GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN
Computer Security Group

Smartphone Security

Computer and Network Security

Hugo Gascón
Daniel Arp
Computer Security Group
Georg-August-Universität Göttingen

Summary

In this lecture...

- Evolution of mobile devices and markets
- Security architecture of modern platforms
- Vulnerabilities and attacks in smartphones
- Security countermeasures and current research

Evolution of Mobile Devices & Markets

Evolution of Mobile Phones

From DynaTAC to iPhone

1983

Motorola DynaTAC 8000X

- Analog Networks
- 30 min conversation

1992

Nokia 1011

- First mass-produced GSM phone
- SMS send/recv

2002

Nokia 3510

- GSM 900/1800
- GPRS enabled
- WAP, MMS
- Java Apps

2011

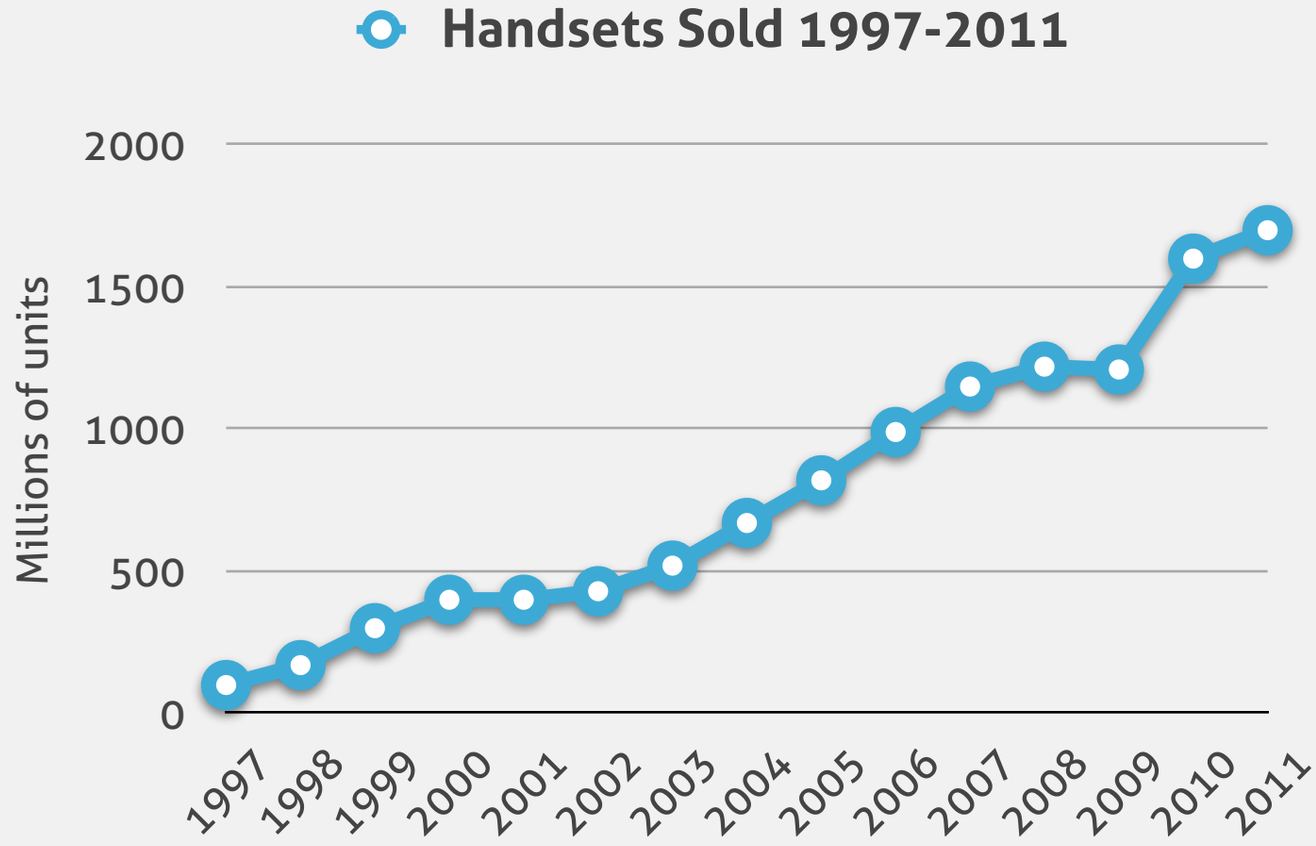
iPhone 4S

- CPU 800 Mhz Dual Core
- 64 GB storage
- 512 Mb RAM
- WIFI (802.11 b/g/n)
- GSM/GRPRS/UMTS
- ....

Mobile Market Evolution

Some interesting numbers & trends

- **2011**: 6k million mobile connections
- **2011**: 141$ million mobile payment
- **2012**: >50% of mobile communication devices are smartphones
- **2015**: More mobile internet users than wireline users
- **2016**: 44 billion mobile app downloads

