Enhancing Security of Linux-based Android Devices

Aubrey-Derrick Schmidt, Hans-Gunther Schmidt, Jan Clausen, Kamer Ali Yüksel, Osman Kiraz, Ahmet Camtepe, and Sahin Albayrak

This work was funded by Deutsche Telekom Laboratories
- Research Institute with ~100 employees
- Six core departments:
  - Agent Core Technologies
  - Next Generation Services
  - Information Retrieval
  - Cognitive Architectures
  - Education
  - Security
DAI-Labor Security Department

- Works on:
  - Smartphone Security
  - Agent Security
  - Network Security Simulation
  - Critical Infrastructures
  - PKI / Cryptography
  - Next Generation Homes - Security
TOC

- Motivation
- Android Security
- Adding Linux Security Tools to Android
- Enhancing Security with self-built IDS
Motivation

- Smartphones getting increasingly popular
- Various smartphone malwares appeared
- Signature-based approaches only efficient for “known” malware
- Anti-Virus engines need avg. time of 48 days to get capable of detecting new malware [Oberheide08]
- More than 700,000 can be infected via MMS in about three hours [Bulygin07]
Motivation

- Android already very popular (Java on Linux)
- Android sources will be set open-source
  - Opportunity to develop low-level security tools for commonly used smartphones the first time
- Linux security research is mature
  - A lot lessons learned
  - A lot of open source tools available
TOC

- Motivation

- Android Security

- Adding Linux Security Tools to Android

- Enhancing Security with self-built IDS
Android Security

- Images on emulator
  - System Image (YAFFS2, 65 MB / 21 MB free)
    - Mounted to /system
    - OS files, libraries, drivers, system bins
    - Android config files
    - Android framework
    - Android base applications (e.g. Browser)
    - +R(W)X
Android Security

- Images on emulator
  - Userdata Image (YAFFS2, 65 MB / 40 MB free)
    - Mounted to /data
    - Used for applications, user data, DRM, ...
    - +RWX
  - Cache Image (YAFFS2, usage not specified yet)
  - SD-Card Image (no “obvious” size limitations)
    - Mounted to /sdcard
    - Files created as user and group “system”
    - +RW
Applications are “location-aware”

- Can only be executed in /data or /system
- Any changes on file permissions succeed there
- Changes in e.g. /sdcard do not succeed (e.g. set execute bit)
- Most probably, (Linux) applications cannot be started via SD-Card
Android Security

- (Java) Application signing is required
  - Linux state not clear
  - developer signs his application with own certificate at the moment
  - System might change to something similar to Symbian OS
    - Central authority for assigning certificates
    - Limited access to APIs
    - Each, Google and T-Mobile announced application store (might include application testing and verification)
Android Security

- **File rights:**
  - `/data/data/<package.application_name>`
  - “application land”
  - `drwxr-xr-x app_14 app_14 2008-09-17 14:26 com.android.sample`

- **Application can access other application directories signed with identical certificates**
  - “Certification land”
TOC

- Motivation
- Android Security
  - Adding Linux Security Tools to Android
  - Enhancing Security with self-built IDS
Adding Linux Security Tools to Android

General Information

- Emulator is used as basis
- OHA/Google modified a lot of standard libraries and binaries
  - Reason: opportunity for business costumers to claim “intellectual property”
- Application space is limited (~40 MB)
- Common security tools were tested
  - But: special build environment needed
Creating a Build Environment for Android

- Ubuntu 8.04
- Two toolkits can be used
  - Sourcery cross-compile toolchain
  - Scratchbox cross-compilation toolkit
    - Emulated ARM environment
    - “Common” Linux file system layout
Creating a Build Environment for Android

Important Facts

- Files are located in:
  - System files are placed in `/system`
  - Binaries in `/system/bin`
  - Libraries in `/system/lib`
  - Config files in `/system/etc`

- System configuration in OpenBinder

- Page alignment causes changes in linking

- Only way to get available applications run is compiling them statically
Adding Tools

- “Top 100 Network Security Tools” [Insec06]
- Tested from 5 main categories:
  - Anti-Virus: ClamAV
  - Firewall: iptables
  - Rootkit Detectors: chkrootkit
  - Intrusion Detection: Snort
  - Other useful tools: Busybox, Bash, OpenSSH, strace, Nmap
Anti-Virus: ClamAV

