Smart Card APDU Analysis

Black Hat Briefings 2008
Las Vegas

Ivan "e1" Buetler
ivan.buetler@csnc.ch
Compass Security AG - Switzerland
Hypothesis::Statement

SOFTWARE cannot protect SOFTWARE
Hypothesis::Situation

Attacker Toolkit: *Please choose your victim...*

**Victim 1:**
- E-Mail Contamination
- Visits to malicious Web Sites
- Second Channel Attacks

**Victim 2:**
- Phishing, Pharming
- DNS Spoofing
- Network Interception

**Victim 3:**
- Web 2.0 Hacking
- Cross Site Scripting ... Malicious Web Sites
Hypothesis::Situation

Attacker Toolkit: Please enter the attacking strategy ...

Most promising target
-> Client Computer

E-Mail
Malicious Web Site

Client Infrastructure
Network Connectivity
Server Infrastructure
Hypothesis::Situation

Client Infection Approaches

- E-Mails
- Malicious Web Sites
- Rogue Access Points (drive-by-injection)
- Exploitation of internet enabled client software
- Malicious U3, USB stick
- Malicious CD-Rom
- …. [many infection strategies – as you know]

Client Security Defense Strategies

- Latest patches / Update services
- Firewall / Personal Firewall
- Anti-Virus protection
- Spyware protection
- Device Locking Suite
- Hard disk encryption

SOFTWARE cannot protect SOFTWARE

Pentest Experience:
Success rate in client exploitation = 95%
We need Secure Devices - Tamper Proof – Trusted Minicomputers
Secure devices provide ... 
- Authentication
- Encryption
- Signatures

Secure devices are ... 
- Tamper Proof
- Virus/Trojan resistant
Smart Card Life Cycle
Smart Card::Life Cycle

Producer -> Company -> User

SmartCard Producer

MyBank

John Doe

Personalize()

Initialize()
sw + policy
MyBank::Unitialized Smart Card

Smart Card needs to be initialized before usage!

Initialization means:

a) PIN policy
b) PUK policy
c) Key generation
d) MasterKeySet

... and more...see next page
Smart Card::Life Cycle::Initialize()
During Initialization...
- Applets are configured (policy)
- Applets are loaded from computer to Smart Card
- Applets are instantiated on Smart Card

This is like “initial software package” on a Personal Computer
- The password for doing so must be known => Master Key Set!!!
Certificate Enrollment

- Generate Key on Card
- Generate CSR (certificate signing request)
- Send CSR to CA (certification authority)
- Receive Certificate from CA
- Store Certificate on Card

Smart Card is then useable

- Authentication
- Encryption
- Signatures
Smart Card::APDU

Smart Card Communication APDU

John Doe
Application Protocol Data Unit
- Communication between CSP/PKCS#10 and Smart Card
- ISO 7816 Specification + Vendor extensions
Smart Card::APDU::Architecture

Architecture

APDU

Microsoft Crypto API

CSP – Cryptographic Service Provider

CSP

Cryptoki

Pkc#10 DLL from vendor

winscard.dll

Java

Java
# Smart Card::APDU::Command

## APDU Command and Response Structure

<table>
<thead>
<tr>
<th>Command APDU</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CLA</td>
<td>INS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response APDU</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>SW1</td>
</tr>
</tbody>
</table>

## APDU Command Details

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Length</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLA</td>
<td>Class</td>
<td>1 Byte</td>
<td>Class of the command (e.g.: if a command uses secure messaging or not)</td>
</tr>
<tr>
<td>INS</td>
<td>Instruction</td>
<td>1 Byte</td>
<td>Command instruction</td>
</tr>
<tr>
<td>P1</td>
<td>Parameter 1</td>
<td>1 Byte</td>
<td>First parameter of the instruction</td>
</tr>
<tr>
<td>P2</td>
<td>Parameter 2</td>
<td>1 Byte</td>
<td>Second parameter of the instruction</td>
</tr>
<tr>
<td>Lc</td>
<td>Length command</td>
<td>0 - 3 Bytes</td>
<td>Length of the command data</td>
</tr>
<tr>
<td>Data</td>
<td>Data</td>
<td>Lc Bytes</td>
<td>Command data (apdu request)</td>
</tr>
<tr>
<td>Le</td>
<td>Length expected</td>
<td>0 - 3 Bytes</td>
<td>Length of the response data (apdu response)</td>
</tr>
</tbody>
</table>
Smart Card::APDU::Response