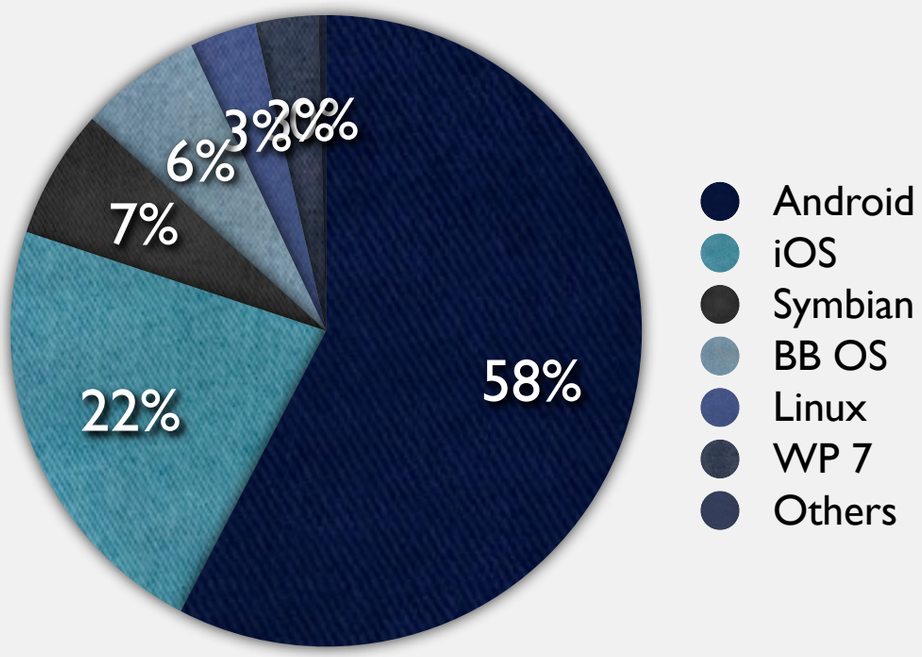


Smartphone Stats

The current picture

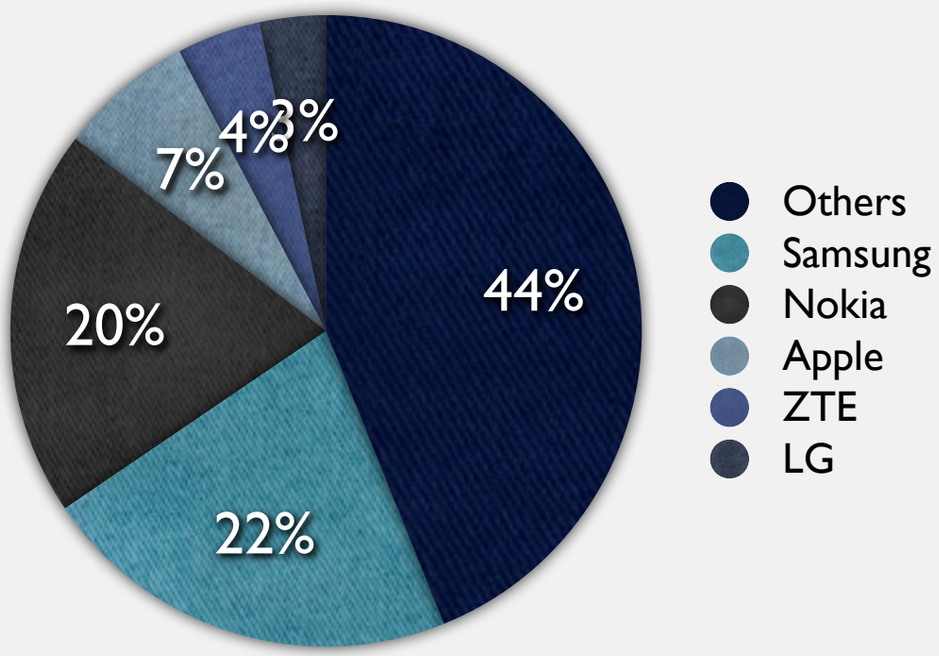
Smartphone: *high-end mobile phone built on a mobile computing platform, with more advanced computing ability and connectivity than a feature phone (Wikipedia).*

Operative Systems



Source: IDC (Q1 2012)

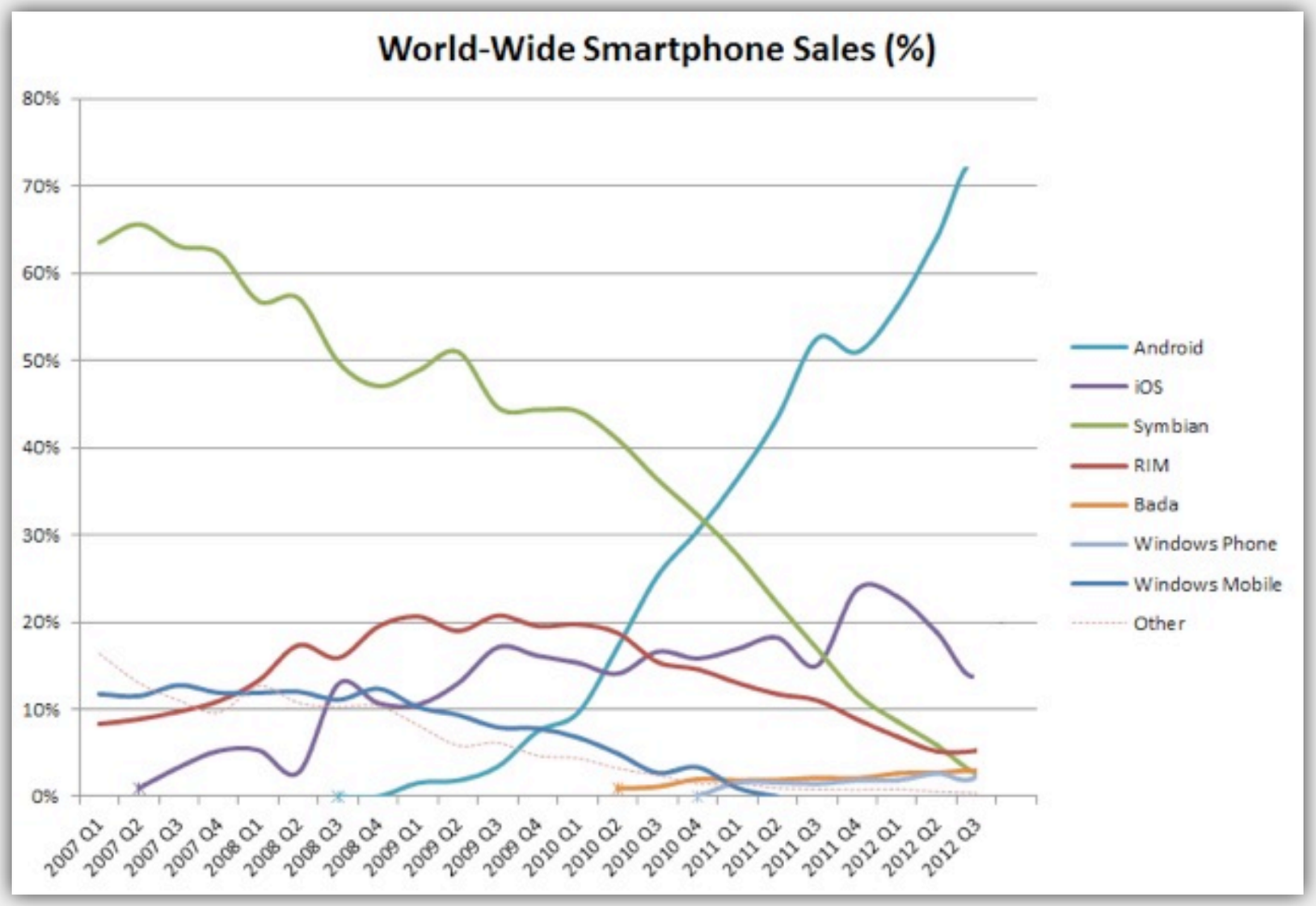
Manufacturers



Source: Gartner (Q2 August 2012)

Smartphone Stats

The current picture



Applications Markets

Different approaches, different risks

Apple App Store

- ~700.000 apps (Sep'12)
- iPhone users can only install applications from the App Store
- Applications are tested by Apple
- 25.000 mill. apps downloaded (March' 12)

Android Market

- ~675.000 apps (Oct'12)
- Android users can install applications from any source
- Apps are checked by an automatic security system known as "Bouncer"
- 25.000 mill. apps downloaded (Sep'12)

Some Conclusions

Evolution of mobile devices and markets

- **Convergence** of mobile phones and personal computers in power and capabilities
- A real and fast shift towards **mobility** in computing devices and online services
- Application **markets** as a standard software distribution system for new mobile environments
- Rise of **new paradigms** in software, services and threats

