Einsatz von CyanogenMod

Android selbst gemacht

Ein aus dem Quellcode erstelltes Android bietet die größtmögliche Kontrolle über Ihr Mobilgerät und ermöglicht im Zusammenspiel mit CyanogenMod einen wirksamen Datenschutz. **Von Martin Gossen**

AUTOR

Martin Gossen
ist als IT-Berater bei der IKS GmbH tätig und befasst sich unter anderem mit Themen der Sicherheit im Bereich Web und Mobile.
▸ m.gossen@iks-gmbh.com

Inhalt
Erstellung einer sicheren Android-Variante mit CyanogenMod.

Ressourcen
Windows 7 Professional SP 1, 64 Bit, 8 GByte RAM, Samsung Galaxy S3 (GT-I9300).

Smartphones und Tablets werden vom Hersteller grundsätzlich in einem vorkonfigurierten Zustand ausgeliefert. Es liegt auf der Hand, dass diverse Erweiterungen, Apps und Einstellungen vor allem im Interesse des Herstellers (des Geräts oder Betriebssystems) liegen – sei es zur Kundenbindung oder zur Verwertung Ihrer persönlichen Daten. Dem können Sie aber entgegenwirken.

Obwohl es keinen hundertprozentigen Schutz gegen Angriffe von Hackern gibt, können Sie das Risiko des Datenmissbrauchs erheblich verringern, indem Sie Ihr Mobilgerät von Grund auf selbst installieren und mit Sicherheitslösungen aus dem Open-Source-Bereich versorgen. In der Summe lässt sich so ein Smartphone oder Tablet erstellen, das gegen die Sammelwut der Hersteller und gegen gewöhnliche Datendiebe hinreichend gesichert ist.

Ein solches Gerät lässt sich auch innerhalb von Firmen funktional so weit einschränken, dass es einen genau definierten Anwendungszweck erfüllt, denn neue Apps sind nur mit tiefgehenden Kenntnissen zu installieren, und ein firmeneigenes Branding ist mit wenig zusätzlichem Aufwand realisiert.

Dieser Artikel beschreibt exemplarisch, wie Sie das Samsung Galaxy S3 mit dem beliebten CyanogenMod (CM) ohne Play Store, Google-Apps und Samsung-Apps aufsetzen, wie Sie CM anpassen und aus dem Quellcode kompilieren können, und was Sie bei der Konfiguration des Geräts beachten sollten. Außerdem werden Apps vorgestellt, die in puncto Sicherheit besonders empfehlenswert sind.

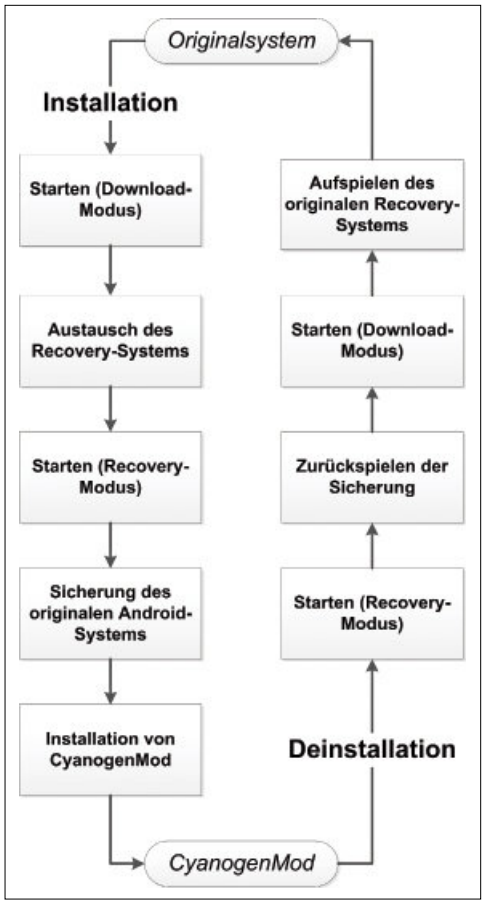
Ein wichtiger Hinweis vorab: Die Eingriffe erfolgen auf eigene Gefahr. Wenn Sie Ihr Gerät wie hier beschrieben verändern, könnte es unbrauchbar werden, beispielsweise wenn Installationsdateien defekt sind oder das USB-Kabel unzuverlässig ist, oder wenn das Kabel schlicht während der Installation aus dem Anschluss rutscht.

Auch verlieren Sie in jedem Fall die Herstellergarantie – idealerweise benutzen Sie daher nur Geräte, deren Garantie bereits abgelaufen ist.

BEGRIFFSERKLÄRUNGEN

Einige Begriffe aus dem Umfeld der Mobilgeräte-Technologie sind historisch bedingt.

ROM und Firmware bedeuten bei heutigen Smartphones und Tablets nichts anderes als Betriebssystem. Ein Stock ROM ist ein vom Hersteller ausgeliefertes, ein Custom ROM ein unabhängig vom Hersteller erstelltes Betriebssystem. Flashen bezeichnet den Installationsvorgang.



Die Vorgehensweise beim Austausch des Betriebssystems (Bild 1)

Davon abgesehen ist die Prozedur aber beim Galaxy S3 gut erprobt und Sie können in aller Regel auch den Originalzustand wieder herstellen. Hilfe bei Problemen finden Sie zum Beispiel auf der CM-Website oder bei XDA Developers.

Vorbereitung

Im ersten Schritt installieren Sie CM von einem fertig erstellten Image, da dies einen späteren Schritt erleichtert, bei dem proprietäre Dateien aus dem Gerät extrahiert werden müssen. CM kann auch mit Hilfe einer App installiert werden (CM-Installer, http://beta.download.cyanogenmod.org/install), was den Installationsprozess wesentlich vereinfacht – allerdings werden hierbei zwangsläufig Google-Apps mitinstalliert, was für unseren Zweck nicht akzeptabel ist. Laden Sie daher zunächst die für die Instal-

lation beziehungsweise Deinstallation nötigen Dateien herunter und kontrollieren Sie – soweit vorhanden – die MD5-Prüfsummen (Tabelle 1).

Installieren Sie die USB-Treiber auf Ihrem PC und kopieren Sie das Image *ClockWorkMod Recovery* in das Heimdall-Verzeichnis. Kopieren Sie außerdem das CM-Paket per USB-Kabel auf die (interne oder externe) SD-Karte des Smartphones (nochmal die Prüfsumme kontrollieren).

Wenn Sie später keinerlei App-Store installieren wollen, müssen Sie Ihre Wunsch-Apps per apk (Android Application Package) installieren. Laden Sie daher die *.apk*-Dateien (aus vertrauenswürdigen Quellen) herunter. Alternativ können Sie die Apps vorab auf einem laufenden Gerät installieren und die *.apk*-Datei aus dem App-Verzeichnis */data/app* herauskopieren.

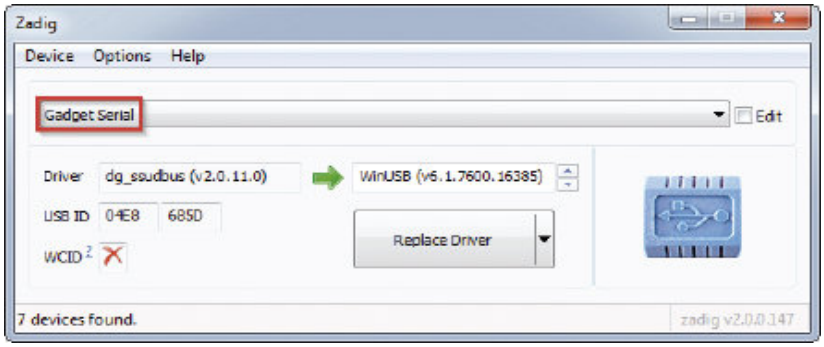
Im Folgenden sollten Sie ein USB-Kabel guter Qualität benutzen und dies direkt am Mainboard des PC anschließen, nicht an einem USB-Hub. Bewegen Sie das Gerät während der Installation möglichst nicht; stellen Sie insbesondere sicher, dass das USB-Kabel nicht getrennt wird. Laden Sie außerdem den Akku mindestens zur Hälfte auf.

Während der Installation werden diverse Daten auf der internen SD-Karte gelöscht. Sichern Sie daher alle noch benötigten Dateien, am besten auch von einer eventuell vorhandenen externen Karte. Danach kann die eigentliche Arbeit beginnen (Bild 1).