APDU Command and Response Structure

<table>
<thead>
<tr>
<th>Command APDU</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CLA</td>
<td>INS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response APDU</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>SW1</td>
</tr>
</tbody>
</table>

APDU Response Details

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Length</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Body</td>
<td>0 - 3 Bytes</td>
<td>Data of the response (Lc) Can be NULL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SW1</td>
<td>Status Word 1</td>
<td>1 Byte</td>
<td>Status Word 1</td>
</tr>
<tr>
<td>SW2</td>
<td>Status Word 2</td>
<td>1 Byte</td>
<td>Status Word 2</td>
</tr>
</tbody>
</table>
Smart Card::APDU::Enter PIN

Example: APDU Enter PIN

APDU Command

```
C0  20  00  01  08  3030303030303030
```

APDU Response

```
90  00
```
Smart Card::APDU::Standards

GSC-IS (Government Smart Card Interoperability Specification)
- ISO Standard (APDU)
  - 7816-4: Organization, security and commands for interchange
  - 7816-8: Commands for security operations
- Goal of GSC-IS
  - Interoperability requirements of the enterprise market

EMV - CAP
- Europay/MasterCard/Visa - Chip Authentication Program

GSM (Global System Mobile)
- GSM Standard
Smart Card::ATR::Answer to Reset

ATR String: Unique Identification for Smart Cards

- ATR (Answer to Reset) returns unique number
- Unique number references to the appropriate DLL (registry key)
Smart Card::APDU::CSP

**ATR:** HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Calais\Smart Cards

**Service Provider:** HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider
ATTACKING
Smart Card SOLUTIONS
Introduction::Smart Card Attacks

Attacking Approaches

- Host Computer (Software)
- Transmission (Link Layer)
- Internal Smart Card (Physical, Side Channel Attacks, not covered here)
Hardware::Sniffing APDU

Hardware APDU Sniffing Device
- The APDU sequences are not commonly known – hidden secret disclosure
- ATM APDU analysis
- GSM APDU analysis
Hardware::Sniffing APDU

Season2 Interface

To the SC Reader

Smart Card

RS232 Sniffing Port

To the SC Reader
Hardware::Sniffing APDU

RS232 Sniffing Port

SC Reader

Serial Port Monitor
Software::Scanning APDU Commands
Software::APDU LiveDebugger

Live Debugger
- DLL Proxy winscard.dll
- Analyzing any software that communicates with the Smart Card with winscard.dll
- Works with PKCS#10 or CSP enabled applications

Live Debugger Features
- Command Modification
- Response Modification
- Logging
**APDU Live Debugger: APDU Inspection/Interception**

- Live Debugging
- Command & Response Interception

### APDU LiveDebugger Interface

<table>
<thead>
<tr>
<th>Time</th>
<th>Handle</th>
<th>Command APDU</th>
<th>Respond APDU</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>2008.05.30</td>
<td>DCEA01</td>
<td>00A4040007A0000000000000000000000</td>
<td>611A</td>
<td>ISO/IEC 7816-6</td>
</tr>
<tr>
<td>2008.05.30</td>
<td>DCEA01</td>
<td>00A4040007A0000000000000000000000</td>
<td>612A</td>
<td>ISO/IEC 7816-6</td>
</tr>
<tr>
<td>2008.05.30</td>
<td>DCEA01</td>
<td>00A4040007A0000000000000000000000</td>
<td>611A</td>
<td>ISO/IEC 7816-6</td>
</tr>
<tr>
<td>2008.05.30</td>
<td>DCEA01</td>
<td>00A4040007A0000000000000000000000</td>
<td>612A</td>
<td>ISO/IEC 7816-6</td>
</tr>
<tr>
<td>2008.05.30</td>
<td>DCEA01</td>
<td>00A4040007A0000000000000000000000</td>
<td>611A</td>
<td>ISO/IEC 7816-6</td>
</tr>
<tr>
<td>2008.05.30</td>
<td>DCEA01</td>
<td>00A4040007A0000000000000000000000</td>
<td>612A</td>
<td>ISO/IEC 7816-6</td>
</tr>
<tr>
<td>2008.05.30</td>
<td>DCEA01</td>
<td>00A4040007A0000000000000000000000</td>
<td>611A</td>
<td>ISO/IEC 7816-6</td>
</tr>
<tr>
<td>2008.05.30</td>
<td>DCEA01</td>
<td>00A4040007A0000000000000000000000</td>
<td>612A</td>
<td>ISO/IEC 7816-6</td>
</tr>
<tr>
<td>2008.05.30</td>
<td>DCEA01</td>
<td>00A4040007A0000000000000000000000</td>
<td>611A</td>
<td>ISO/IEC 7816-6</td>
</tr>
<tr>
<td>2008.05.30</td>
<td>DCEA01</td>
<td>00A4040007A0000000000000000000000</td>
<td>612A</td>
<td>ISO/IEC 7816-6</td>
</tr>
</tbody>
</table>

