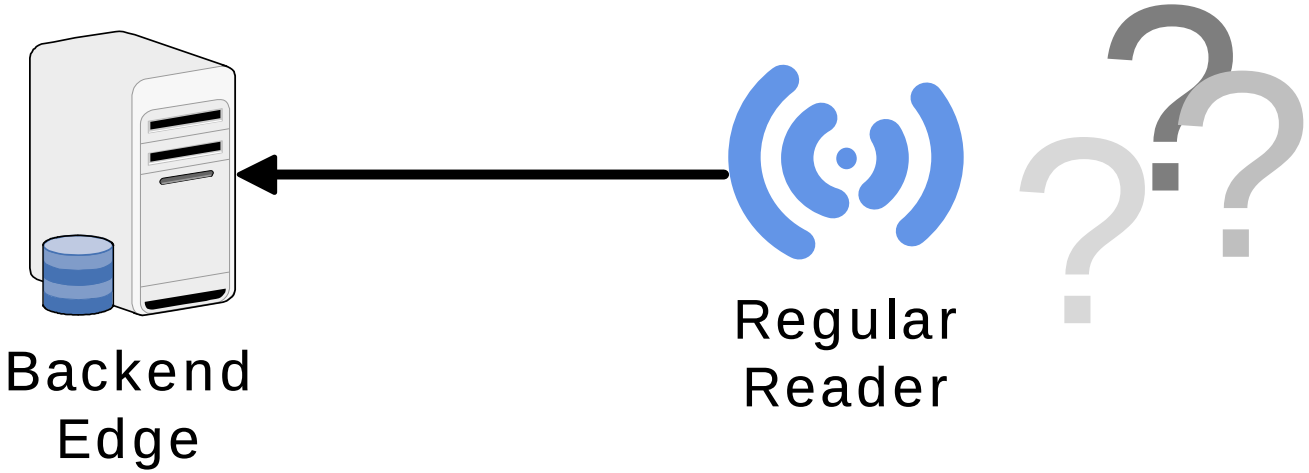
Malware Injection



Malware / Injection

- Self-replicating Malware (Code) Injection
  - Spreading attack infecting other tags, other systems
    - Database worms
    - RFID Virus
  - Denial of Service

Destruction



Destruction

- Deactivation of the transponder
  - Disable the traceability of objects
  - Disable the visibility of objects

Denial of Service (DoS)

- Jamming of the RFID frequencies
  - Use “out-of-the-box” police jammer (broadband jamming transmitter)
- Attack against anti-collision (RSA attack)
  - Prevent reading of any tags
- Shut down
  - Production
  - Sales
  - Access

Encrypted RFID

- MIFARE tags are the most used RFID transponders featuring encryption
  - Technology is owned by Philips Austria GmbH
  - Technology is based on
    - ISO 14443
    - 13.56 MHz Frequency

MIFARE Classic

- Proprietary high-level protocol
- Philips proprietary security protocol for authentication and ciphering
  - Cipher1
- MIFARE UltraLight: Same tag without encryption

MIFARE Pro, ProX, SmartMX and DESFire

- Fully comply with ISO 14443-4 standard
- The different types of tags offer memory protected by two different keys (A and B)
- Each sector can be protected with one of these keys
- DESFire featuring 3DES encryption

Physical Attacks

- MIFARE Classic
  - Broken by reverse engineering of the cipher-algorithm
  - Access to the die, layer by layer
  - Analyze photos taken with an electron microscope
  - Linear bit shift encryption
  - Not secure at all

Physical Attacks

Copyright by Karsten Nohl, reverse engineering of the Crypto 1 Cipher from the NXP MIFARE Classic Chip

MIFARE Sector Keys

- Philips puts all information under NDA
- We are not interested in signing an NDA
- Extract information from RFID software via “UNIX strings” method
- Google desktop search is very popular among smartcard developers

Sign in

Google

Web Images Groups News Froogle Maps more »

A0A1A2A3A4A5 Search Advanced Search Preferences

Search the Web Search English pages

Web Results **1** - **10** of about **18 English** pages for **A0A1A2A3A4A5**. (**0.20** seconds)

