

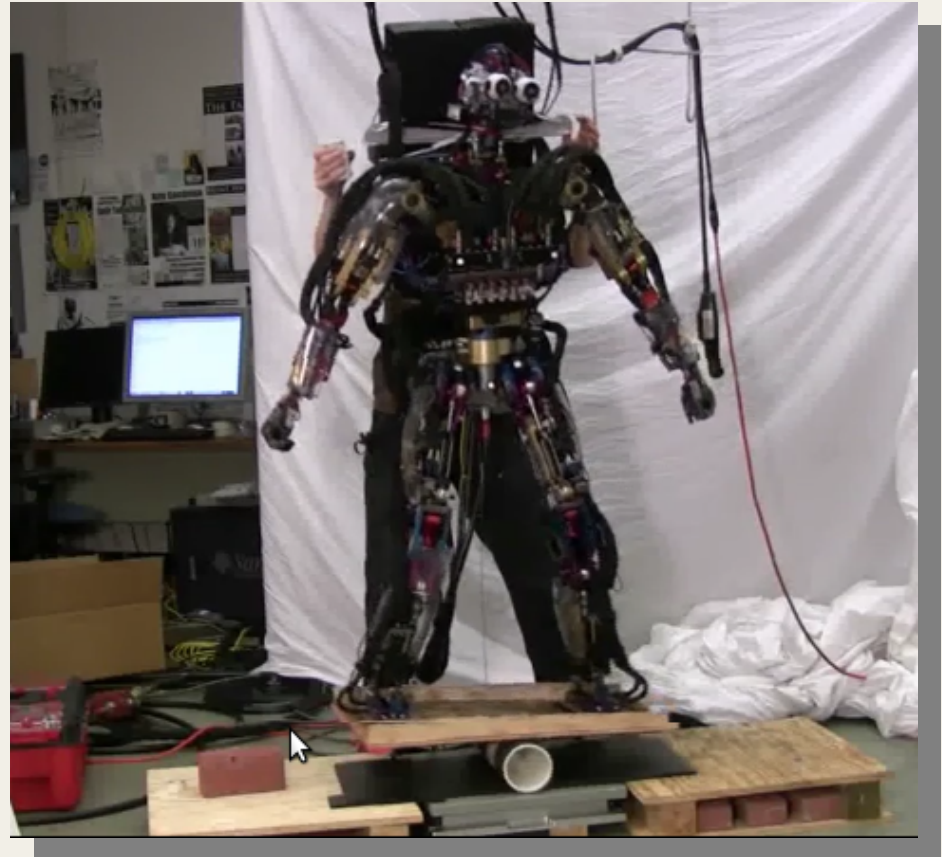
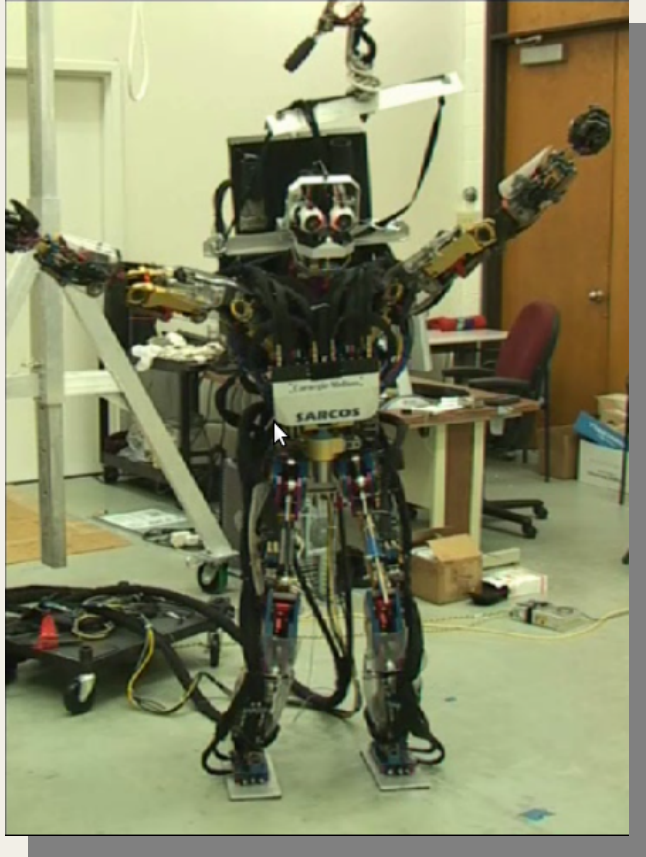
Android Security