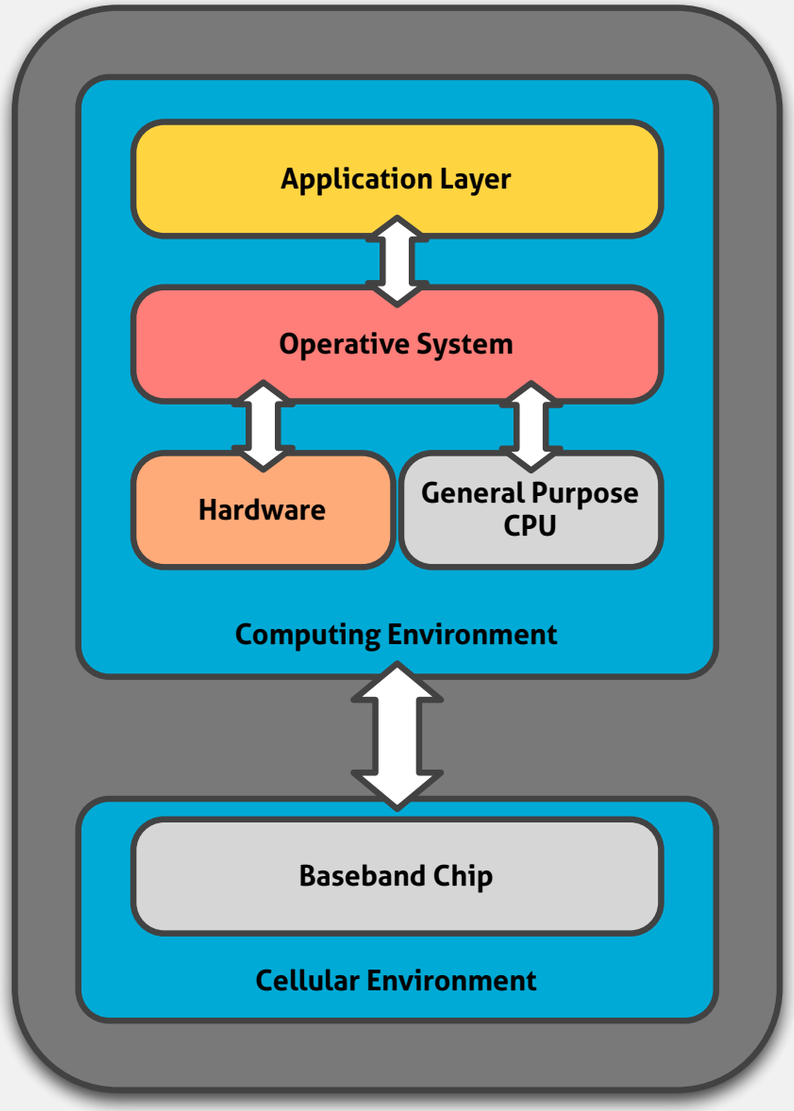
- **Location based**
- **Mobile banking / payments**
- **User behavior / sensor based**
- **Mobile corporate services**

Security Architecture of Modern Platforms

Classic Architecture in Smartphones

2 devices in 1

- Combination of two environments in one device:
  - **Cellular**
  - **Computing**
- Mobile communication security deals with protection of the modem and the mobile network
- We focus on security threats and defenses at the computing environment



Apple iOS

One more thing....

Application Development & Code Signing

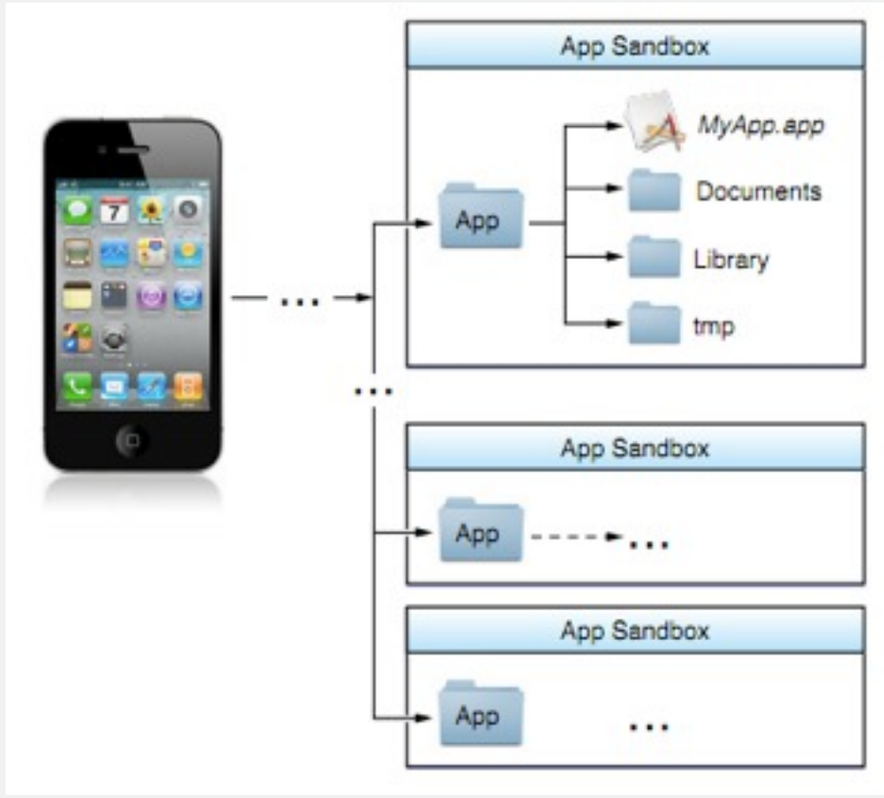
- iOS applications are written in Objective-C, using Cocoa API and Touch Framework
- iOS applications must be signed by a valid code-signing certificate
  - **Code Signing Request (CSR)**

Apple iOS

One more thing....

Kernel Protection & Permissions

- **Seatbelt:** kernel sandboxing mechanism
  - Based on policy files that describe what system permissions an application should have
  - Prevent malicious apps from reading data of other apps or modifying the system

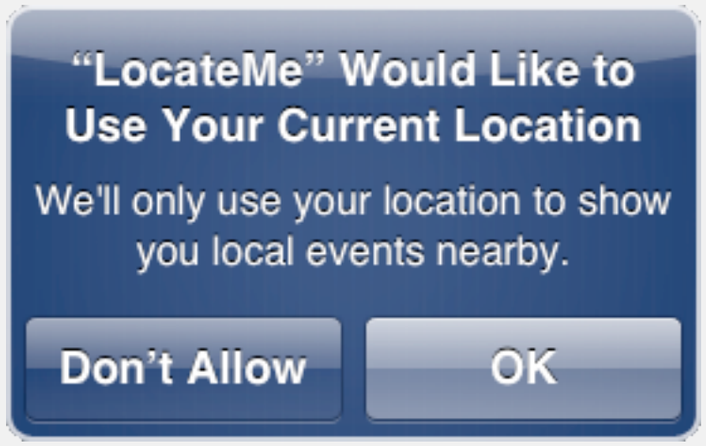


Apple iOS

One more thing....

Kernel Protection & Permissions

- **Seatbelt:** kernel sandboxing mechanism
  - Based on policy files that describe what system permissions an application should have
  - Prevent malicious apps from reading data of other apps or modifying the system
- Exploit Mitigation: **ASLR** since iOS 4.3
- **Permissions**: granted for specific functionality by user on running time



Windows Phone

Developers...

Application Development & Code Signing

- Current version is WP8
  - **Released October 29, 2012**
  - **Based on the Windows NT kernel**
- Applications are written in C#.NET and VB.NET using Visual Studio and Silverlight Platform
- All application binaries must include digital signatures signed by Microsoft in order to run

Windows Phone

Developers...