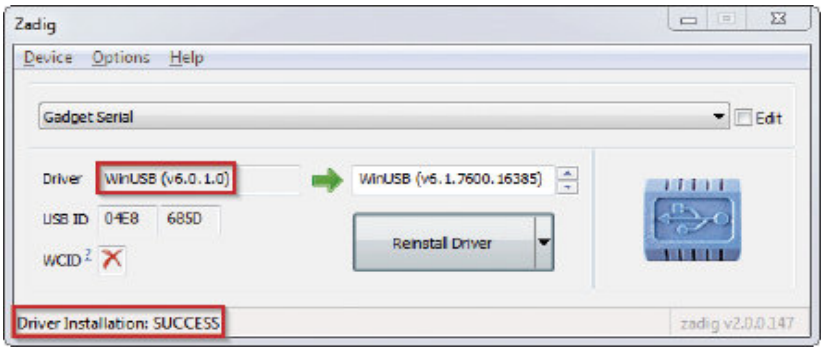
Wechsel in den Download-Modus

Zunächst muss das Gerät im Download-Modus gestartet werden, damit das Recovery-System ersetzt werden kann. Wenn das Smartphone per USB mit dem PC verbunden ist, trennen Sie das USB-Kabel. Drücken und halten Sie am Smartphone gleichzeitig die Tasten *Leiser*, *Home* und *Power* und bestätigen Sie den Dialog durch Drücken der Taste *Lauter*.

Das vorinstallierte Stock Recovery erlaubt keine Installation herstellerfremder Betriebssysteme. Daher muss das Recovery durch das flexiblere ClockWorkMod ausgetauscht werden. Schließen Sie zunächst das USB-Kabel an. Starten Sie das Heimdall-GUI auf Ihrem PC, indem Sie */Heimdall Suite/Drivers/zadig.exe* ausführen. Wählen Sie *Options, List all devices* und wählen Sie *Gadget Serial* aus (Bild 2). Fehlt dieser Eintrag, wählen Sie *Samsung USB Composite Device*, *Device Name* oder *MSM8x60*.



Das Heimdall-GUI vor der Treiber-Installation ... (Bild 2)



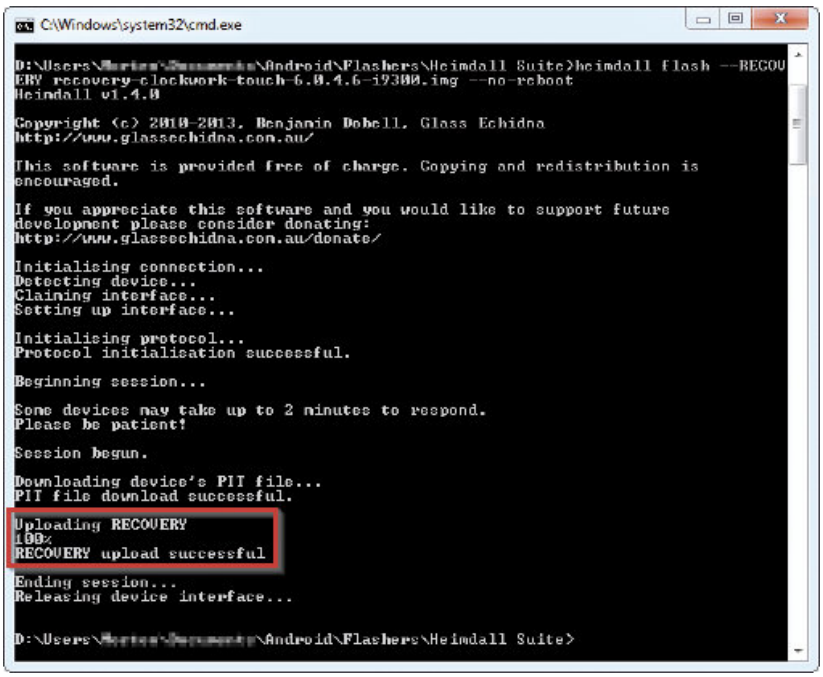
... und danach (Bild 3)

Treiberinstallation

Klicken Sie auf *Replace Driver*. Nach erfolgreicher Treiberinstallation (Bild 3) schließen Sie das Heimdall-GUI und trennen das USB-Kabel. Starten Sie das Gerät erneut im Download-Modus und schließen Sie das USB-Kabel an. Öffnen Sie auf dem PC ein Konsolenfenster im Ordner *Heimdall Suite* und führen Sie folgen- ▶

TABELLE 1: BENÖTIGTE SOFTWARE FÜR DIE INSTALLATION

Software	Beschreibung	Webadresse
Samsung USB Driver for Mobile Phones	Windows-Treiber für Samsung-Mobilgeräte. Offiziell nur für nordamerikanische Geräte verfügbar, funktionieren aber auch mit europäischen Geräten	www.samsung.com/us/support/owners/product/SCH-R530MBBXAR
Heimdall Suite	Tool zum Aufspielen von System-Images. Wird zum Aufspielen von ClockWorkMod Recovery benutzt	www.glassechidna.com.au/products/heimdall
ClockWorkMod Recovery	Tool für die Installation und Wiederherstellung des Systems. Wird zur Installation von CM benutzt	www.clockworkmod.com/rommanager, Spalte *Download Touch Recovery*
CM-Paket für Samsung Galaxy S3 (i9300)	Das fertiggestellte Android/CM-System, das aufgespielt werden soll. Wählen Sie den neuesten Milestone-Snapshot	http://wiki.cyanogenmod.org/w/I9300_Info
Stock Recovery	Original-Recovery-Datei des GT-I9300	http://stockroms.net/file/GalaxyS3/i9300/stockrecovery/International-GT-i9300-StockRecovery.tar



Nach Upload des ClockWorkMod Recovery (Bild 4)

den Befehl aus (ersetzen Sie *clockworkmod.img* mit dem Dateinamen Ihres ClockWorkMod-Recovery-Image):

```
heimdall flash --RECOVERY
clockworkmod.img --no-reboot
```

Nach erfolgreichem Austausch (Bild 4) trennen Sie das USB-Kabel.

Wechsel in den Recovery-Modus

Nun starten Sie das Gerät im Recovery-Modus, sodass das Betriebssystem ersetzt werden kann. Dazu drücken und halten Sie am Smartphone gleichzeitig die Tasten *Lauter*, *Home* und *Power*.

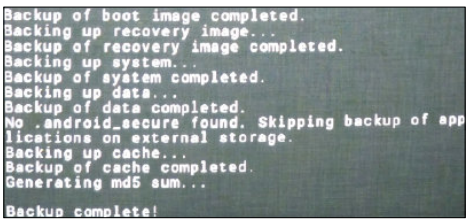
Vor dem Austausch sollten Sie in jedem Fall das ausgelieferte Betriebssystem auf der SD-Karte sichern, um es bei Bedarf zurückspielen zu können. Wählen Sie hierzu *backup and restore, backup to /sdcard* (Bild 5).

Nun erfolgt die eigentliche Installation. Setzen Sie das Gerät zunächst auf Werkseinstellungen zurück *(wipe data/factory reset)* und löschen Sie den Dalvik-Cache *(advanced, wipe dalvik cache)*. Installieren Sie CM über *install zip, choose zip from sdcard* und starten Sie das System neu *(reboot system now)*.

Warten Sie, bis der Startvorgang abgeschlossen ist; dies dauert einige Minuten. Anschließend ist Ihr neues System betriebsbereit (Bild 6).

Kopieren Sie abschließend die erstellten Sicherungsdateien von der SD-Karte auf Ihren PC. Die Dateien befinden sich auf dem Smartphone im Verzeichnis */storage/emulated/clockworkmod/backup*, das Sie zum Beispiel mit Hilfe des vorinstallierten Dateimanagers finden, nachdem Sie dessen Zugriffsmodus unter *Einstellungen, Allgemeine Einstellungen* auf *Rootzugriffsmodus* gesetzt haben.

Das Backup wurde erfolgreich abgeschlossen (Bild 5)



Deinstallation

Sollten Sie zu einem späteren Zeitpunkt zu Ihrem ursprünglichen System zurückkehren wollen, können Sie das Gerät wie folgt in seinen Ausgangszustand zurückversetzen:

Entpacken Sie das Stock Recovery Image (zum Beispiel mit 7-Zip) und kopieren Sie es in das *Heimdall*-Verzeichnis. Sollten Sie die Samsung-Software Kies auf Ihrem PC installiert haben, stellen Sie sicher, dass diese nicht aktiv ist.

Nun wechseln Sie in den Recovery-Modus. Spielen Sie die Originalsicherung zurück *(backup and restore, restore from /sdcard)*, setzen Sie das Gerät auf seine Werkseinstellungen zurück *(wipe data/factory reset)* und löschen Sie den Dalvik-Cache *(advanced, wipe dalvik cache)*.