Stuart O. Anderson
June 23, 2011

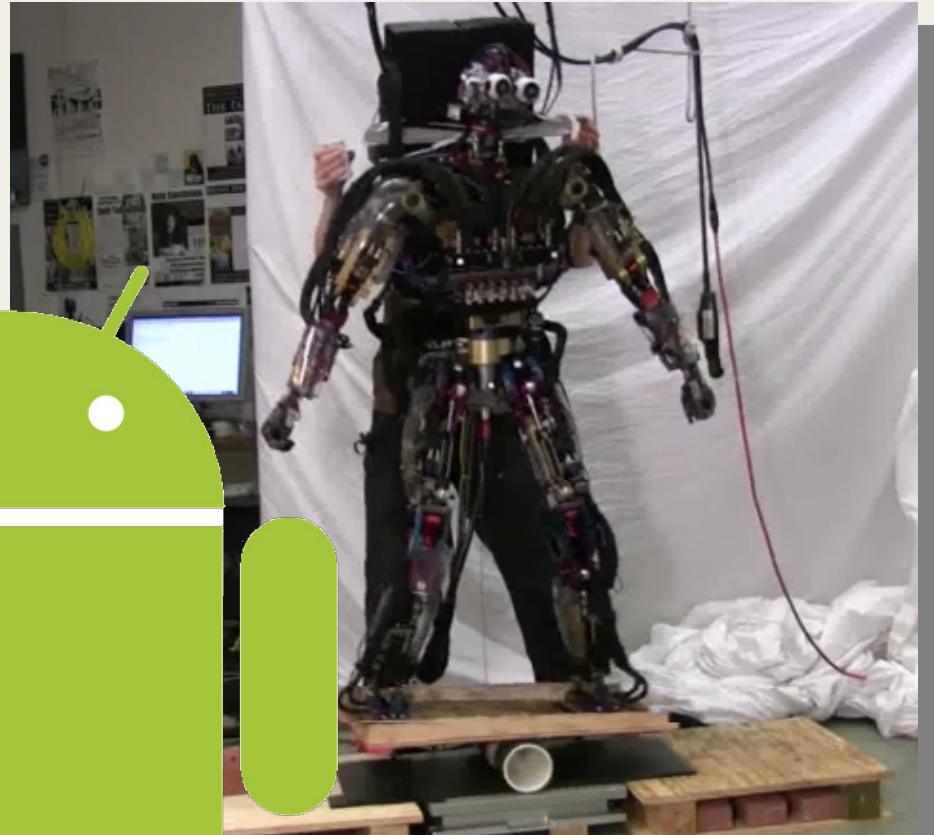
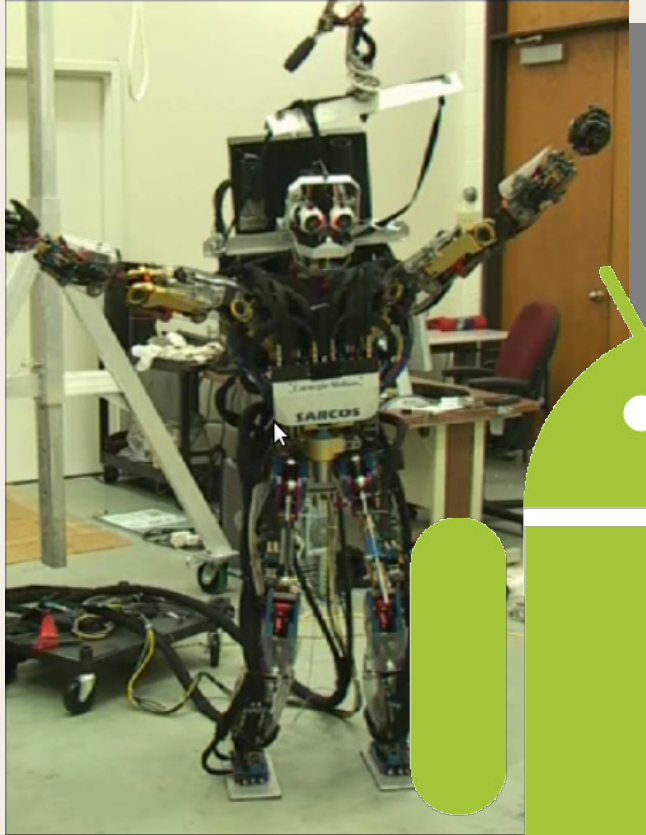
Stuart O. Anderson