**Edit APDU Command**  
**Edit APDU Response**
APDU LiveDebugger Discovery!
APDU::LiveDebugger::Results

APDU Control Sequences

80 XX XX XX Not encrypted (Axalto Commands)
84 XX XX XX Encrypted
C0 XX XX XX Not encrypted
00 XX XX XX ISO Standard APDU

APDU Instructions

XX B0 XX XX Read
XX D6 XX XX Write
C0 D2 XX XX Generate keys on Smart Card
C0 12 XX XX Generate keys on PC
XX A4 XX XX Select Instance
C0 12: Generate Keys on Computer (not on Smart Card)

- First: Offcard key generation
- Then: Storing keys onto the Smart Card

<table>
<thead>
<tr>
<th>Code</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C0 12</td>
<td>00 00 00 02 02 00 30</td>
<td>[Cyberflex C0] Create PrivateKeyFile: Creates the private portion of a public key file</td>
</tr>
<tr>
<td>00 C0 00 00 02</td>
<td>11 A3 90 00</td>
<td>[Opencard] Get residual data (2 Bytes)</td>
</tr>
<tr>
<td>C0 D6 00 0D 02 58 11</td>
<td>90 00</td>
<td>[Cyberflex C0] Update binary</td>
</tr>
<tr>
<td>C0 80 00 4B 04</td>
<td>58 11 00 00 90 00</td>
<td>[Cyberflex C0] Read Binary</td>
</tr>
<tr>
<td>C0 D6 00 4B 02 44 11</td>
<td>90 00</td>
<td>[Cyberflex C0] Update binary</td>
</tr>
<tr>
<td>C0 D6 11 6F 14 14 00 00 00 00 00 00 00 00 0...</td>
<td>90 00</td>
<td>[Cyberflex C0] Update binary</td>
</tr>
<tr>
<td>C0 D6 00 0F 06 AD BB 85 11 11 00</td>
<td>90 00</td>
<td>[Cyberflex C0] Update binary</td>
</tr>
<tr>
<td>C0 D6 00 01 01 01</td>
<td>90 00</td>
<td>[Cyberflex C0] Update binary</td>
</tr>
<tr>
<td>C0 D6 11 92 11 00 00 00 00 00 00 00 00 0...</td>
<td>90 00</td>
<td>[Cyberflex C0] Update binary</td>
</tr>
<tr>
<td>C0 D6 00 0F 06 AD BB 85 11 11 00</td>
<td>90 00</td>
<td>[Cyberflex C0] Update binary</td>
</tr>
<tr>
<td>C0 B0 00 4B 04</td>
<td>44 11 00 00 90 00</td>
<td>[Cyberflex C0] Read Binary</td>
</tr>
<tr>
<td>C0 D6 00 4B 02 17 11</td>
<td>90 00</td>
<td>[Cyberflex C0] Update binary</td>
</tr>
<tr>
<td>C0 D6 11 62 2D 20 00 01 34 7C 33 35 7C 36 ...</td>
<td>90 00</td>
<td>[Cyberflex C0] Update binary</td>
</tr>
<tr>
<td>C0 D6 00 15 06 35 D4 58 11 2A 00</td>
<td>90 00</td>
<td>[Cyberflex C0] Update binary</td>
</tr>
</tbody>
</table>
## C0 D2: Generate Keys on Card