[DOC] Access7CW ACCESS 9 CM OUTPUT FORMAT DESCRIPTION Version Author ...
File Format: Microsoft Word - View as HTML
AA <CR>, authenticate with keytype A using tranportkey **A0A1A2A3A4A5** ... Authentication to sector 01 by using transportkey **A0A1A2A3A4A5** as key A ...
aut-bscw.hut.fi/pub/bscw.cgi/d6792/T00723E.doc - Supplemental Result - Similar pages

Mlfare smart card NO_TAG
Command for loadkey function is 0x4C : Where Key A = **a0a1a2a3a4a5** Key B = b0b1b2b3b4b5 : Then may be the key set 0, key set 1, and key set 2, was wrong. ...
www.epanorama.net/wwwboard/messages/4136.html - 9k - Cached - Similar pages

[PDF] ap dev data sheet
File Format: PDF/Adobe Acrobat - View as HTML
The cards do not contain access control data, but are programmed with. Philips default keys (**A0A1A2A3A4A5** & B0B1B2B3B4B5) in all sector. trailers. ...
www.hidcorp.com/pdfs/products/mifare_devloperskit.pdf - Similar pages

[PDF] standardisation group observing the following proposed opens a lot ...
File Format: PDF/Adobe Acrobat
released for public reading using the default key A: **a0a1a2a3a4a5** hex. ... key A: **a0a1a2a3a4a5** hex. Access conditions should allow reading with key A|B and ...
www.semiconductors.philips.com/acrobat/other/identification/M001824.pdf - Similar pages

[PDF] CardMan 5x21-CL Reader Developer-222s Guide
File Format: PDF/Adobe Acrobat - View as HTML
Key A: **A0A1A2A3A4A5**, Key B: B0B1B2B3B4B5. The Mifare cards supplied with the ... The public key for MAD is "**A0A1A2A3A4A5**". For complete understanding of MAD ...
www.omnikey.com/index.php?id=5&rName=RFID%20Developer%20Guide&did=5 - Similar pages

Default Keys

- Found the following default keys:
  - Key A A0 A1 A2 A3 A4 A5
  - Key A FF FF FF FF FF FF
  - Key B B0 B1 B2 B3 B4 B5
  - Key B FF FF FF FF FF FF
  - About 60 keys from example applications
  - No protection 00 00 00 00 00 00

Example Layouts

- In the datasheets and googled documentation are a lot of examples.
- These examples include different keys and tag / memory layout and data structure for:
  - Ticketing
  - Access Control
  - Online Payment

Software developers are lazy…

- Probing a couple of cards shows that more than 75% use one of these default keys!
- “It compiles, let's ship it!”
- Some programmers not only use the example layouts, they also use the example keys!

Attack the Tag

- Directory attacks are possible with default and example keys
  - Variations of the directory are always possible
- “Smart“ brute-force attack against the tag are possible
  - Never encountered a lockout or false login counter
  - A delay for a false key does not exist