- Background in robotics and applied math
- Fellow at the Institute for Disruptive Studies
- Co-founded Whisper Systems with Moxie Marlinspike

Stuart O. Anderson



Stuart O. Anderson



What this talk covers

The Android System

Android's security model

Malware and exploit examples

Best practices for improving security



The Android System: Overview

Android is

- A system architecture
- A business and legal framework

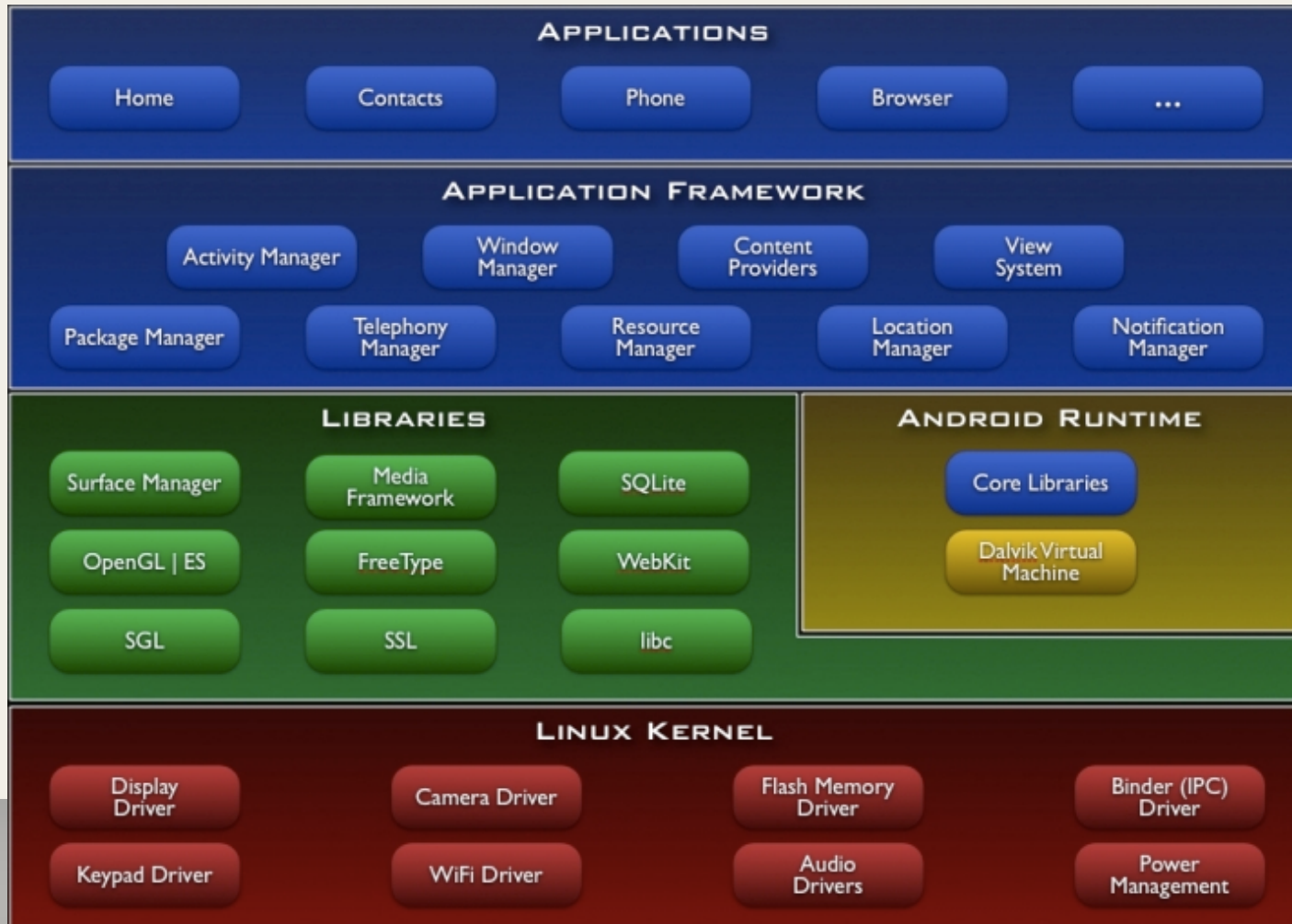
Security is affected by both aspects

Android: Embedded and Consumer

Android's design is driven by

- Resource constraints
 - Memory
 - Battery
- Consumer market
 - iPhone competition

Android System Architecture



Android Kernel

Modified for resource constrained environments

- Binder
- Ashmem and Pmem
- Logger
- Wakelocks
- Out-Of-Memory Handler

Android Userspace

Driven by resource and legal constraints

- Bionic (Non-POSIX libc)
- Prelinked system libraries
- Dalvik VM
- Native Libraries

Bionic: Android libc

BSD License

- No GPL in userspace

Small

- About 200K

Fast

- Especially pthreads

Apriori: Android Prelinker

System libraries are internally pre-linked

Must be loaded at specific vaddr

Look in `/build/core/prelink*.map`

Dalvik

Virtual Machine runs most apps

DEX byte code compiles from Java

Register and not stack based

– i.e. trying real hard not to be a JVM

Native Libraries

Webkit

Media

SQLite

SurfaceManager

...

Android Framework

Components

- Activities
- Services
- Receivers
- ContentProviders

Android Framework

Intents connect components through Binder

- Action
- Data
- Categories
- Extras
- Flags – can grant permissions...

Android: Business Relationships

Google – Develops platform

Chipset vendors – Broad market

OEMs – Shorter time to market

Carriers – Easier to customize

Developers – Easy to publish, free SDK

OEMs

Chipset vendors are limited