Wechseln Sie anschließend in den Download-Modus und schließen Sie das USB-Kabel an. Öffnen Sie auf dem PC ein Konsolenfenster im Ordner *Heimdall Suite* und führen Sie folgenden Befehl aus (ersetzen Sie *stockrecovery.img* mit dem Dateinamen des Stock Recovery Image):

```
heimdall flash --RECOVERY
stockrecovery.img --no-reboot
```

Kommt es bei diesem Schritt zu einem *Failed to access device*-Fehler, installieren Sie zunächst den USB-Treiber.

Zum Abschluss trennen Sie das USB-Kabel und starten das Gerät neu. Danach befindet sich Ihr System im Ausgangszustand, allerdings mit einer Einschränkung: Der Zähler für die Anzahl durchgeführter Installationen wurde hochgezählt. Sie können diesen Zähler im Download-Modus sehen. Zwar gibt es Ansätze, den Zähler zurückzusetzen, aber diese sind umständlich, basieren auf fraglicher Software und schließen

ODIN UND TRIANGLE AWAY

Um ein Mobilgerät komplett in den Auslieferungszustand zurückzusetzen, müssen nicht nur ein Stock ROM und eine Stock Recovery aufgespielt werden, sondern es muss auch der Zähler für durchgeführte Installationen zurückgesetzt werden. Hierzu werden oft das Tool Odin und die App Triangle Away genutzt. Odin ist allerdings ein Tool, das Samsung intern einsetzt und offiziell gar nicht zur Verfügung stellt; Triangle Away wird vom Entwickler nicht mehr unterstützt und funktioniert seit Android 4.3 nicht mehr. Es ist daher nicht ratsam, sich auf diese Werkzeuge zu verlassen.

ANDERE ANDROID-GERÄTE

CM ist für unzählige Smartphones und Tablets verfügbar, das Verfahren zur Installation unterscheidet sich aber von Modell zu Modell. Ob Ihr Gerät offiziell unterstützt wird, erfahren Sie im CM-Wiki unter http://wiki.cyanogenmod.org/w/Devices. Dort finden Sie auch eine detaillierte Beschreibung des jeweiligen Installationsprozesses.

letztendlich nicht aus, dass der Hersteller an anderen Parametern dennoch erkennen kann, ob das Betriebssystem neu installiert wurde.

Da sowohl Android als auch CM quelloffen und als freie Software lizenziert sind, steht einer Anpassung an Ihre eigenen Bedürfnisse nichts im Wege. Daher wird im Folgenden gezeigt, wie CM aus dem Quellcode kompiliert werden kann.

Als Umgebung soll Kubuntu dienen, das in einer virtuellen Maschine unter VirtualBox läuft; dies hat den Vorteil, dass zu Lernzwecken beliebige Änderungen ausprobiert und bei Bedarf die ganze Umgebung zurückgesetzt werden kann. Anschließend wird das Smartphone vorbereitet, die Build-Umgebung eingerichtet und ein Installationspaket aus dem zunächst unveränderten Quellcode erstellt. Im letzten Schritt wird die Installation angepasst – eine vorinstallierte App wird entfernt, eine neue hinzugefügt, und die Startanimation wird ausgetauscht, um ein eigenes Branding zu realisieren.

Einrichtung einer virtuellen Maschine

Laden Sie zunächst Oracle VM VirtualBox (mit Extension Pack für USB-2.0-Unterstützung) sowie Kubuntu 64 Bit herunter (Tabelle 2). Installieren Sie VirtualBox und das Extension Pack. Anschließend erstellen Sie eine neue virtuelle Maschine und binden die Kubuntu-ISO-Datei als CD/DVD-Laufwerk (Primärer Master) ein. Es ist wichtig, dass Sie die 64-Bit-Version wählen, denn aktuelle Android-Versionen können nur auf 64-Bit-Systemen erstellt werden. Binden Sie außerdem die VirtualBox-Gasterweiterungen ein (die ISO-Datei befindet sich im Virtual-Box-Programmverzeichnis).

Stellen Sie sicher, dass *IO-APIC* aktiviert ist (Bild 7), dies ist Voraussetzung für 64-Bit-Gastsysteme). Weisen Sie die maximal mögliche Anzahl an CPU-Kernen zu und wählen Sie als Netzwerkadapter *Netzwerkbrücke* aus (Bild 8), da es sonst bei der CM-Entwicklung zu Problemen mit OpenSSL kommen kann. Schließen Sie das Smartphone per USB an und fügen Sie den passenden USB-Filter hinzu (Bild 9).

Starten Sie nun die virtuelle Maschine und damit die Kubuntu-Installation. Wenn das System Sie fragt, ob Sie Software von Drittanbietern installieren wollen, können Sie dies ablehnen. Installieren Sie die Unterstützung für dynamische Kernelmodule (dkms). Dazu führen Sie in der Konsole folgenden Befehl aus:

```
sudo apt-get install dkms
```

Installieren Sie die VirtualBox-Gasterweiterungen (die Werte in Klammern müssen Sie mit Ihren Werten ersetzen):

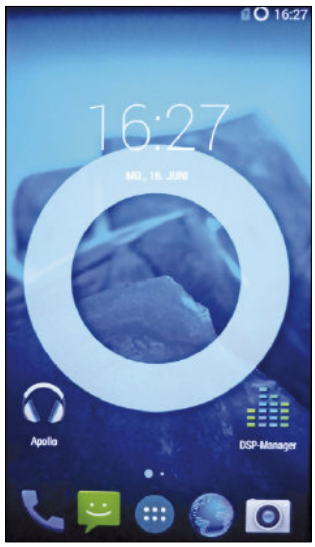
```
sudo sh /media/(username)/
VBOXADDITIONS_(version)/autorun.sh
```

Abschließend führen Sie einen Neustart aus und aktualisieren alle Kubuntu-Pakete.

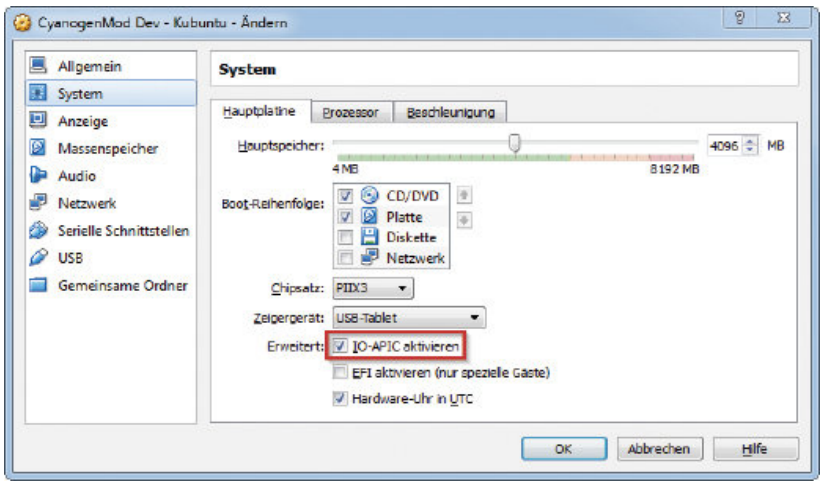
Um direkt auf das Smartphone zugreifen zu können, aktivieren Sie zunächst die Entwickleroptionen. Das erreichen Sie, indem Sie unter *Einstellungen, Über das Telefon* siebenmal auf *Build-Nummer* tippen. Anschließend aktivieren Sie unter *Einstellungen, Entwickleroptionen* den Punkt *Aktiv lassen*, setzen den Eintrag *Root-Zugriff* auf *Apps & ADB* und aktivieren den Punkt *USB-Debugging*.