Kernel Protection & Permissions

- Trusted boot loaders
  - **Provisioning the hash of the public key used to sign the initial boot loaders**
  - **Unique device and Microsoft keys embedded in chip (UEFI)**
- Chamber model
  - **Security boundaries defined in 4 chamber types**
- Capabilities
  - **Expressed in application manifest**
  - **Disclosed on Marketplace**

Google Android

Testing ground

Overview

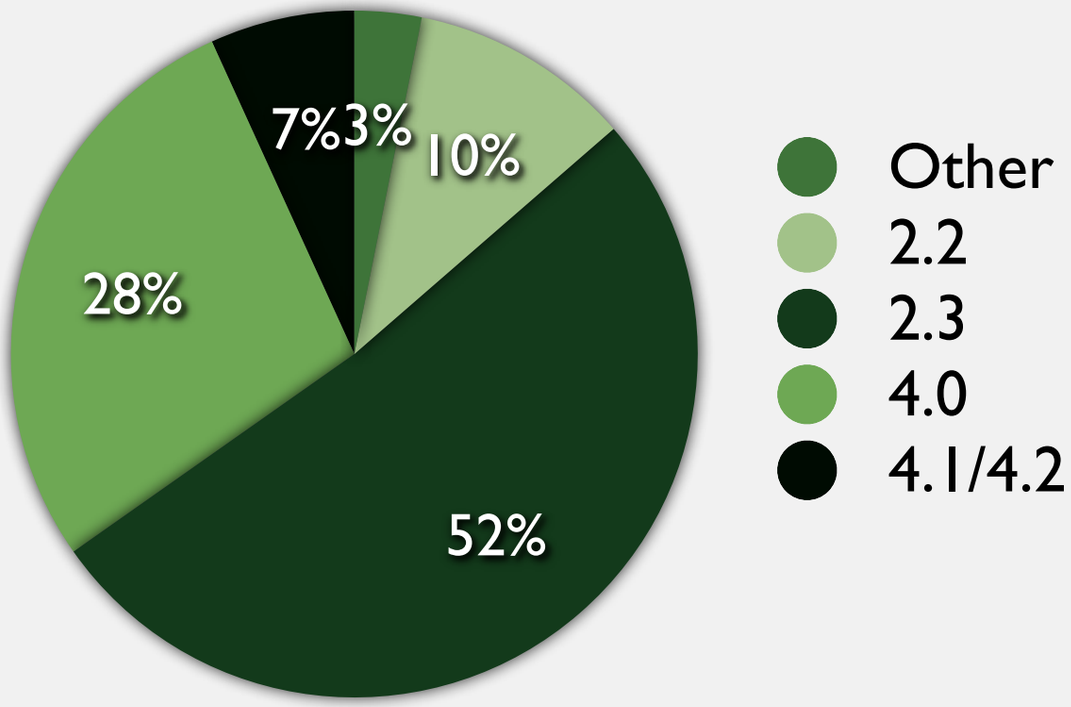
- Developed by OHA led by Google
- Open-source OS Based on Linux Kernel
- Most popular OS for smartphones

Google Android

Testing ground

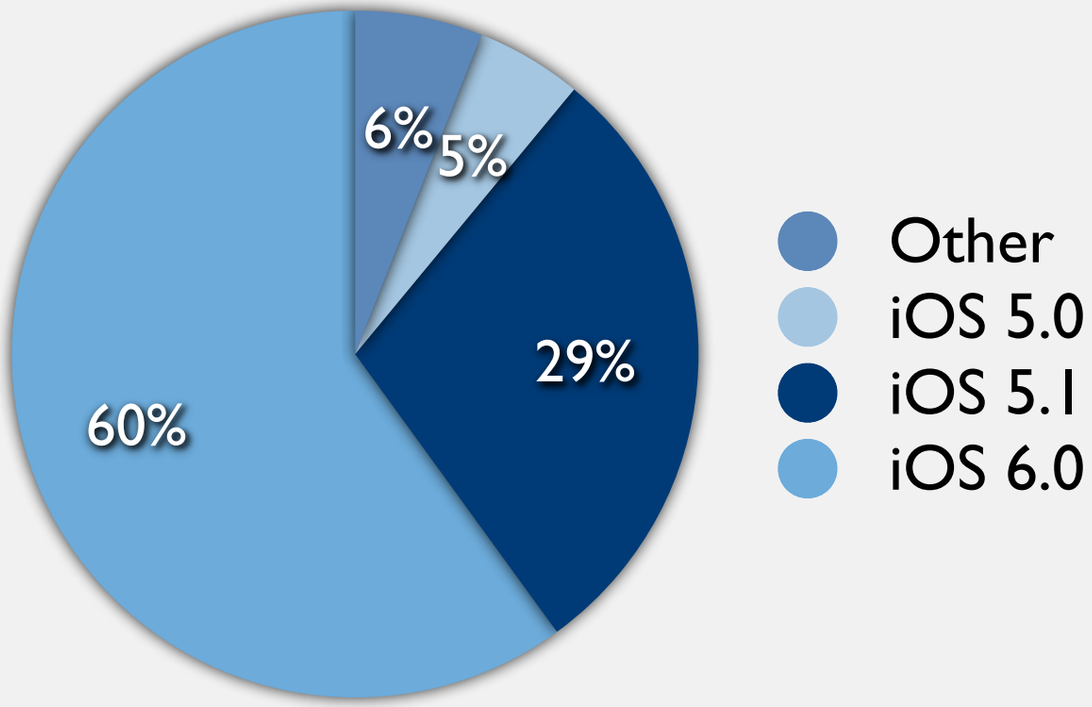
Fragmentation

Android



Source: Heise (Q4 2012)

iOS



Source: Chitika (Q4 2012)

Google Android

Testing ground

GooglePlay Store

- Application distribution through the GooglePlay Store
- Uploaded Apps are analyzed by Google Bouncer
- Real-time scanning of third-party apps (since Android 4.2)
- Remote uninstall of malicious software

Google Android

Testing ground

Code Signing

- Each App needs to be signed by developer
- Otherwise it is rejected by Store/Package Installer
- Android doesn't perform CA authentication
  - Apps can be self-signed

Google Android

Testing ground

Application Development

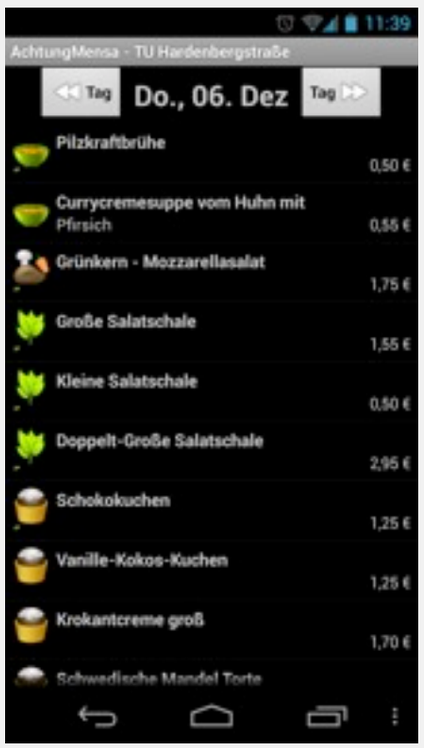
- Applications written in Java and run in the Dalvik VM
- Native code programming in C/C++ is possible using Native Development Kit

Google Android

Testing ground

Application Components

- Apps consist of multiple components
- Component types:
  - Activity
  - Service
  - BroadcastReceiver
  - ContentProvider