- Qualcomm, TI (OMAP3), Ericsson, Broadcom
- Faster development cycle (9-12 months) for OEMs
- Budget goes to differentiation

Carriers

Slow updates

- Known webkit bugs linger
 - M.J. Keith at Alert Logic

Google's Points of Control

- Access to latest source code
- Control of review process
- Proprietary apps (Market, Maps, ...)
- Trademark
- AFA, CTS/CDD

Orphaned Devices

Last Google I/O

- 18 months support for new devices
- Verizon, HTC, Samsung, Sprint, Sony Ericsson, LG, T-Mobile, Vodafone, Motorola, and AT&T

Android: Future Directions

New Devices

- Tablets
- Readers
- PCs / Dockables

Android's Security Model

Linux Kernel

- Process separation
- Access to resources by UID/GID

Android Framework

- Signed packages
- Per-package Permissions

Android UID and GID

Most packages have their own UID

Some share a UID

GID is used for Kernel level resources

- Camera, bluetooth, display, ...

Android UID and GID

```
app_49    384    86    114796 33796 ffffffff 00000000 S com.android.launcher
app_37    385    86    94468  16152 ffffffff 00000000 S com.android.voicedialer
app_10    410    86    97044  19312 ffffffff 00000000 S com.android.vending
app_8     428    86    119840 23376 ffffffff 00000000 S com.google.process.gapps
app_27    480    86    97624  20496 ffffffff 00000000 S android.process.media
app_48    674    86    102452 20256 ffffffff 00000000 S com.google.android.apps.genie.genie
app_26    686    86    97912  17880 ffffffff 00000000 S com.android.quicksearchbox
app_36    725    86    96092  18176 ffffffff 00000000 S com.cooliris.media
app_41    737    86    120740 22184 ffffffff 00000000 S com.google.android.apps.maps
app_18    764    86    103200 20160 ffffffff 00000000 S com.google.android.voicesearch
app_12    824    86    94336  15836 ffffffff 00000000 S com.whispersys.updater
app_9     832    86    97516  16112 ffffffff 00000000 S com.whispersys.monitor
```