Einige Skripts, die später eingesetzt werden, erwarten als Standard-Konsole die bash-Shell. Da Ubuntu und seine Derivate jedoch per Voreinstellung die dash-Shell benutzen, muss der virtuelle Link *sh* auf die bash-Shell umgebogen werden:

```
sudo rm /bin/sh
sudo ln -s bash /bin/sh
```



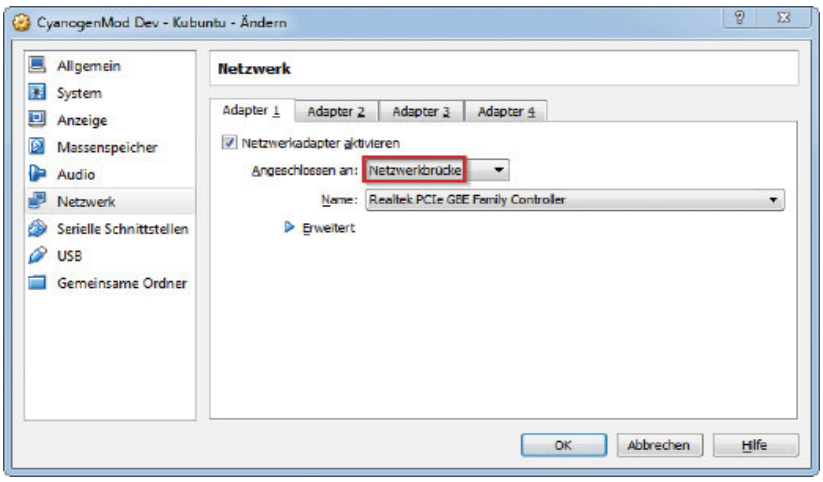
CyanogenMod 11 ist bereit (Bild 6)



▶ VirtualBox: *IO-APIC* muss gesetzt sein (Bild 7)

TABELLE 2: BENÖTIGTE SOFTWARE FÜR DIE ENTWICKLUNG

Software	Webadresse
Oracle VM VirtualBox mit Extension Pack	https://www.virtualbox.org/wiki/Downloads
Kubuntu 64 Bit	www.kubuntu.org/getkubuntu
Android SDK mit Eclipse (ADT Bundle)	http://developer.android.com/sdk



Konfiguration des Netzwerkadapters als Netzwerkbrücke (Bild 8)

Nun beginnt der eigentliche Aufbau der Umgebung. Legen Sie die folgenden Verzeichnisse an:

```
mkdir -p ~/bin
mkdir -p ~/android
mkdir -p ~/android/system
```

Installieren Sie die Bibliotheken zur Unterstützung von 32-Bit-Komponenten:

```
sudo apt-get install lib32z1
lib32ncurses5 lib32bz2-1.0
```

Anschließend installieren Sie die Bibliotheken, die zum Erstellen von Android/CM benötigt werden:

```
sudo apt-get install bison
build-essential curl flex
g++-multilib gcc-multilib git-core
gperf lib32ncurses5-dev
lib32readline-gplv2-dev lib32z1-dev
libesd0-dev libncurses5-dev
libsdl1.2-dev libwxgtk2.8-dev lzop
openjdk-7-jdk pngcrush schedtool
squashfs-tools xsltproc zlib1g-dev
```

Beachten Sie, dass hier gemäß der Empfehlung der Ubuntu-Community (http://wiki.ubuntuusers.de/Java/Installation) OpenJDK installiert wird, obwohl für Android offiziell Oracle (Sun) Java benötigt wird. Erfahrungsgemäß gibt es mit OpenJDK keinerlei Probleme.

Laden Sie nun das ADT Bundle herunter und entpacken Sie es nach *~/android*. Die resultierende Dateistruktur sehen Sie in Bild 10. Fügen Sie zwei Linux-Pfadvariablen hinzu, damit Sie einen bequemen Zugriff auf diverse Tools wie etwa adb haben. Dazu öffnen Sie die Datei *environment*:

```
sudo nano /etc/environment
```

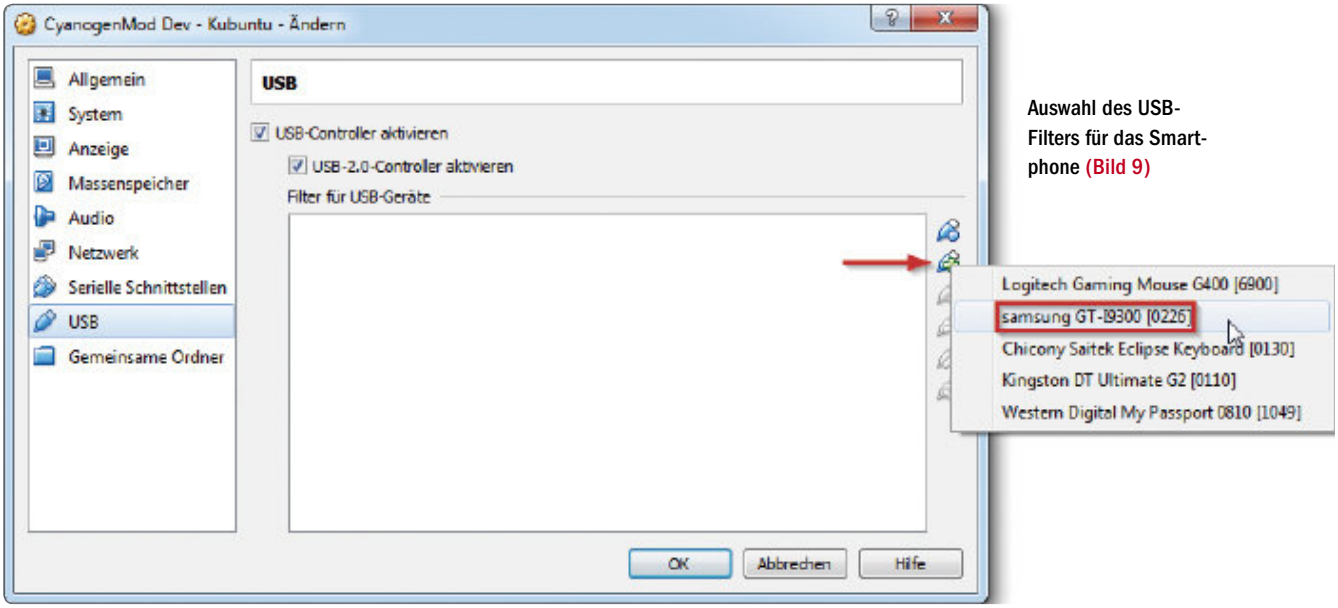
und fügen folgenden Eintrag am Zeilenende (vor dem schließenden Anführungszeichen) ein:

```
:~/bin:~/android/sdk/platform-tools
```

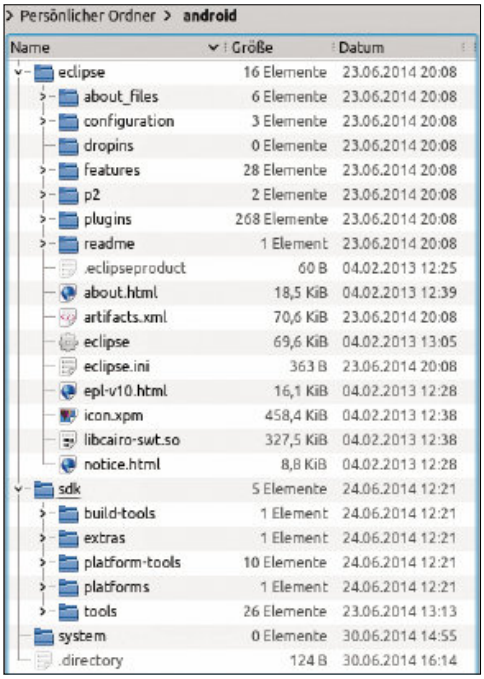
Stellen Sie sicher, dass die aktuelle SDK-Plattform und die SDK-Werkzeuge installiert sind. Dazu öffnen Sie den Android SDK Manager unter *~/android/sdk/tools/android* (Bild 11). Hier können Sie optional auch zusätzliche oder ältere Komponenten hinzufügen. Sollte der Android SDK Manager nicht alle Pakete laden, schauen Sie im Log-Fenster nach. Eine *Peer not authenticated*-Fehlermeldung deutet auf ein Problem mit der SSL-Verbindung hin.

Als Nächstes installieren Sie *repo*. Dies ist ein Skript, das im Kontext von Android die Arbeit mit Git erleichtert:

```
curl http://commondatastorage.
googleapis.com/git-repo-downloads/
```



Auswahl des USB-Filters für das Smartphone (Bild 9)



Die Dateistruktur nach dem Entpacken des ADT-Bundles (Bild 10)

```
repo > ~/bin/repo
chmod a+x ~/bin/repo
```

Setzen Sie in Git Ihre Standardidentität, indem Sie Ihre E-Mail-Adresse angeben:

```
git config --global user.email
"(Email-Adresse)"
```

Führen Sie anschließend einen Neustart aus.

Im nächsten Schritt bereiten Sie das CM-Quellcode-Repository vor. Dazu müssen Sie zunächst sicherstellen, dass Port 9418 für den Zugriff auf GitHub.com offen ist. Entscheiden Sie sich anschließend, auf welchem CM-Zweig (Branch) Sie arbeiten wollen. Die vorhandenen Zweige finden Sie unter https://github.com/CyanogenMod; suchen Sie hier nach der Modellnummer i9300. Empfehlenswert ist das letzte stabile Release, in unserem Beispiel *stable/cm-11.0* (Bild 12). Navigieren Sie dann in das *system*-Verzeichnis:

```
cd ~/android/system
```

und führen Sie den folgenden Befehl aus:

```
repo init -u git://github.com/
CyanogenMod/android.git
-b stable/cm-11.0
```

Beantworten Sie die gestellten Fragen. Anschließend ist das Repository initialisiert.