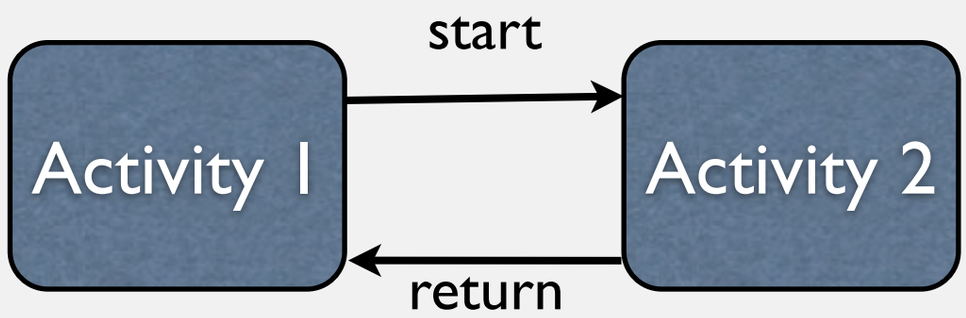
Example: Activity

Google Android

Testing ground

Interprocess Communication (IPC)

- Apps and App Components communicate using intents
- Explicit Intents: Receiver is specified

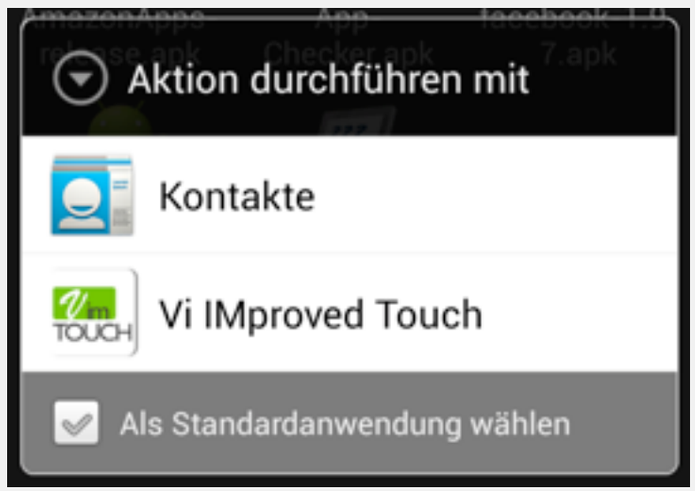


Google Android

Testing ground

Interprocess Communication (IPC)

- Apps and App Components communicate using intents
- Implicit Intents: Android System finds component which is suitable for the intent



Google Android

Testing ground

Sandboxing

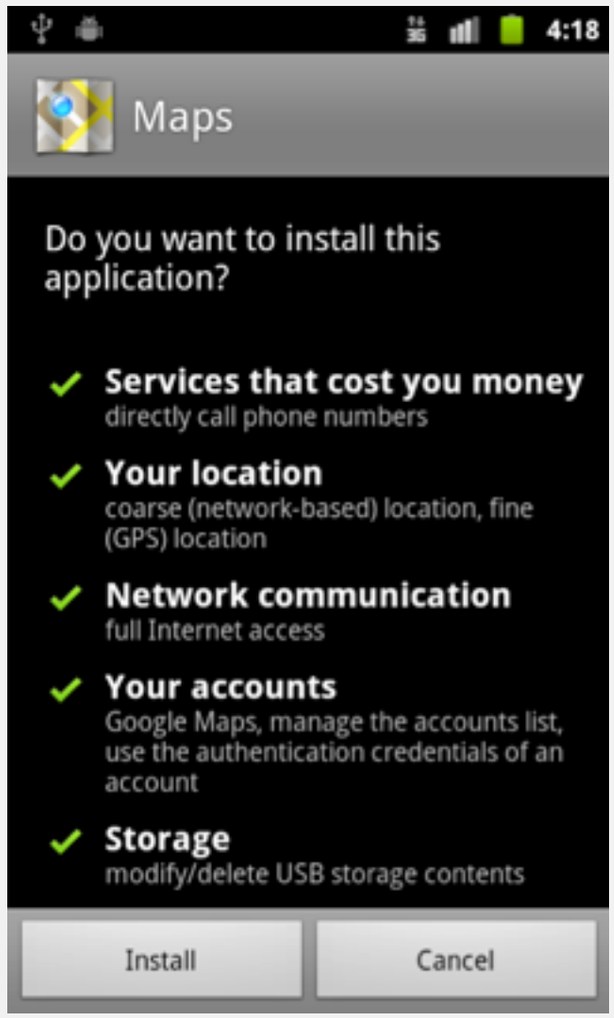
- Dalvik and native apps run in application sandbox
- Each application runs as a separate process which has its own user ID
- Apps can only access their own files
- Apps are isolated from each other, except IPC

Google Android

Testing ground

Permissions

- Sensitive API calls are protected by permissions
- Permissions granted at install time by the user
- Permissions are not modifiable afterwards



Google Android

Testing ground

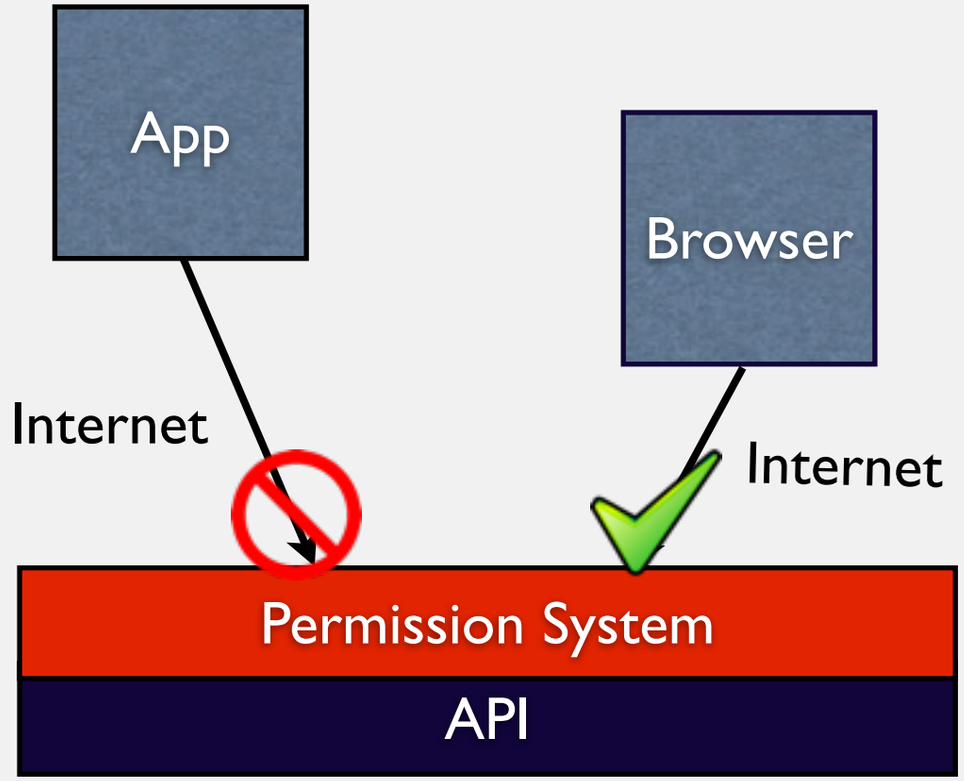
Security Leak: Permission Re-Delegation

- App1 has no permission to access certain resource
- But App2 has the required permission and a public interface
- App1 can gain additional privileges through App2