Android Framework Security

Code Signing

- Links a package to a developer

Permissions

- Grants a package a capability

Code Signing

Packages are signed when published

- You trust the publisher with the security of their private key
- If the keys don't match, app must be manually removed and reinstalled
- Packages that share keys can share UIDs

Remote Pull and Push

Google can add and remove packages

- GtalkService
- Malware may attempt to disable these features

Permissions

Every UID has an associated set of permissions it has been granted

android.permission.SEND_SMS

android.permission.WRITE_CALENDAR

android.permission.READ_PHONE_STATE

Permissions

Packages request permissions in their manifest

User is prompted to approve these permissions as a single block

- Only once, at install time
- Permissions not marked 'dangerous' are not displayed

Permissions

Most permissions declared in

- /core/res/AndroidManifest.xml

Not all permissions require user approval

- Signature
- SignatureOrSystem

Permissions: Granularity

Granularity in the permissions themselves

- Internet is a single permission

Granularity in user control

- Can't approve a subset of the requested permissions

Permissions: Granularity

Too fine granularity overloads users

Overloaded users stop paying attention

Permissions: Enforcement

Permission checks are performed in
PackageManagerService

```
public int checkUidPermission(String permName, int uid) {
    synchronized (mPackages) {
        Object obj = mSettings.getUserIdLP(uid);
        if (obj != null) {
            GrantedPermissions gp = (GrantedPermissions)obj;
            if (gp.grantedPermissions.contains(permName)) {
                return PackageManager.PERMISSION_GRANTED;
            }
        } else {
            HashSet<String> perms = mSystemPermissions.get(uid);
            if (perms != null && perms.contains(permName)) {
                return PackageManager.PERMISSION_GRANTED;
            }
        }
    }
    return PackageManager.PERMISSION_DENIED;
}
```

Permissions: Services

Services must explicitly check permissions at IPC entry points

```
public void call(String number) {  
    // This is just a wrapper around the ACTION_CALL intent, but we still  
    // need to do a permission check since we're calling startActivity()  
    // from the context of the phone app.  
    enforceCallPermission();  
  
    String url = createTelUrl(number);  
    if (url == null) {  
        return;  
    }  
  
    Intent intent = new Intent(Intent.ACTION_CALL, Uri.parse(url));  
    intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);  
    intent.setClassName(mApp, PhoneApp.getCallScreenClassName());  
    mApp.startActivity(intent);  
}
```

Permissions: ContentProviders

Read and Write permissions handled by system

Must implement per-URI permission granting

Malware and Exploit Examples

Getting root

Remote exploits

Protocol weaknesses

Making money

Leaky Apps

Content Providers, SD/Card

Network communication

- Spoofed http responses
- Authtokens

Unreliable deputies

GSM Weaknesses

Well publicized attacks on GSM

– See Karsten Nohl

The cost of intercept equipment is marginal

Privilege Elevation

Send an Intent or Binder data to another app that causes unexpected behavior

- Some critical services have very complicated interfaces

Change your own uid or gid

- Kernel, zygote, etc

Android Exploit Examples

Sebastian Kraemer (stealth)

- Zimperlich
 - Forkbomb to process limit
 - Zygote will fail to change uid from root on fork
- Gingerbreak
 - Unchecked array index in vold
 - Rewrite GOT entry for strcmp()

Android Remote Exploit Examples

Colin Mulliner

- NFC remote application crash
- NFC remote NFC service crash

Charlie Miller

- PacketVideo media library

Malware Threats

Jon Oberheide

- Rootstrap
- Download and execute exploits as they become available

Malware Threats

Untargeted Monetization

- Premium SMS
- 1-900 Numbers

Persistence

- Remount /system r/w
- Turn off AV tools

Solutions and Best Practices

System Level Changes

Security Applications

Auditing Applications

System Level Changes

Full disk encryption

Dynamic egress filtering

Selective permissions

Extended code signing

Disk Encryption

Honeycomb

- MTD devices only
- Tied to screen lock

WhisperCore

- yaffs variant supports MTD and block devices
- Enhanced screenlock

Dynamic Information Flow Tracking

DIFT inside the Dalvik VM

TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones

- *William Enck, Peter Gilbert, Byung-gon Chun, Landon P. Cox, Jaeyeon Jung, Patrick McDaniel, and Anmol N. Sheth. In Proc. of the USENIX Symposium on Operating Systems Design and Implementation (OSDI), October 2010 in Vancouver*