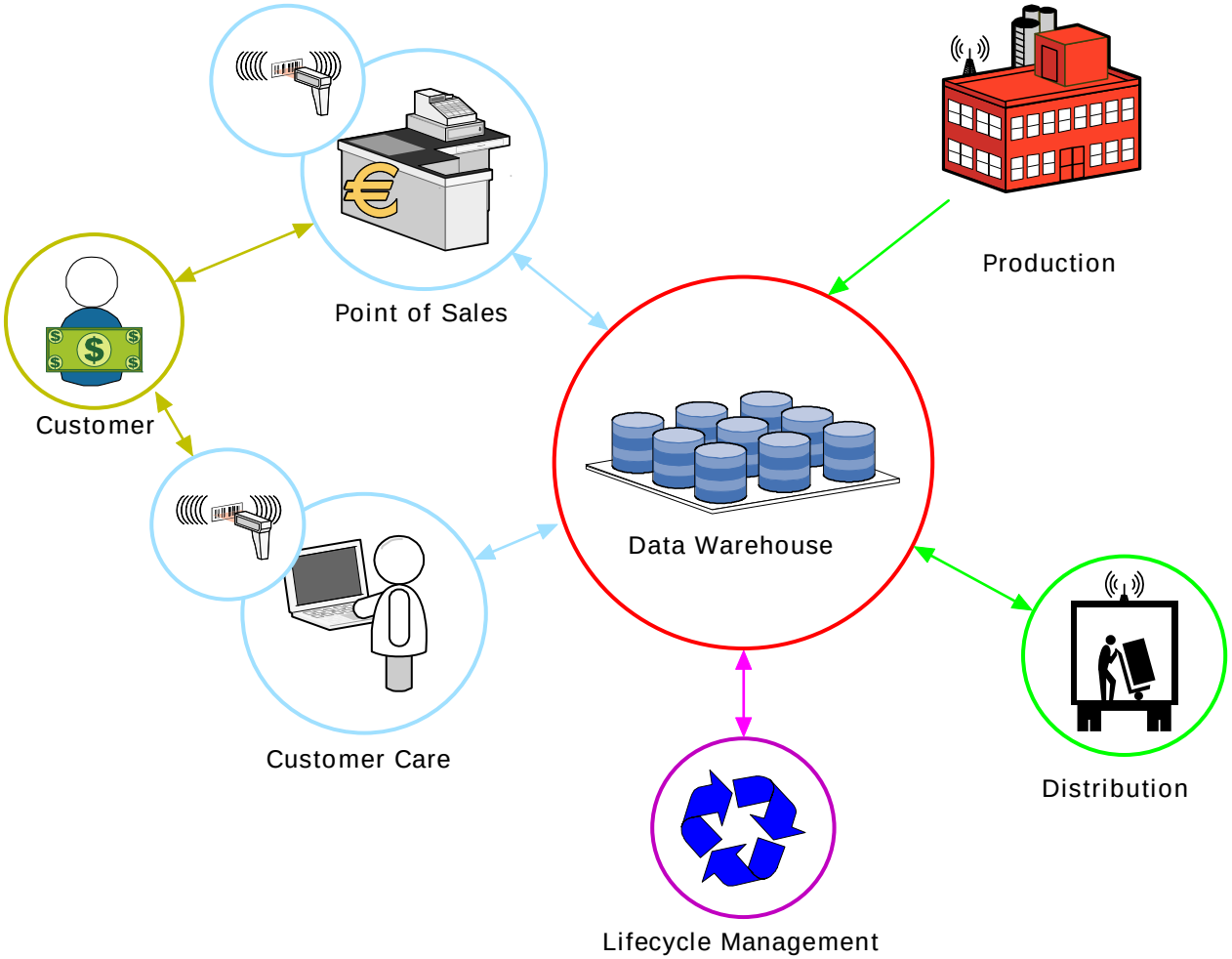
Attacks against the Backend

- The memory of an ISO 15693 tag acts like normal storage
- RFDump (Black Hat 2004) could help to manipulate data on the tag using a hex-editor-like user interface
- SQL-Injection and other malware injection attacks are possible