Google Android

Testing ground

Security Leak: Permission Re-Delegation



Google Android

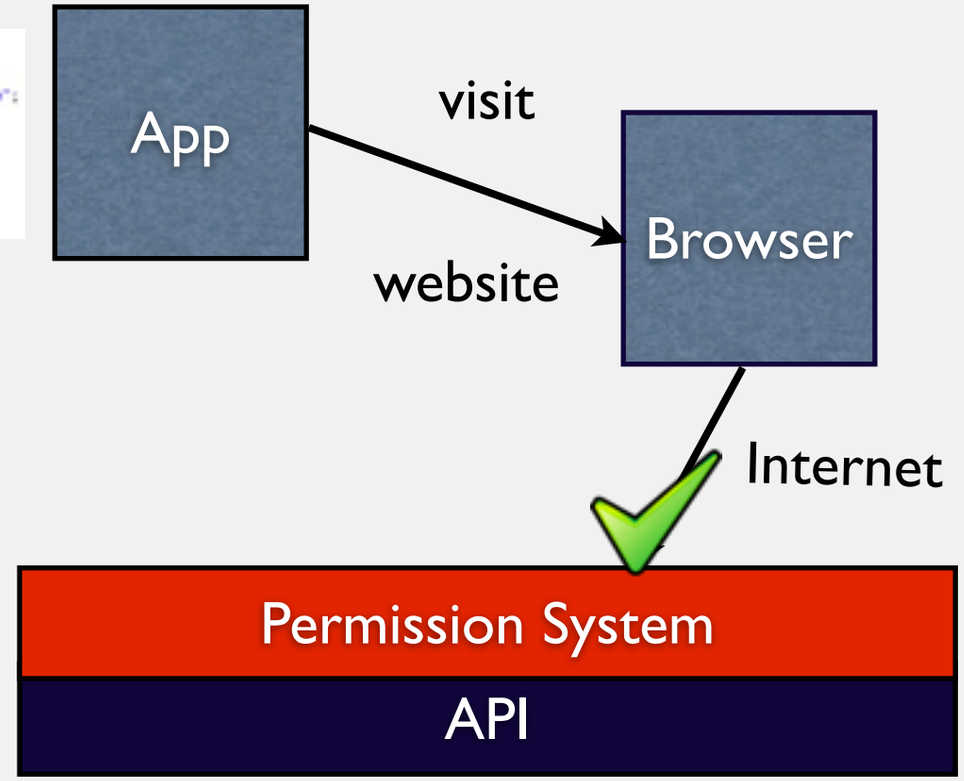
Testing ground

Security Leak: Permission Re-Delegation

```
public void sendMessage(View view) {

    Intent intent = new Intent();
    String urlString = "http://www.very-malicious-site.com";
    Uri intentUri = Uri.parse(urlString);

    intent.setAction(Intent.ACTION_VIEW);
    intent.setData(intentUri);
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    startActivity(intent);
}
```



Google Android

Testing ground

Further Security Features

- Memory management security enhancements
  - ASLR, DEP
- Filesystem Encryption (since Android 3.0)

Security Threats In Smartphones

Application-Based Threats

Beware of app

Malware

- Motivation & Profit
  - Diallerware/SMS attacks
  - Financial malware (e.g. mTAN stealing)
  - Spyware/Surveillance
  - Search Engine Poisoning/PPC
  - Botnets

Application-Based Threats

Beware of app

Malware

- Attack Vectors
  - Application Markets
  - Third-Party Repositories (repackaging)
  - Update attacks
  - Malvertising

Application-Based Threats

Beware of app

Malware

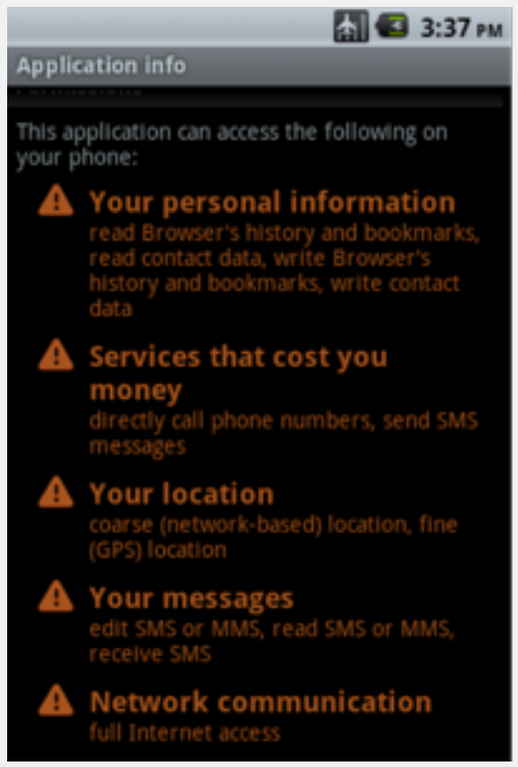
- Vulnerabilities & Exploits
  - Poor application isolation
  - Privilege escalation attacks
  - Insecure data and shared storage

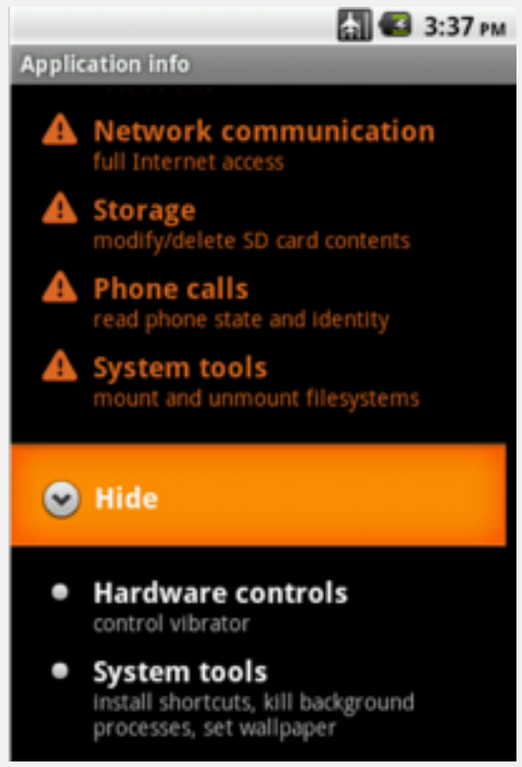
Application-Based Threats

Beware of app

Malware Example: Android/Geinimi

- Malicious code inserted into legitimate apps
- Additional permissions requested
  - Read/write SMS
  - Read contacts
  - Access GPS
  - Make phone calls
  - ...