- Android Compatibility: Works
- Problems, solutions, and size:
  - Static compilation (linking) required
  - Dependent on static compiled version of "zlib" (zlib-1.2.3)
  - Total size of all ClamAV relevant files (approx. 28MB) exceeds available size in System image
  - (21MB). ClamAV virus signature database needs to be placed in a different location.
  - Size (approx.): 11140 KB libraries and binaries (/opt), 17324 KB database (/data)
Anti-Virus: ClamAV Results

---------- SCAN SUMMARY ----------
Known viruses: 407205
Engine version: 0.94
Scanned directories: 0
Scanned files: 106
Infected files: 0
Data scanned: 5.12 MB
Time: 107.236 sec (1 m 47 s)
#
Firewall: iptables

- Problems:
  - Kernel needs to be recompiled from source. Sources can be freely downloaded from Android Project website. Enable NETFILTER in kernel configuration and recompile!
  - “iptables” cannot be compiled due to linker issues: It requires statically compiled parts of libc which Android does not provide.
Rootkit Detector: Chkrootkit

- Android Compatibility: Works with minor dependencies

- Problems, solutions, and size:
  - Static compilation (linking) required
  - Requires "netstat" (provided by "busybox")
  - Requires standard directories (/lib, /etc, etc.) provided by symbolic links pointing to the correct Android directories
  - Size (approx.): 588 KB
Rootkit Detector: Chkrootkit Results

# ./chkrootkit
[: gid: unknown operand
ROOTDIR is `/'
Checking `amd'... not found
Checking `basename'... INFECTED
Checking `biff'... not found
Checking `cron'... not infected
Checking `echo'... INFECTED
Checking `egrep'... not infected
Checking `env'... INFECTED
Checking `find'... not infected
Searching for common ssh-scanners default files... nothing found
Searching for suspect PHP files... find: /var/tmp: No such file or directory
nothing found
Searching for anomalies in shell history files... nothing found
chkproc: Warning: Possible LKM Trojan installed
chkdirs: Warning: Possible LKM Trojan installed
Checking `sniffer'... ./chkrootkit: ./ifpromisc: not found
Intrusion Detection: Snort

- Problems:
  - Dependencies to libpcap, libdnet, libnet, pcre and iptables (all as statically compiled/linked solutions)
  - Requires statically compiled/linked libc parts which are not available on Android
Other Useful Tools: Busybox, Bash, OpenSSH, strace, Nmap

- Busybox: works
- Bash: works
- OpenSSH: Can be executed but is not fully functional (requires users that do not exist in the android environment)
- strace: works
- Nmap: works with minor dependencies
TOC

- Motivation
- Android Security
- Adding Linux Security Tools to Android
- Enhancing Security with self-built IDS
Enhancing Security with a Self-built Intrusion Detection System
Detecting Intrusions and Malware Overview
Detecting Intrusions and Malware
Static Function Call Approach

- Planned to present metric for weighing suspiciousness of function/system calls
- Solution far more easier on Android
- Simple decision tree can achieve 95% detection rate
  - Tested with Linux malware
    - Some of them were recompiled for Android, but only minor differences
- Still has to be tested on real device!
Detecting Intrusions and Malware
Static Function Decision Tree

__bss_start = y
| gethostbyname = y
|   | sigaction = y: normal
|   | sigaction = n: malicious
| gethostbyname = n
| fork = y
|   | strerror = y
|   |   | getgrgid = y: malicious
|   |   | getgrgid = n: normal
|   | strerror = n: malicious
|   | fork = n: normal

continued on the right side

__bss_start = n
| printf = y: malicious
| printf = n
| fprintf = y: malicious
| fprintf = n
| execv = y: malicious
| execv = n
| memmove = y: malicious
| memmove = n
| perror = y: malicious
| perror = n: malicious

... continued
Summary

- Android Security
- How to enhance security
  - Add Linux security tools
  - Light weight IDS
References


Thank you for your patience!

Q&A
Contact

Dipl.-Inf. Aubrey-Derrick Schmidt
Researcher
+49 (0) 30 / 314 – 74 039
+49 (0) 30 / 314 – 74 003
aubrey.schmidt@dai-labor.de

Hans-Gunther Schmidt
Student Researcher
+49 (0) 30 / 314 – 74 041
+49 (0) 30 / 314 – 74 003
hans-gunther.schmidt@dai-labor.de