- First: Oncard key generation
- Then: Smart Card generates keys on card

<table>
<thead>
<tr>
<th>APDU</th>
<th>LiveDebugger::Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>C0 D2 03 00 04 00 01 00 01</td>
<td>61 84</td>
</tr>
<tr>
<td>C0 00 00 80</td>
<td>EB 37 E3 97 F2 7A...</td>
</tr>
<tr>
<td>C0 B0 05 4C 04</td>
<td>07 00 49 05 90 00</td>
</tr>
<tr>
<td>C0 B0 05 53 04</td>
<td>83 00 A4 04 90 00</td>
</tr>
<tr>
<td>C0 D6 05 4C 04 07 00 A4 04</td>
<td>90 00</td>
</tr>
<tr>
<td>C0 D6 05 53 83 83 00 01 85 53 00 15 E2 68 3...</td>
<td>90 00</td>
</tr>
<tr>
<td>C0 D6 00 15 06 78 F8 49 05 80 00</td>
<td>90 00</td>
</tr>
<tr>
<td>C0 D6 05 D9 12 00 47 00 00 03 00 00 00 00 0...</td>
<td>90 00</td>
</tr>
<tr>
<td>C0 D6 00 DF 06 43 E6 CC 05 12 00</td>
<td>90 00</td>
</tr>
<tr>
<td>C0 B0 05 4C 04</td>
<td>07 00 A4 04 90 00</td>
</tr>
<tr>
<td>C0 D6 00 0B 02 A4 04</td>
<td>90 00</td>
</tr>
<tr>
<td>C0 D6 05 4C 07 07 00 01 01 00 01 00</td>
<td>90 00</td>
</tr>
<tr>
<td>C0 D6 00 1B 06 1C 0A 42 05 04 00</td>
<td>90 00</td>
</tr>
</tbody>
</table>

- [Cyberflex C00] Generate RSAKey: Generation of a public key and a private key CRT
- [OpenCard] Get residual data (128 Bytes)
- [Cyberflex C00] Read Binary
- [Cyberflex C00] Read Binary
- [Cyberflex C00] Update binary
- [Cyberflex C00] Update binary
- [Cyberflex C00] Update binary
- [Cyberflex C00] Update binary
- [Cyberflex C00] Read Binary
- [Cyberflex C00] Update binary
- [Cyberflex C00] Update binary
- [Cyberflex C00] Update binary
The flag "Generate Keys on Card" is not enforced

This results in the following attack vector

- The CSP asks the card for oncard, or offcard key generation because the card itself knows the status
- The APDU interceptor responds: "I am an offcard keygen Smart Card"
- The CSP will then perform the generate key functions on the computer
- The CSP will send the CSR to the CA
- After all, the certificate and key material will be stored onto the Smart Card
- The hacker who did all the man in the middle stuff "knows" all the keying and certificate details! **Trust is lost!**
PoC
Smart Card APDU Attack
Smart Card Man-in-the-Middle Attack

SmartCard

APDU Debugger

CSP Proxy

Browser

Certification Authority

Attacker Host

Client Cert Enrollment

GenerateKey

GenerateKeys

Generate CSR

SendCSR

Create Certificate

Client Certificate

Store to SmartCard

Store to SmartCard

Trust is Lost! Send details to Attacker

Flag: Generate Key on Card?()
Smart Card Man-in-the-Middle Attack

Conclusion

- The use of Smart Cards does not make you independent from the host computer in any case and situation!

- The flag „Generate Keys on Card“ does still allow key material being stored onto the Smart Card.

- This demonstration was solely related to Smart Cards an end-user has. If the attacker has some sort of virus/trojan running where the Smart Cards are initialized, even more fraud can occur (MasterKeySet attacks, Rogue Applet Uploads, ...)

- The PIN has been seen in plain-text within the memory segment of the Smart Card software. The PIN can be gathered without administrative privileges. By knowing the PIN, the Smart Card could be used behind the scenes without the users knowledge (signing, encryption).
Thank you!

Questions?
  ✤ ivan.buetler@csnc.ch

See you at the Swiss Cyber Storm II – Switzerland - 2009
  ✤ www.hacking-lab.com