Application-Based Threats

Beware of app

Malware Example: Android/Geinimi

- Encryption of network traffic

```
himi?|?{?w¿NF??n?!z??wkg0@Q&??%??????*E??WC{???I\?5
                                                ??^?{?
/?b???3????????悲+%????<d????i???
HT?S?+???<"?hP?????zF??]>??
```

- Receives commands from C&C-Server in XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Root>
        <Action>
                <CmdID>2</CmdID>
                <AdID>12</AdID>
                <sms>5555665688;lookout</sms>
        </Action>
</Root>
```

- Sends premium sms

Application-Based Threats

Beware of app

Malware Example: DroidKungFu Trojan

- tries to gain root access using multiple exploits
- evolved over time

	Version 1	Version 2	Version 3
Discovery Date	June 2011	July 2011	August 2011
Embedded Root Exploits	Exploid RageAgainstTheCage	Exploid RageAgaintTheCage	Exploit RageAgainstTheCage
C&C Server	1 (hardcoded in Java as plaintext)	3 (hardcoded in native code as plaintext)	3 (encrypted as ciphertext)

Web-Based Threats

Risky browsing

- Phishing scams

Web-Based Threats

Risky browsing

- Phishing scams
- Drive-by-downloads
  - **Automatic download of apps when visiting a web page**
- Browser exploits
- Web portals bad implementation
  - **Lack and mix of HTTP/HTTPS**

Physical Threats

Hide & seek

- Lost or stolen devices
  - **Lack of full encryption**
  - **Vulnerable locks**
- Improper decommissioning
  - **Devices disposed or transferred to another user without removing sensitive data**
- Lack of multiple user accounts
  - **People share smartphones**
  - **Devices are multi environment**

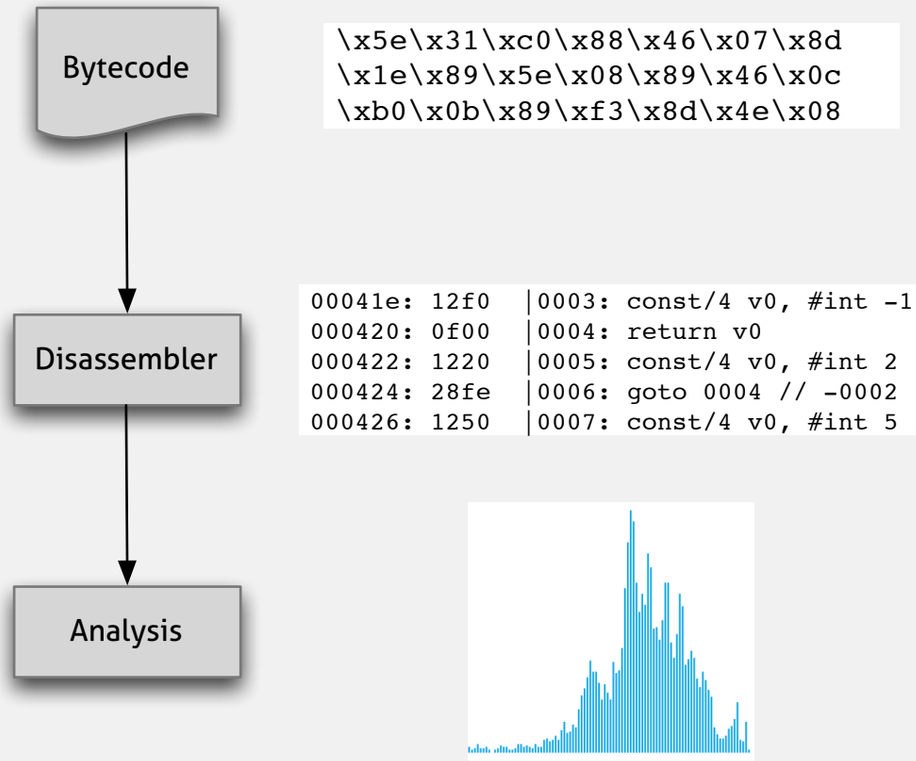
Security Measures in Smartphones

Malware Detection & Mitigation

Hunting season

Static Analysis

- System call based
  - Disassemble application
  - Extract system calls
  - Anomaly detection

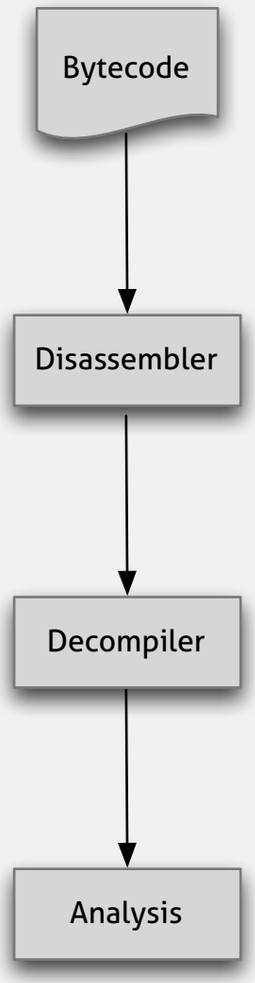


Malware Detection & Mitigation

Hunting season

Static Analysis

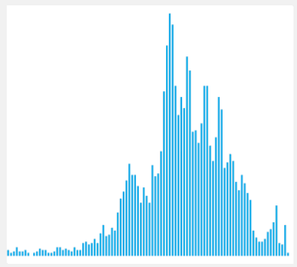
- Source code based
  - **Decompile application**
  - **Static code analysis**
  - **Anomaly detection**




```
\x5e\x31\xc0\x88\x46\x07\x8d
\x1e\x89\x5e\x08\x89\x46\x0c
\xb0\x0b\x89\xf3\x8d\x4e\x08
```

```
00041e: 12f0   |0003: const/4 v0, #int -1 // #ff
000420: 0f00   |0004: return v0
000422: 1220   |0005: const/4 v0, #int 2 // #2
000424: 28fe   |0006: goto 0004 // -0002
000426: 1250   |0007: const/4 v0, #int 5 // #5
```