Dynamic Information Flow Tracking

Variable tracking in Dalvik

Message tracking in Binder

Method tracking in system libraries

File tracking via file-system extension

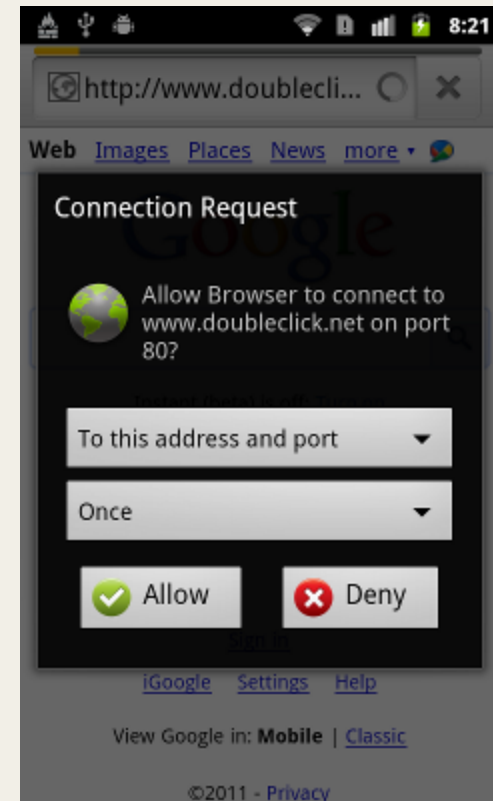
Dynamic Egress Filtering

Monitor outgoing network connections.

Filter connections by:

- Initiating app.
- Destination.
- Network type and location.