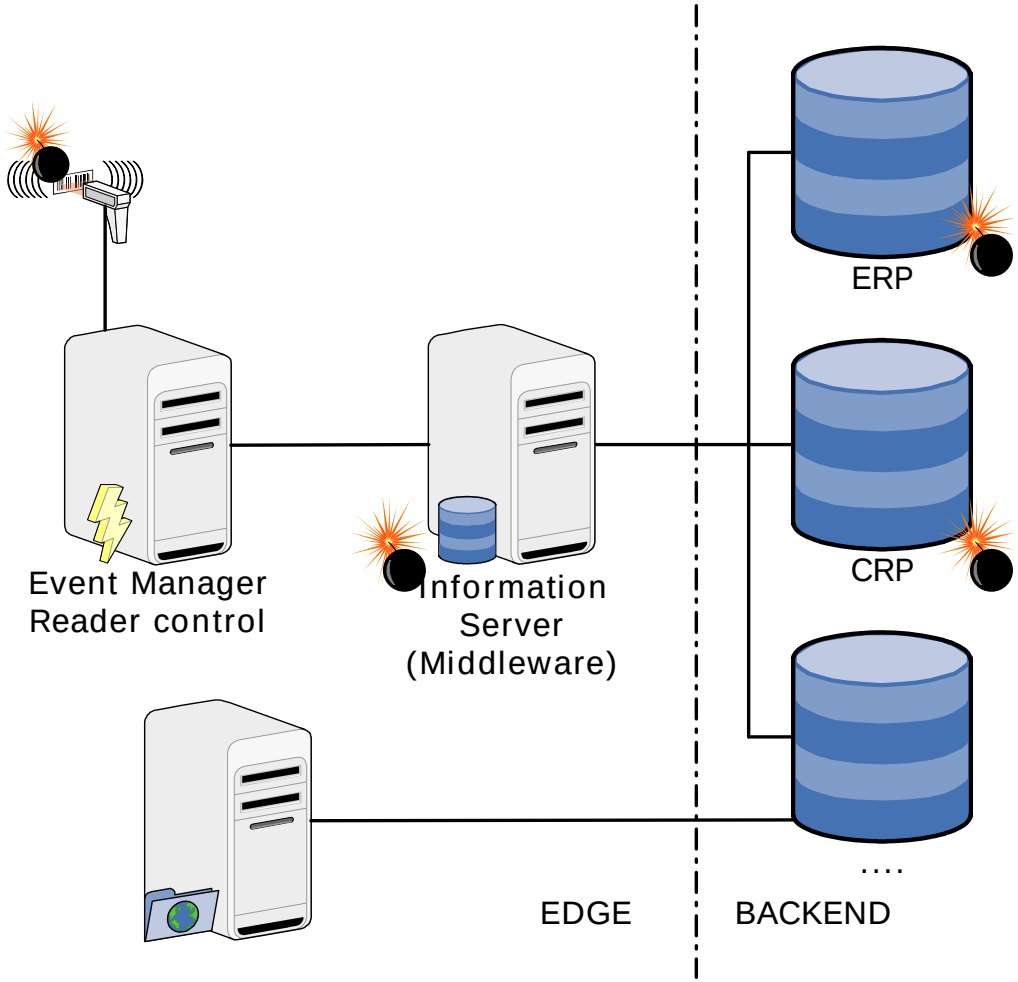
Bypassing Security Features

- If the tag is “read-only”, read it with RFDump and write the manipulated data to an empty tag with no write-protection
- Checksum, some implementations use the UID (Unique ID) as mirror block in the UD, both must be changed
- If a data block is encrypted, the Sector Key must be broken

The Digital Supply Chain



Break into the System



Problem Memory Size

Adr	Memory
0x1	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x2	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x3	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x4	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x5	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x6	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x7	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x8	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x9	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0xa	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0xb	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0xc	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0xd	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0xe	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0xf	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

Page 0x76
Byte 6

Backend Perspective

- Looks like unlimited space on the tag
  - E.g. RFDump uses a tag database to avoid reading over the boundary
- Normally reading is event-driven
  - Reading up to the EOF
  - Input is unchecked many implementations we have seen

DoS Attack with C-Strings

End of String

Adr	Memory
0x1	68547369 69202073 6e616520 6178706d 656c6f20 20662061 616d696e 75706100
0x2	FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFFF FFFFFFF FFFFFFF FFFFFFF
0x3	FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFFF FFFFFFF FFFFFFF FFFFFFF
0x4	FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFFF FFFFFFF FFFFFFF FFFFFFF
0x5	FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFFF FFFFFFF FFFFFFF FFFFFFF
0x6	FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFFF FFFFFFF FFFFFFF FFFFFFF
0x7	FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFFF FFFFFFF FFFFFFF FFFFFFF
0x8	FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFFF FFFFFFF FFFFFFF FFFFFFF
0x9	FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFFF FFFFFFF FFFFFFF FFFFFFF
0xa	FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFFF FFFFFFF FFFFFFF FFFFFFF
0xb	FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFFF FFFFFFF FFFFFFF FFFFFFF
0xc	FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFFF FFFFFFF FFFFFFF FFFFFFF
0xd	FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFFF FFFFFFF FFFFFFF FFFFFFF
0xe	FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFFF FFFFFFF FFFFFFF FFFFFFF
0xf	FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFFF FFFFFFF FFFFFFF FFFFFFF

Tag DoS with XML

Mass reading

Addr	Memory in ASCII
0x1	<rfiduid:ID>urn:epc:1:4.16.36</rfuid:ID>
0x2	<rfidcore:Observation><rfidcore:DateTime>
0x3	<rfidcore:DateTime>2002-11-06T13:04:34-06:00
0x4	</pmlcore:DateTime>
0x5	
0x6	