```
Intent localIntent1 = new Intent
("android.intent.action.SEND", localUri2);
Intent localIntent2 =
localIntent1.addFlags(524288);
String str =
this.activity.getResources().getString(2131230758);
```

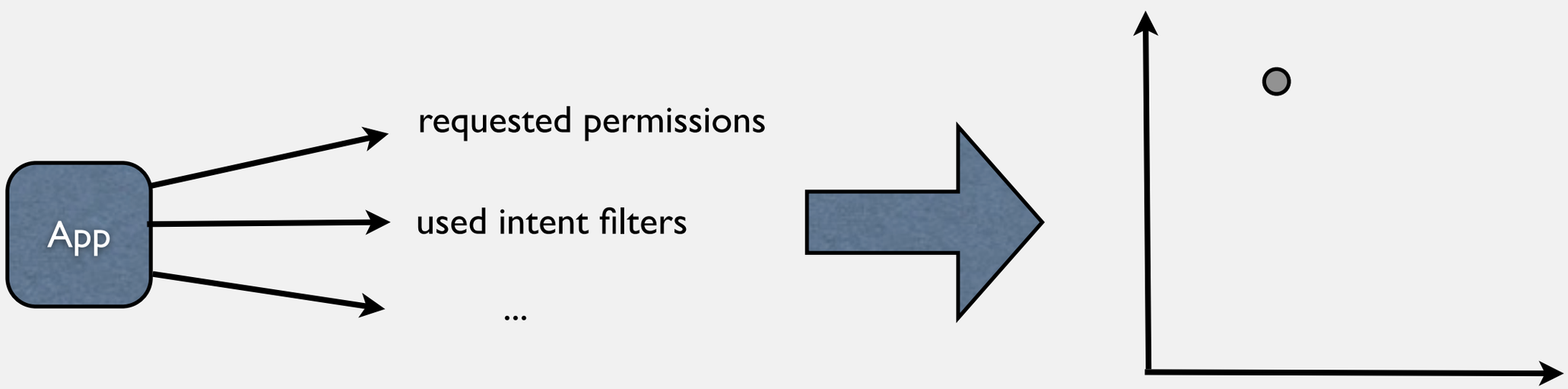


Malware Detection & Mitigation

Hunting season

Research: Static Analysis & Machine Learning

- Static analysis of app
- Transfer app features into vector space

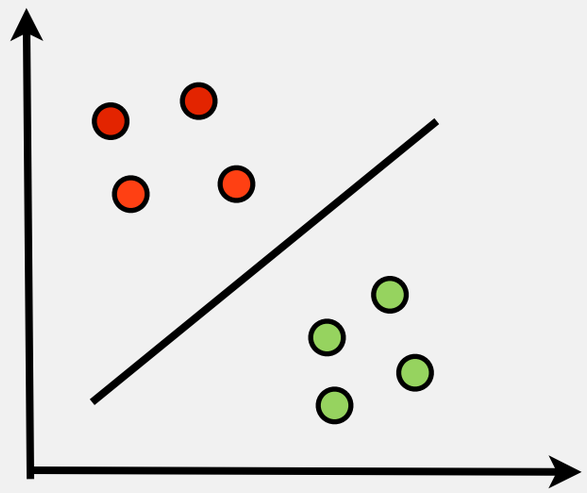


Malware Detection & Mitigation

Hunting season

Research: Static Analysis & Machine Learning

- A Support Vector Machine learns difference between benign and malicious apps

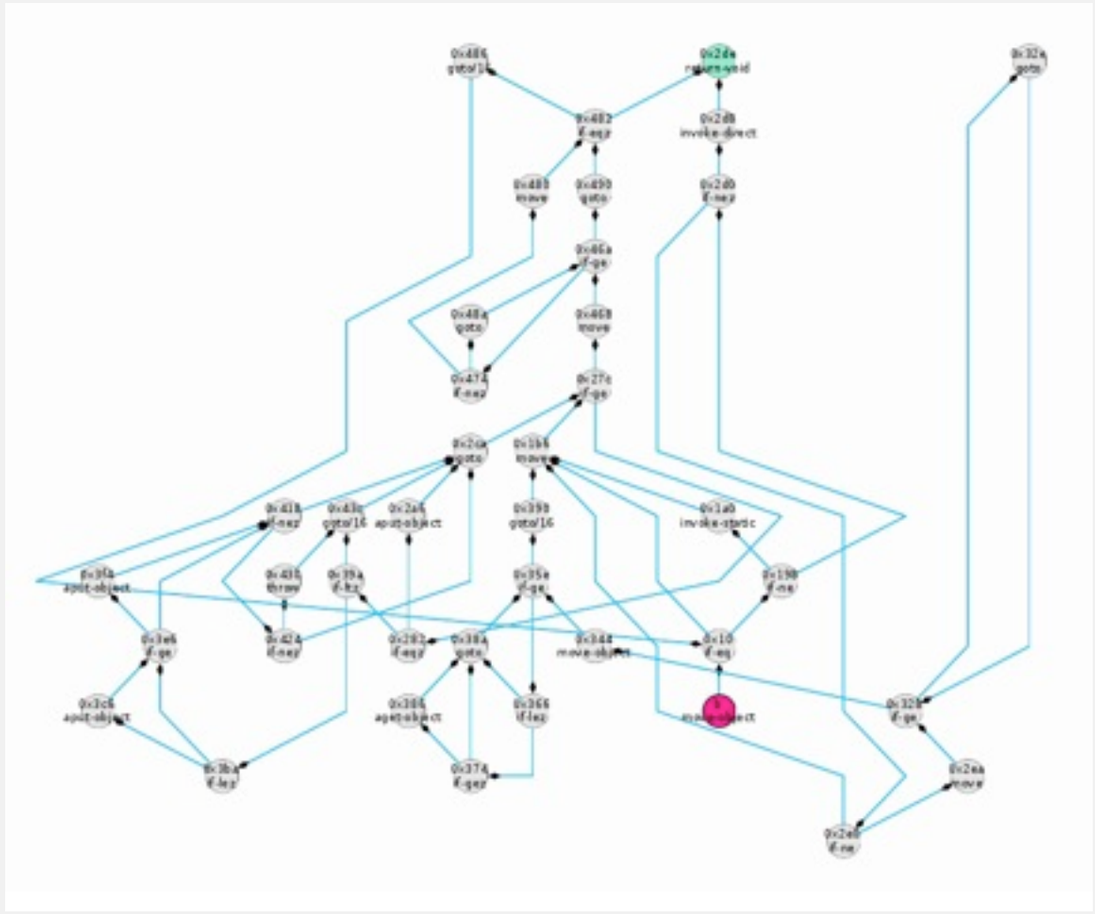


Malware Detection & Mitigation

Hunting season

Static Analysis

- Static taint analysis
  - Disassemble application
  - Construct control flow graph (CFG)
  - Static taint analysis
- e.g. android.Geimini function CFG



Malware Detection & Mitigation

Hunting season

Dynamic Analysis

- System call tracking
  - Install binary in **Sandbox**
  - Execute binary
  - Analyze system calls
- Dynamic Taint Analysis
  - Install binary in **VM**
  - Taint tracking
    - Variable-level
    - Method-level
    - Message-level
    - File-level

Malware Detection & Mitigation

Hunting season

Permission Analysis

- Permission policies
  - **Combination of perms. can be denied at install time**
- Statistical analysis
  - **Detection of overprivileged and potentially dangerous apps**

Type	Category	Permissions	Avg. Perms.
App.	Comics	9	0,98
	Communication	62	6,72
	Demo	16	1,46
	Entertainment	21	2,86
	Finance	21	1,84
	Health	15	1,50
	Libraries	40	1,36
	Lifestyle	45	3,42
	Multimedia	34	3,60
	News	22	3,62
	Productivity	52	3,98
	Reference	21	2,20
	Shopping	35	4,08
	Social	37	4,52
	Sports	17	2,20
	Themes	1	0,02
	Tools	49	3,88
	Travel	40	3,74
Games	Arcade	7	1,74
	Casino	15	2,30
	Casual	14	2,00
	Puzzle	10	1,30

Web Security Measures

Safe surfing

- User active protection
  - **HTTPS vs HTTP**
  - **Pop-ups blocking**
  - **Disabled cookies**
  - **Avoid links in e-mails**
  - **Avoid links in SMS**
- Automated protection
  - **Virtual mobile honeyclients**

Network Security

Preventing eavesdropping

Encryption

- WiFi Networks
  - Secure access points
  - WPA/WPA2
  - VPN connections
- SSL
  - Web-based apps
  - E.g. Whatsapp Messenger

Physical Security

This phone is mine

Authentication

- Password protection
  - **Automatic lock**
- Biometric authentication
  - **Face recognition unlock**
    - **Android 4.0 ICS**
  - **Not effective yet**
- Continuous authentication
  - **Behavioral biometrics**
  - **Classic biometrics**

Physical Security

This phone is mine

Other strategies

- Backups
  - Local storage
  - Clouds storage
- Remote geolocation & wipe
  - Internet based services
    - iCloud
    - Prey
    - Lookout Mobile
- Encryption
  - Disk encryption
  - Memory encryption

Some Conclusions

Security Threats & Measures in Smartphones

- **Smartphones** are already totally ubiquitous and have become the most personal and feature-filled gadget ever built.
- **Security measures** are introduced in mobile OS by design but there is still a large room for improvement.
- **Malware** is on its way to become as large in numbers as classic desktop malware and **Android** is its main target.
- **Proactive strategies** as threat modeling and active vulnerability analysis are more than ever the best tools to count on.

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN
Computer Security Group

Thanks

And keep safe