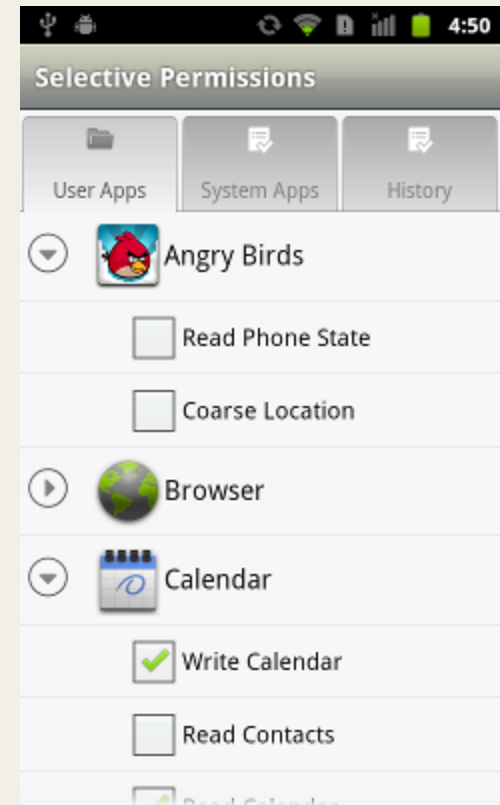
Prompts when connections are initiated



Selective Permissions

Remove specific permissions

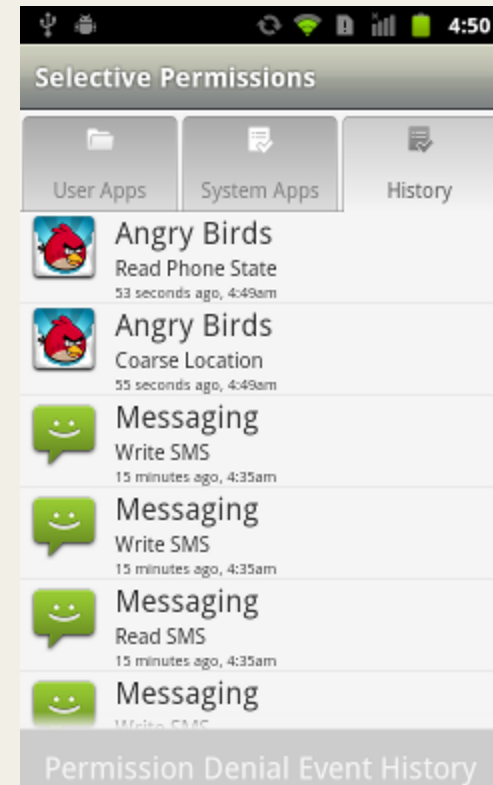
Create temporary and isolated copies of requested resources



Selective Permissions

Remove specific permissions

Create temporary and isolated copies of requested resources



Exploit Mitigation

Address Space Randomization for Mobile Devices

- *Hristo Bojinov, Dan Boneh, Rich Cannings, Iliyan Malchev – WiSec 2011*
- Randomizes addresses
 - Even with prelinked libraries
- Android moving to ld.so
- Still forking zygote?