Bevor Sie den CM-Quellcode synchronisieren, sollten Sie über *repo selfupdate* sicherstellen, dass das *repo*-Skript aktuell ist. Streng genommen ist dies hier nicht notwendig, da Sie es gerade erst installiert haben, aber man sollte es sich zur Gewohnheit machen, das Skript immer mal wieder zu aktualisieren, da es sonst bei der Synchronisierung leicht zu Problemen kommen kann. Der Fehler *fatal: Invalid gitfile format* ist beim ersten Ausführen normal und kann ignoriert werden.

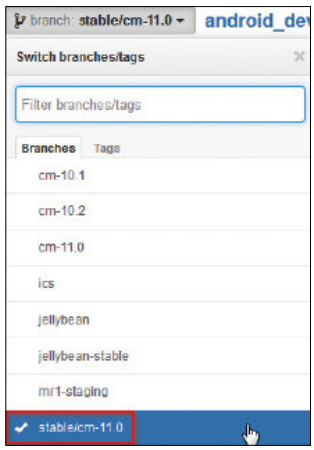
Nun können Sie mit *repo sync* den Quellcode synchronisieren, was initial mehrere Stunden dauert. Nach erfolgreicher Synchronisierung integrieren Sie wie folgt die von CM zur Verfügung gestellten Apps:

```
cd ~/android/system/vendor/cm
./get-prebuilts
```

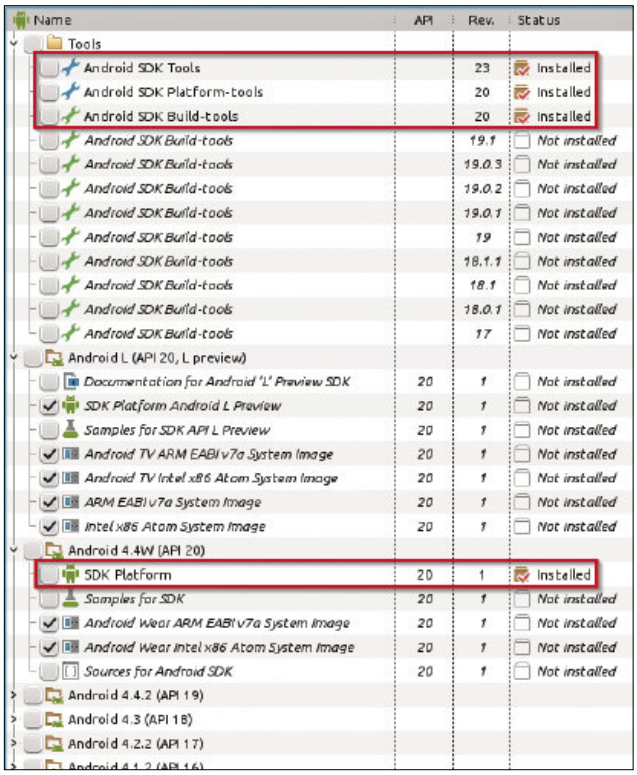
Als Nächstes bereiten Sie die Build-Umgebung vor, indem Sie das Skript *envsetup* im system-Ordner ausführen. Dies erleichtert die anschließende Arbeit, da zusätzliche Befehle zur Verfügung stehen. Das Skript muss allerdings jedes Mal neu ausgeführt werden, wenn Sie ein Konsolen-Fenster öffnen. Sollte also im Folgenden ein Befehl nicht ausführbar sein, könnte es daran liegen, dass Sie *envsetup* erneut ausführen müssen.

```
cd ~/android/system
source build/envsetup.sh
```

Nun konfigurieren Sie mit *breakfast i9300* die Build-Umgebung für das Smartphone-Modell und starten den adb-Server mit Root-Rechten *(adb root)*. Damit erhalten Sie Vollzugriff auf ▶



Auswahl eines CM-Quellcode-Zweiges (Bild 12)



Die minimal benötigten Komponenten, dargestellt im Android SDK Manager (Bild 11)

das Gerät. Die Fehlermeldung *device not found* können Sie an dieser Stelle ignorieren, da noch kein Gerät angeschlossen ist.

Jetzt entsperren Sie das Smartphone und verbinden es per USB mit dem PC. Verbinden Sie das Gerät direkt mit dem Computer, nicht über einen USB-Hub. Seit Android 4.2 fordert das Smartphone Sie an dieser Stelle auf, eine Erlaubnis für die Verbindung zu erteilen (Bild 13). Bestätigen Sie den Dialog und testen Sie mit *adb devices*, ob die Verbindung steht. Bei Erfolg werden eine ID sowie der Begriff *device* angezeigt (Bild 14).

Das Gerät benötigt noch einige proprietäre Dateien des Herstellers, die nicht frei erhältlich sind. Extrahieren Sie diese BLOBs wie folgt aus dem Smartphone:

```
cd ~/android/system/device/samsung/
i9300
./extract-files.sh
```

Nun kann kann das Installationspaket erstellt werden. Aktivieren Sie aber zunächst den Compiler-Cache, damit nachfolgende Builds schneller erfolgen:

```
cd ~/android/system
export USE_CCACHE=1
```

Anschließend starten Sie den Build-Vorgang. Auch dies dauert mehrere Stunden, insbesondere bei der ersten Ausführung:

```
brunch i9300
```

ADB

Die Android Debug Bridge ist ein unverzichtbares Werkzeug bei der Arbeit mit Android-Geräten; sie erlaubt das Kopieren von Dateien zwischen PC und Mobilgerät, die Installation von Apps über Paketdateien, das Öffnen von Log-Dateien und vieles mehr. Eine Befehlsreferenz finden Sie unter http://developer.android.com/tools/help/adb.html.

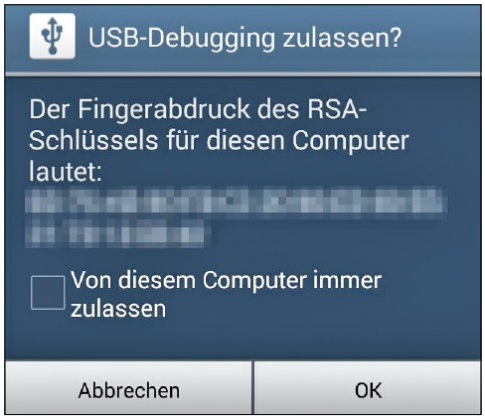
Sie finden das Kommandozeilen-Tool *(adb.exe)* im Unterverzeichnis *platform-tools* des Android-SDK-Verzeichnisses.

LISTING 1: MAKE-DATEI (ANDROID.MK) FÜR K-9 MAIL

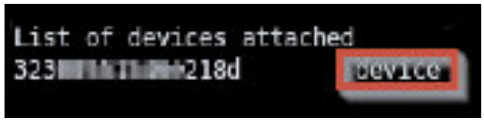
```
LOCAL_PATH := $(call my-dir)
include $(CLEAR_VARS)
LOCAL_MODULE := K9Mail
LOCAL_SRC_FILES := com.fsck.k9-1.apk
LOCAL_MODULE_CLASS := APPS
LOCAL_MODULE_SUFFIX := $(COMMON_ANDROID_PACKAGE_SUFFIX)
LOCAL_MODULE_TAGS := optional
LOCAL_CERTIFICATE := PRESIGNED
include $(BUILD_PREBUILT)
```

LISTING 2: LOKALE GIT-REPO-KONFIGURATION

```
<?xml version="1.0" encoding="UTF-8"?>
<manifest>
<remove-project name=
"CyanogenMod/android_packages_apps_VideoEditor" />
<remote name="k9mail" fetch=
"file:///home/(username)/android/projects/" />
<project path="external/k9mail" name="k9mail" remote="k9mail"
revision="master" />
</manifest>
```



Sicherheitsabfrage des Smartphones beim ersten Zugriff per USB (Bild 13)



Die Verbindung zum Smartphone ist hergestellt (Bild 14)

Das fertige Installationspaket finden Sie anschließend im Build-Verzeichnis, das Sie über *cd $OUT* erreichen. Zur Installation kopieren Sie es per *adb* auf das Smartphone. Ersetzen Sie dabei den Dateinamen durch Ihren eigenen:

```
adb push ~/android/system/out/
target/product/i9300/cm-11-(Datum)-
UNOFFICIAL-i9300.zip /sdcard/
cm-11-(Datum)-UNOFFICIAL-i9300.zip
```

Sie finden im Build-Verzeichnis zudem ein Image von ClockWorkMod Recovery *(recovery.img)* und können es wie beschrieben im Download-Modus installieren.

