Mobile Security Evolution

From Fixed Defenses to Defense in Depth

Stefan Andersson
Sony Mobile Communications

© Sony Mobile Communications
Content

• A brief history of mobile device security
  • From closed to open systems
• Scaling up to meet the new threat
  • Building Security into the Development Lifecycle
• Security Development Lifecycle Activities
  • Threat modeling, static code analysis, fuzz testing …..
• Security Development Lifecycle Roll-out
  • Adapting the SDL to multiple targets
A brief history of mobile device security
From closed to open systems
Well known assets, the IMEI

IMEI: 00-123456-123456

GSM/UMTS/LTE Network
IMEI Blacklist
Well known assets, SIMLock

Carrier A
SIM

Carrier B
SIM

SIMLocked to carrier A
Well known threats

SIMLock example:

- Eco-system value
  - Volume x Subsidy x Pre-paid ratio x Hack ratio
  - 50 M units x 100 EUR/unit x 40% x 5 % = 100M EUR
Well known attack surface
The game shift

Incidents

Q1-08  Q1-09  Q1-10  Q1-11  Q1-12  Q1-13
Scaling up to meet the new threat

Building Security into the Development Lifecycle
Others faced this problem before us

Microsoft®
Security Development Lifecycle

SAFECode
Software Assurance Forum for Excellence in Code
Driving Security and Integrity

BSIMM
The Software Security Group

Feedback from reality

Strategy and expertise  
SSG Line Work

Define activities  
SSG Project Work

Roll out to targets  
SSG in dev. projects

Maintain and mentor
Security Development Lifecycle Activities

Threat modeling, static code analysis, fuzz testing ….
Threat modeling

- Does the application expose any API?
- What happens if an attacker modifies the parameters?

MyDataCloud Server

Protocol

MyData Cloud

• Does the server send any sensitive data?
• What happens if an attacker modifies the data?

/\data/MyDataCloud/data.dat

Application

Data

• Does the application read any sensitive data?
• What happens if an attacker modifies the data?

/SD Card/data.dat

Linux User:

MyDataCloud
Static code analysis

- Coding guidelines
- Tool rules
- Automated Code Review
Fuzz testing

My Application

- Protocol parser
- IPC Receiver
- Pipe/IOCTL listener
- File parser

App. core

Shared memory

/data/file.txt

Fuzz data/file.txt

Fuzzer
Incident management, patch roll out

Bring up
(develop patch)

Certification & Approval

Android™ source code available

Software upgrade released to consumer
Training

- Gold
- Silver
- Bronze
- Introduction
- Developers & Architects
- Managers
3rd party suppliers

Risk level

Low risk applications
Incident management Mandatory

High risk applications
Partner SDL Mandatory
Security Development Lifecycle Roll-out

Adapting the SDL to multiple targets
Adapting the SDL to dev. teams

**SDL Main**
- Threat Modeling
  - Agile
    - Android threats
  - Core
    - Waterfall
      - Linux threats
  - Backend
    - Agile
      - Server threats

**Static code analysis**
- Java

**Fuzz testing**
- Java API fuzzer

**Incident management**
- Map incident severity to apps. release cycle
- Map incident severity to product SW release cycle
- Map incident severity to server SW release cycle
Defence in depth

- Apps teams
- Core teams
- Server Teams
- 3rd parties
- SSG
Thank you for listening!
Keeping the enthusiasts happy

Other OS

Bootloader unlock
Incident Management Ecosystem

• Google patch policy:
  • Limited back porting
  • Conformance Test Suite (CTS)

• Android security team
  security@android.com

• Community contact list
  android-partner-security@googlegroups.com
Factors that Boost Security

- Patch Roll out speed
- Brand Attraction
- Development Environment
- Environment

Factors that Decrease Security

- Patch Roll out speed
- Brand Attraction
- Development Environment
- Environment

What happened?