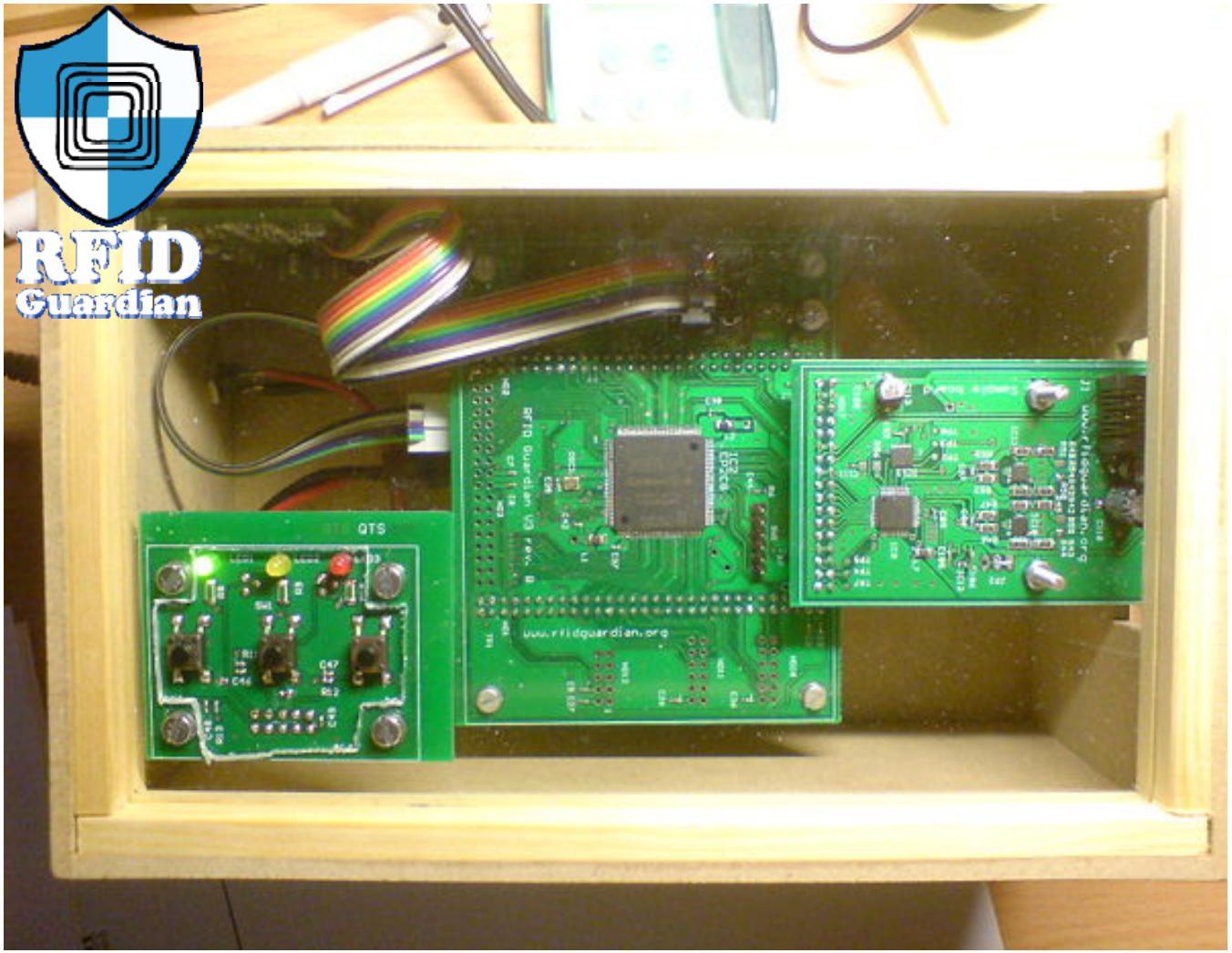
Inf. Items in one Tag

Addr	Memory in ASCII
0x1	<rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID>
0x2	<rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID>
0x3	<rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID>
0x4	<rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID>
0x5	<rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID>
0x6	<rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID><rfiduid:ID>