CM aus dem Quellcode zu erstellen ist vor allem dann sinnvoll, wenn man eigene Anpassungen am System vornehmen will. Beispielhaft soll hier eine CM-App (Video Editor) entfernt und eine neue App (K-9 Mail) in die Installation aufgenommen werden. Legen Sie dazu zunächst ein Projektverzeichnis für die neu zu integrierende App an:

```
mkdir ~/android/projects
mkdir ~/android/projects/k9mail
```

Besorgen Sie sich anschließend die *.apk*-Datei von K-9 Mail und legen Sie sie im neu erzeugten Verzeichnis ab. Dann erzeugen Sie eine Make-Datei für den Erstellungsprozess:

```
nano ~/android/projects/k9mail/
Android.mk
```

In diese Make-Datei tragen Sie den Inhalt aus Listing 1 ein. Anschließend legen Sie mit den

nachfolgenden Befehlen ein lokales Git-Repository an:

```
cd ~/android/projects/k9mail
git init
git add .
sudo git commit
```

Hierauf erstellen Sie eine Konfigurationsdatei mit einem beliebigen Dateinamen und der Endung *.xml* (zum Beispiel *custom.xml*):

```
nano ~/android/system/.repo/
local_manifests/custom.xml
```

In diese Datei werden die neu hinzuzufügenden und die zu entfernenden Apps eingetragen (Listing 2). Im Listing müssen Sie *username* durch Ihren eigenen Wert ersetzen. Auf diese Weise berücksichtigt *repo* die App bei der Synchronisierung. Namen existierender Apps sind in *~/android/system/.repo/manifest.xml* hinterlegt und können dort nachgeschlagen werden.

Sofern noch eine *.apk*-Datei der Video-Editor-App aus einem früheren Build existiert, löschen Sie diese:

```
rm ~/android/system/out/target/
product/i9300/system/app/
VideoEditor.apk
```

Um ganz sicherzugehen, dass keine Reste einer vorherigen Version zurückbleiben, können Sie auch das komplette Ausgabeverzeichnis löschen:

```
cd ~/android/system
make clean
```

Synchronisieren Sie erneut *(repo sync)*, um die Anpassungen zu übernehmen. Damit die Make-Datei beim Erstellungsprozess berücksichtigt wird, müssen Sie noch die Standard-Konfigurationsdatei *common.mk* öffnen:

```
nano ~/android/system/device/samsung/
smdk4412-common/common.mk
```

und folgende Zeile hinter einem der vorhandenen *PRODUCT_PACKAGES*-Einträge einfügen:

```
PRODUCT_PACKAGES += K9Mail
```

Nach erneuter Erstellung *(brunch i9300)* werden Ihre Änderungen in das Installationspaket übernommen.

Um dem Gerät ein eigenes Aussehen zu verleihen, soll die Startanimation ausgetauscht werden. Die ausgelieferte Animation besteht aus zwei Sequenzen zu je 48 Bildern sowie einer Konfigurationsdatei *desc.txt*, in der die Bildgröße, die Anzahl der Bilder pro Sequenz und die Anzahl der Wiederholungen jeder Sequenz eingetragen ist (Bild 15).

Konfiguration der Startanimation in der Datei *desc.txt* (Bild 15)

Am besten schauen Sie sich die vorhandene Animation in */system/media/bootanimation.zip* auf dem Gerät an.

Zum Austausch gehen Sie wie folgt vor: Legen Sie ein Projektverzeichnis für die neue Startanimation an:

```
mkdir ~/android/projects/bootanimation
```

Erstellen Sie die Grafikdateien und legen Sie jede Sequenz in einen Unterordner, wie in der vorhandenen *bootanimation.zip* zu sehen. Erstellen Sie außerdem die *desc.txt*-Datei und tragen Sie die passende Konfiguration ein.

Verpacken Sie alles in eine unkomprimierte ZIP-Datei namens *bootanimation.zip* und legen Sie sie im Projektverzeichnis ab. Da hier nur eine einzelne ZIP-Datei an den richtigen Ort kopiert werden muss, benötigen Sie keine Make-Datei und auch keine Synchronisierung per Git-Repository. Stattdessen soll die Datei direkt aus dem Projektverzeichnis an den Zielort kopiert werden. Dazu öffnen Sie *common.mk*:

```
nano ~/android/system/device/samsung/
smdk4412-common/common.mk
```

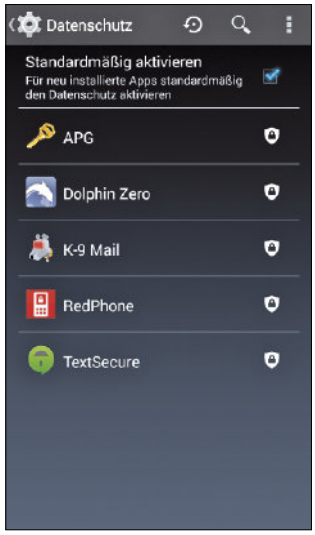
und fügen den folgenden Eintrag hinter einem der vorhandenen *PRODUCT_COPY_FILES*-Einträge hinzu:

```
PRODUCT_COPY_FILES += ~/android/
projects/bootanimation/bootanimation.
zip:system/media/bootanimation.zip
```

Damit wird die Standard-Startanimation bei Erstellung des Installationspakets mit Ihrer eigenen Animation überschrieben.

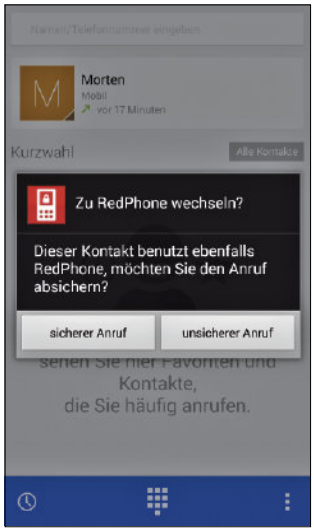
Konfiguration

Das neu erstellte CM-System ist noch sehr karg ausgestattet. Im Folgenden wird gezeigt, wie Sie einerseits mit Bordmitteln, andererseits durch Installation zusätzlicher Apps das Gerät so einrichten können, dass der Schutz Ihrer persönlichen Daten gewährleistet und eine sichere Kommunikation über verschiedene Kanäle (Telefon, Instant Messaging, E-Mail) möglich ist. Dabei wird die CM-Version 11 M9 (Android 4.4.4) vorausgesetzt. ▶

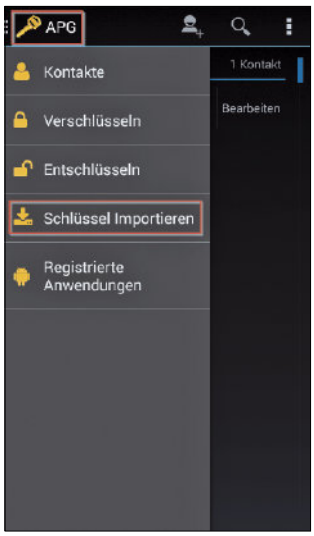


Der Privacy Guard schränkt die Zugriffsrechte einzelner Apps ein und sollte standardrmäßig aktiviert sein (Bild 16)

Gruppierte Ansicht der App-Zugriffsrechte mit Nutzungsstatistik (Bild 17)



RedPhone schlägt ein verschlüsseltes Telefonat vor (Bild 18)



Schlüsselimport in APG (Bild 19)

Der Schlüssel wurde erfolgreich von einem öffentlichen Schlüssel-Server importiert (Bild 20)

Der wichtigste und erste Schritt: Aktivieren Sie die Bildschirmsperre, damit niemand an Ihre Daten gelangt, wenn Sie das Gerät unbeaufsichtigt lassen. Dazu setzen Sie unter *Einstellungen, Gruppe Personalisierung, Bildschirmsperre, Bildschirmsicherheit* den Punkt *Automatisch sperren* auf *Sofort* und aktivieren den Punkt *Ein/Aus sperrt Gerät*. Im Untermenü *Display-Sperre* wählen Sie entweder *PIN* oder *Passwort*.

Damit die Bildschirmsperre schnell greift, sollten Sie sicherstellen, dass der Ruhezustand nach relativ kurzer Ruhezeit einsetzt. Unter *Einstellungen, Gruppe Gerät, Display & LED, Gruppe Display* setzen Sie den Punkt *Ruhezustand* auf einen niedrigen Wert, zum Beispiel auf 15 Sekunden.

Um einem Missbrauch Ihres Mobilfunkvertrags vorzubeugen, sperren Sie außerdem den Zugriff auf die SIM-Karte; hierzu aktivieren Sie unter *Einstellungen, Sicherheit, Gruppe SIM-Kartensperre, SIM-Sperre einrichten* den Punkt *SIM-Karte sperren*.