Extended Code Signing

Management of which apps can run

- Whitelist or blacklist
- Installed apps can be blocked

Lets administrators sign, update, install,
and remove apps remotely

Security Applications

Secure backup

Secure communications

Secure storage

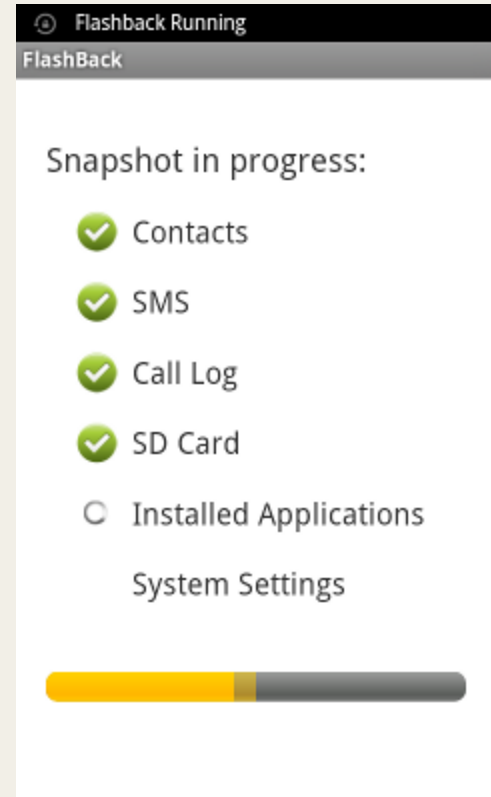
Secure Backup

Secure incremental backup

Cloud or backend storage

Remote image management

Remote wipe



Secure Communication

Voice calls

- VoIP solutions: RedPhone, PrivateWave, Cellcrypt

Messaging

- SMS/MMS/IM

Email

- Good, MobileIron, TouchDown

Malware Detection

Google

- Can remove malware from Market
- Can remotely disable and update

Lookout

- At the endpoint, limited access
- Can be disabled by malware

Auditing an Application

Examine the Manifest

Decompilers

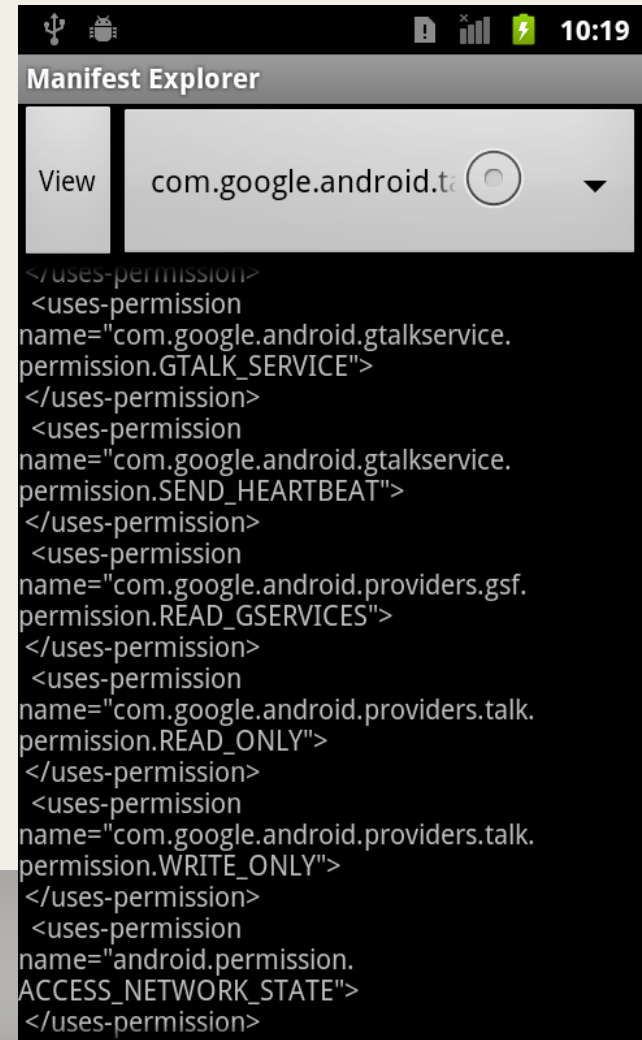
Other Tools

Auditing an Application

ISEC's

Manifest Explorer

Author: Jesse Burns



```
</uses-permission>
<uses-permission
name="com.google.android.gtalkservice.
permission.GTALK_SERVICE">
</uses-permission>
<uses-permission
name="com.google.android.gtalkservice.
permission.SEND_HEARTBEAT">
</uses-permission>
<uses-permission
name="com.google.android.providers.gsf.
permission.READ_GSERVICES">
</uses-permission>
<uses-permission
name="com.google.android.providers.talk.
permission.READ_ONLY">
</uses-permission>
<uses-permission
name="com.google.android.providers.talk.
permission.WRITE_ONLY">
</uses-permission>
<uses-permission
name="android.permission.
ACCESS_NETWORK_STATE">
</uses-permission>
```

Auditing an Application

Use adb to pull the apk from the phone

```
adb pull /data/app/packageName.apk
```

```
adb pull /system/app/packageName.apk
```

Auditing an Application

Use dedex (Nathan Keynes) and jd-gui to inspect DEX code

```
unzip package.apk  
dedex classes.dex  
jd-gui classes.jar
```

JD-GUI

File Edit Navigate Help



DecisionActivity.class

```
54  initializeResources();
    }

    public void onResume()
    {
59      super.onResume();
60      redraw();
    }

    public void onStop()
    {
65      int k = 0; String str3 = "Allow outgoing connection?"; super.onStop();

67      boolean bool = this.verdictRegistered; if (!(bool)) {
68      String str1 = "notification"; Object localObject1 = getSystemService(str1); localObject1 = (NotificationManager)loc
69      Notification localNotification = new android/app/Notification; int i = 2130837513; String str2 = "Allow outgoing co

71      Intent localIntent = getIntent(); PendingIntent localPendingIntent = PendingIntent.getActivity(this, k, localIntent

73      int j = localNotification.flags; j |= 16; localNotification.flags = j;
74      Object localObject2 = "Allow outgoing connection?"; localObject2 = this.label; localObject2 = ((TextView)localObjec
75      ((NotificationManager)localObject1).notify(k, localNotification);
    }
}

private class DenyClickListener
implements View.OnClickListener
{
    DenyClickListener(, DecisionActivity.1 param1)
    {
143      this(paramDecisionActivity); }

145      public void onClick() { Object localObject1 = new android/content/Intent; Object localObject2 = this.this$0; WhisperM
146      localObject2 = "com.whispersys.monitor.VERDICT_ACTION"; ((Intent)localObject1).setAction((String)localObject2);
```

Other Audit Tools

- Dynamic Information Flow Tracking
 - TaintDroid
- Mandatory Access Control
 - TOMOYO Linux
- Emulator
 - Scott Dunlop's JDWP->JDP method
- Network Monitoring
 - WhisperMonitor
 - Wireshark

Summary: Android Security

Embedded and consumer

Tradeoffs made against security

Divided responsibility for security

System and application layer solutions