Soft-Tags

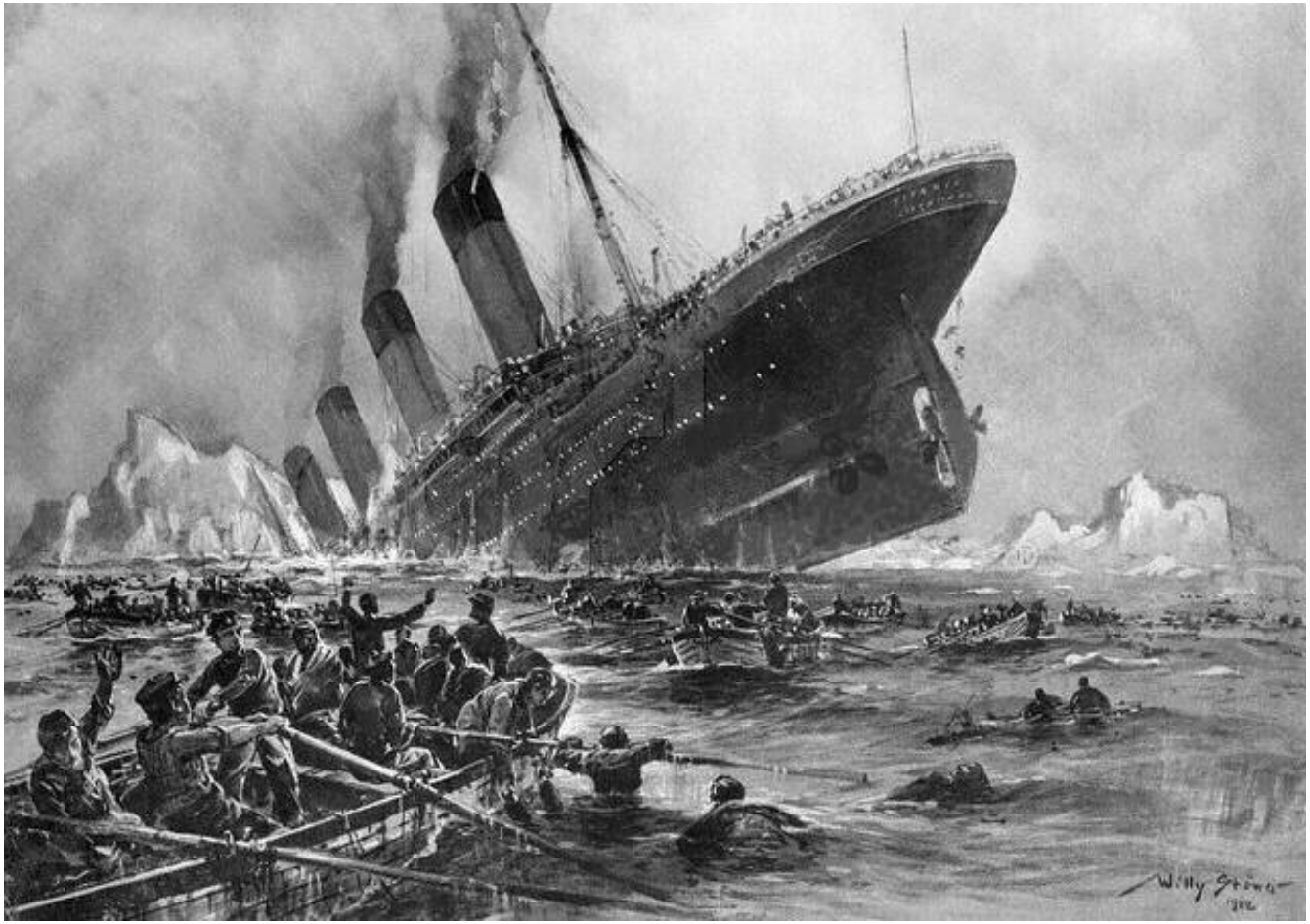
- Emulation of RFID-Tag and/or reader
- Emulation of any data and meta data
- Useful for testing backend and middleware
- Cloning of real tags
- Manipulation of any UID, User Data or Administrative Block
- Emulation of “non-standard” features
- Brute-force attacks

RFID Guardian



RFID Guardian Prof. Andrew Tanenbaum, Melanie Rieback et al.

Security Concerns



High-tech ≠ High-security

Security Concerns

1. Trust means broken chain of security
2. The whole system is secure as it's weakest link

Design Goal

- Design goal for a secure RFID system
  - Keep It Stupid and Simple
  - Do not trust input from any source
  - Verify every data on every channel
    - Is it valid?
  - Filter and log non-valid Protocol Data Units (PDUs)
  - Analyze your log and audit-trail

Thank You

Questions?

info@neocatena.com