Es sind einige (eher theoretische) Szenarien denkbar, in denen die Bildschirmsperre umgangen werden kann (beispielsweise durch Ausnutzen einer Sicherheitslücke in adb oder Android, durch Hochfahren des Geräts von einer SD-Karte, oder durch Manipulation der Hardware), sodass Sie unter Umständen die Datensicherheit durch Komplettverschlüsselung des Geräts weiter erhöhen wollen. Die entsprechende Funktion finden Sie unter *Einstellungen, Gruppe Nutzer, Sicherheit, Gruppe Verschlüsselung, Telefon verschlüsseln*. Beachten Sie aber, dass Ihre Daten bei Verlust des Verschlüsselungspassworts unwiderruflich verloren sind.

Wenn Sie möglichst wenig Informationen über Ihren Standort preisgeben wollen, setzen Sie unter *Einstellungen, Gruppe Nutzer, Standort* den Punkt *Modus* auf *Nur Gerät*. Stellen Sie außerdem sicher, dass das Smartphone bei abgeschaltetem WLAN-Modul nicht nach Netzwerken in Ihrer Nähe sucht (tippen Sie unter *Einstellungen, Gruppe Drahtlos & Netzwerke, WLAN* im Menü auf *Erweitert* – der Punkt *Erkennungsfunktion immer verfügbar* muss deaktiviert sein). Diese Maßnahmen verhindern allerdings nicht, dass Sie ein Bewegungsprofil über die Mobilfunkverbindung hinterlassen, sofern Sie das Gerät nicht komplett ausschalten.

In der Öffentlichkeit ist es häufig ein Leichtes, Personen beim Eintippen ihres Passworts über die Schulter zu schauen. Diese Form des Passwortdiebstahls können Sie erschweren, indem Sie die Anzeige von Passwörtern ausschalten. Dazu deaktivieren Sie unter *Einstellungen, Gruppe Nutzer, Sicherheit, Gruppe Passwörter* den Punkt *Passwörter sichtbar*.

Falls Sie zuvor Apps per *.apk*-Datei installiert haben, denken Sie daran, vor Inbetriebnahme des Geräts die Installation aus unbekannten Quellen wieder zu verbieten. Deaktivieren Sie dazu unter *Einstellungen, Gruppe Nutzer, Sicherheit, Gruppe Geräteverwaltung* den Punkt *Unbekannte Herkunft*.

Auch die Zugriffsrechte der installierten Apps sollten Sie nach Möglichkeit einschränken. CM bringt zu diesem Zweck den Privacy Guard mit. Stellen Sie sicher, dass unter Einstellungen, *Gruppe Nutzer, Datenschutz, Datenschutz* der Punkt *Standardmäßig aktivieren* gesetzt ist (Bild 16). An dieser Stelle können Sie auch selektiv Rechte einräumen, sollte eine App aufgrund fehlender Berechtigungen nicht funktionieren (tippen und halten Sie dazu den Eintrag der entsprechenden App). Über den Menüpunkt *Erweitert* gelangen Sie in eine ausführliche Übersicht über die aktuell genehmigten Einzelrechte und eine Statistik über deren Nutzung (Bild 17).

Wenn Sie öffentliche WLAN-Hotspots nutzen, können Angreifer, die im gleichen Netz angemeldet sind wie Sie, Ihren Datenverkehr oftmals leicht mitlesen. Das beste Mittel gegen diese Gefahr ist ein VPN (Virtual Private Network). Hierbei stellen Sie eine verschlüsselte Verbindung zu einem VPN-Server her, sodass Sie nicht mehr über ein fremdes Netzwerk, sondern über den VPN-Server ins Internet gelangen. Die Verbindung zum VPN-Server konfigurieren Sie über *Einstellungen, Gruppe Drahtlos & Netzwerke, Mehr …, VPN*.

Antivirus

Obwohl die üblichen Schädlinge wie Viren, Trojaner oder Würmer im Mobilbereich (noch) nicht die gleiche Bedeutung haben wie im Desktop-Segment, ist die Installation einer Antiviren-App empfehlenswert – insbesondere, da ohne Play Store alle Apps per *.apk*-Datei installiert werden müssen.

Für welche Antiviren-App Sie sich entscheiden, sollten Sie von aktuellen Testberichten, wie sie zum Beispiel AV-Test liefert, abhängig machen. Neben den bekannten Namen wie Avira, Bitdefender oder Kaspersky haben sich im Mobilbereich auch neuere Anbieter wie AhnLab, 360 oder Quick Heal bewährt.

Sichere Telefonie und SMS

Die quelloffene App RedPhone der Firma Open Whisper Systems ermöglicht Ende-zu-Ende-verschlüsselte Telefonie. Die Gespräche werden dabei mittels Voice over IP über die Server des Herstellers geleitet. Zu diesem Zweck müssen Sie sich nach der Installation mit Ihrer Telefonnummer registrieren. Die Bedienung ist sehr intuitiv – die App integriert sich in den regulären Telefonie-Vorgang und schlägt ein sicheres Te-

lefonat vor, wenn beide Gesprächspartner die App installiert haben (Bild 18). Ist dies nicht der Fall, können Sie Ihr Gegenüber zur Installation von RedPhone auffordern, indem Sie einen entsprechend vorbereiteten Link verschicken.

Ebenfalls Open Source, ermöglicht Linphone die sichere Kommunikation per Sprache und Video. Die Software ist für alle gängigen Betriebssysteme (mobil und Desktop) verfügbar. Linphone überträgt auch Bilder und Text, allerdings unverschlüsselt. Außerdem ist die App nur in englischer und französischer Sprache verfügbar. Eine weitere Alternative ist CSipSimple, das als SIP-Client (Session Initiation Protocol) mit mehreren Anbietern von SIP-Diensten genutzt werden kann.

Die App TextSecure, die ebenso wie RedPhone von Open Whisper Systems stammt, galt lange Zeit als erste Wahl für verschlüsselte SMS- und MMS-Nachrichten. Nachdem der Hersteller jedoch angekündigt hat, die Verschlüsselung für diese Nachrichtentypen aus der App zu entfernen, scheidet sie – zumindest für SMS / MMS – aus.

Als kommerzielle Alternative bleibt noch die App-Suite der Firma Silent Circle, die mit Silent Phone und Silent Text eine Lösung sowohl für die Telefonie als auch den Versand der klassischen Kurznachrichten anbietet. In Ergänzung dazu können Kontaktinformationen mit der App Silent Contacts verschlüsselt werden. Diese Apps sind allerdings nicht quelloffen, sodass keine Kontrolle der verwendeten Verschlüsselungstechnologie möglich ist und dem Hersteller ein gewisses Grundvertrauen entgegengebracht werden muss.

Sicheres Instant Messaging

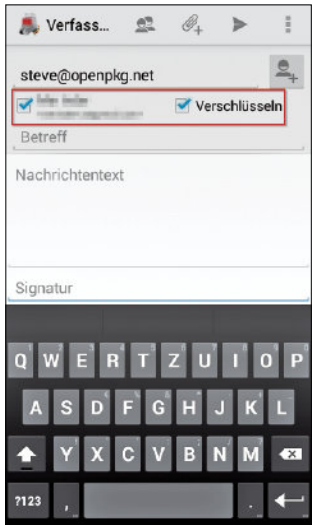
Das bereits erwähnte TextSecure eignet sich auch ohne SMS-Versand hervorragend zur sicheren Kommunikation per Instant Messaging. Die Open-Source-App bietet Ende-zu-Ende-Verschlüsselung, lokale Verschlüsselung der Nachrichtendatenbank sowie Folgenlosigkeit (Forward Secrecy; Nachrichten können nicht im Nachhinein entschlüsselt werden, selbst wenn der private Schlüssel bekannt wird). Die Einrichtung geht schnell und problemlos und die Bedienung ist intuitiv.

Interessant ist, dass CM WhisperPush mitbringt, das mit TextSecure kompatibel ist und jenes ersetzen soll. Allerdings lässt sich WhisperPush nur einsetzen, wenn man die Google-Play-Store-App installiert – was unserem ursprünglichen Ziel, ein System ohne Google Apps zu betreiben, zuwiderläuft. Auch steckt WhisperPush noch in den Kinderschuhen – in einem Schnelltest ließ es sich auch mit installiertem Play Store nicht starten.

Die Messenger-App Telegram ist teilweise quelloffen und besonders im russischen Sprachraum verbreitet. Sie kann auf Mobilgeräten sowie (durch inoffizielle Versionen) auf Desktop-Computern genutzt werden. Kritisch ist allerdings, dass die Verschlüsselung explizit aktiviert werden muss. Außerdem überträgt die App Ihre komplette Kontaktliste an den Server des Herstellers.

Threema wird kommerziell von der Schweizer Firma Threema GmbH betrieben und gehört im deutschsprachigen Raum zu den beliebtesten Messengern mit sicherer Verschlüsselung. Laut Herstellerangabe befinden sich die Server ausschließlich in der Schweiz, sodass die Firma dem schweizerischen Bundesgesetz über den Datenschutz unterliegt. Dementsprechend erhielt die App im Februar 2014 von der Stiftung Warentest das Urteil »unkritisch« im Bereich Datenschutz. Threema ist allerdings nicht quelloffen.

Das Besondere am Redact Secure Messenger ist, dass eine Peer-to-Peer-Verbindung zwischen den Kommunikationspartnern aufgebaut wird; laut Hersteller werden keinerlei In- ▶



K-9 Mail erlaubt nun die Signierung (links) und Verschlüsselung (rechts) von E-Mails (Bild 21)

LINKS ZUM THEMA

CyanogenMod
▶ www.cyanogenmod.org

xda-developers
▶ www.xda-developers.com

Heimdall
▶ www.androidnext.de/howto/heimdall-cross-plattform-alternative-zu-odin

ClockWorkMod
▶ www.droidwiki.de/ClockWorkMod

Originale Firmwares für Samsung Galaxy S3
▶ www.android-hilfe.de/original-firmwares-fuer-samsung-galaxy-s3

HashCheck Shell Extension (Windows)
▶ http://code.kliu.org/hashcheck

MD5 Checker (Android)
▶ http://play.google.com/store/apps/details?id=com.fab.md5

7-Zip
▶ www.7-zip.de

Oracle VM VirtualBox
▶ www.virtualbox.org

Kubuntu
▶ www.kubuntu.org

Android SDK
▶ http://developer.android.com/sdk

AV-Test
▶ www.av-test.org

RedPhone, TextSecure
▶ http://whispersystems.org

Linphone
▶ www.linphone.org

SIMSme
▶ www.sims.me

Silent Phone, Silent Text, Silent Contacts
▶ http://silentcircle.com

Telegram
▶ http://telegram.org

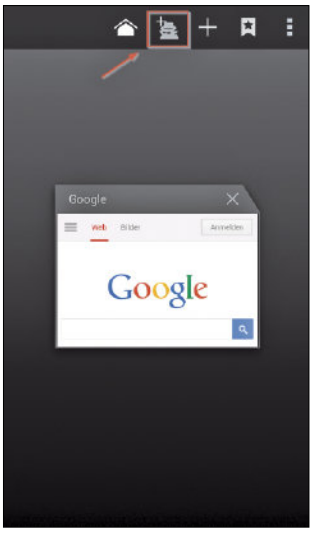
Threema
▶ http://threema.ch/de

Redact Secure Messenger
▶ www.redactapp.com

Wickr
▶ http://wickr.com

Gliph
▶ http://gli.p

Die minimalistische Oberfläche des Dolphin Zero (Bild 22)



Starten einer anonymen Sitzung im Android-Browser (Bild 23)

formationen auf Servern gespeichert. Außerdem muss man bei der Registrierung keine persönlichen Daten wie Benutzernamen, E-Mail-Adresse oder Telefonnummer angeben. Weitere Alternativen sind SIMSme, das zum Zeitpunkt der Erstellung dieses Artikels noch unausgereift wirkt und instabil läuft, sowie Wickr und Gliph. Mit letzterer App lassen sich sogar Bitcoins zwischen den Kommunikationspartnern übertragen.

Manche Messenger bieten die Option, versendete Nachrichten oder Bilder permanent vom Zielgerät zu löschen. Dies darf man aber nicht zu hoch bewerten, da der Empfänger jederzeit einen Screenshot der empfangenen Nachrichten anfertigen und die Vernichtung damit umgehen kann.

Sichere E-Mail

Unverschlüsselte E-Mail stellt noch immer eins der größten Datenschutzrisiken dar, was wohl daran liegt, dass die meisten E-Mail-Programme standardmäßig keine Verschlüsselung unterstützen und die Kommunikationspartner zunächst ihre öffentlichen Schlüssel austauschen müssen. Vergleichsweise einfach können Sie die Verschlüsselung jedoch einrichten, wenn Sie die E-Mail-App K-9 Mail benutzen. Dazu installieren Sie zunächst die App APG, die das Verschlüsselungsformat OpenPGP implementiert. Beim ersten Start können Sie ein neues Schlüsselpaar erstellen oder vorhandene Schlüssel importieren. Anschließend setzen Sie ein Passwort und legen mit Ihrem Namen und Ihrer E-Mail-Adresse eine Nutzer-ID an. Fertigen Sie über *Alle Schlüssel exportieren* und *Alle geheimen Schlüssel exportieren* unbedingt eine Sicherheitskopie der Schlüssel an. Zuletzt öffnen Sie das linke Menü und wählen *Schlüssel Importieren*, um nach den öffentlichen Schlüsseln Ihrer Kommunikationspartner zu suchen und diese zu importieren (Bild 19 und 20).

Installieren Sie nun K-9 Mail und legen Sie Ihr Postfach an. Anschließend navigieren Sie in die Ansicht des Postfachs und wählen *Einstellungen, Kontoeinstellungen, Kryptographie*. Dort aktivieren Sie *Automatisches Signieren und Automatische Verschlüsselung*. Im Folgenden werden alle Mails automatisch verschlüsselt und signiert, sofern der öffentliche Schlüssel des Adressaten bekannt ist (Bild 21).

Sicheres Browsen

Das Browsen im Web stellt keine Kommunikation im engeren Sinne dar, soll hier aber erwähnt werden, weil es datenschutzrelevant ist. Durch Mechanismen wie Cross-Site-Scripting können Angreifer Ihr System kompromittieren und Ihre Daten oder Ihre Identität stehlen. Außerdem sind Website-Betreiber heute in der Lage, Ihr Verhalten im Web sehr genau zu verfolgen und zu verwerten. Diese Risiken können Sie mindern, indem Sie Dolphin Zero einsetzen (Bild 22).

Der Webbrowser der Firma Mobotap befindet sich standardmäßig im anonymen Modus und hinterlässt keine Spuren auf Ihrem Gerät. Zur Suche im Web ist DuckDuckGo vorkonfiguriert – eine auf Anonymität bedachte Suchmaschine, die Ihre Anfragen nicht verfolgt und keine individuellen Daten wie IP oder Browsertyp speichert. Dass der Browser das Recht zum Zugriff auf Ihren Standort benötigt, passt nicht ganz zum Ansatz der Anonymität, ist aber im Zusammenspiel mit CM kein Problem, da der Privacy Guard diesen Zugriff blockiert.

Da Dolphin Zero relativ wenig Komfort beim Surfen im Web bietet, können Sie alternativ auf andere Browser wie den Android-Browser zurückgreifen. Bei diesem sollten Sie aber über *Einstellungen, Erweitert* den Punkt *JavaScript aktivieren* abwählen. Außerdem sollten Sie jede Sitzung im anonymen Modus starten (Bild 23).

Fazit

Das Betriebssystem eines Android-Mobilgeräts zu ersetzen und an eigene Wünsche anzupassen ist nicht ganz trivial. Wer sein Gerät frei von Google-Apps und sonstigem Ballast halten will, kann dies nur durch Aufspielen eines sauberen Betriebssystems erreichen. Will man eine eigene Android-Variante aus dem Quellcode erstellen, ist zudem etwas Übung erforderlich, denn schnell ist am Anfang ein *cd ~/android/system* oder ein *envsetup.sh* vergessen und man wundert sich, warum Befehle nicht ausgeführt werden. Hat man das vorgestellte Vorgehen aber nachvollzogen, ist der wesentliche Schritt zu einem eigenen Android getan. Dabei sind der Austausch von Apps und das Branding erst der Anfang. Sie können eigene Projekte über Eclipse einbinden oder sogar den originalen Android-Quellcode verändern. Alles, was Sie dazu brauchen, ist auf der virtuellen Maschine bereits installiert. Im Hinblick auf ein besonders sicheres Android haben Sie dann sogar die Möglichkeit, bekannt gewordene Sicherheitslecks selbst zu schließen, bevor Google eine Lösung liefert.

Mit Hilfe eines durchdacht konfigurierten CM und Einsatz der richtigen Apps können Sie zudem eine Menge in Richtung Datenschutz tun. Eine klare Empfehlung für die sichere Kommunikation verdienen die Apps RedPhone und TextSecure sowie die Kombination aus K-9 Mail und APG. Vorausgesetzt, dass der Kommunikationspartner ein vergleichbares Setup benutzt, lassen sich damit alle wesentlichen Kanäle verschlüsseln. **[mb]**