Looking Forward: Challenges in Mobile Security

John Mitchell
Stanford University
Outline

• Mobile platform security
• SessionJuggler
  – Using phone as authentication token
• SelectiveAuth
  – Protecting resources on the phone
• Stamp Project
  – Android malware project
• Perspective
  – Server side is also important for overall security
IPhone Flaw Lets Hackers Take Over, Security Firm Says

A team of computer security consultants say they have found a flaw in Apple's wildly popular iPhone that allows them to take control of the device.

Charles Miller, shown on his iPhone, said that after finding a hole in security, “you were in complete control.”

By JOHN SCHWARTZ
Published: July 23, 2007
SAN FRANCISCO — Just days after the T-Mobile G1 smartphone went on the market, a group of security researchers have found what they call a serious flaw in the Android software from Google that runs it.

One of the researchers, Charles A. Miller, notified Google of the flaw this week and said he was publicizing it now because he believed that cellphone users were not generally aware that increasingly sophisticated smartphones faced the same threats that plague Internet-connected personal computers.

Mr. Miller, a former National Security Agency computer security specialist, said the flaw could be exploited by an attacker who might trick a G1 user into visiting a booby-trapped Web site.

The G1 — the so-called Google phone — went on sale at T-Mobile stores on Wednesday.

Google executives acknowledged the issue but said that the security features of the phone would limit the extent of damage that could be done by an intruder, compared with today’s PCs and other cellphones.
Unlike modern personal computers and other advanced smartphones like the iPhone, the Google phone creates a series of software compartments that limit the access of an intruder to a single application.

“We wanted to sandbox every single application because you can’t trust any of them,” said Rich Cannings, a Google security engineer. He said that the company had already fixed an open-source version of the software and was working with its partners, T-Mobile and HTC, to offer fixes for its current customers.

Typically, today’s computer operating systems try to limit access by creating a partition between a single user’s control of the machine and complete access to programs and data, which is referred to as superuser, root or administrative access.

The risk in the Google design, according to Mr. Miller, who is a principal security analyst at Independent Security Evaluators in Baltimore, lies in the danger from within the Web browser partition in the phone. It would be possible, for example, for an intruder to install software that would capture keystrokes entered by the user when surfing to other Web sites. That would make it possible to steal identity information or passwords.
Google moves to delete 'RuFraud' scam Android apps

Google has removed 22 applications from its Android Market after they were discovered to contain fraudulent software.

Apps posing as popular third-party software such as Angry Birds tricked users into sending premium text messages.

Unlike some other app services, Android Market apps are not vetted prior to being added to the store.

Google has said it swiftly removes apps that violate its security policies.

Lookout, a mobile security company based in San Francisco, believes the fraud attempt originated from Russia.

After notifying Google of the 22 affected apps, Lookout said it then identified five more apps running the so-called "RuFraud" scam.

The scam would make a user believe they were about to download a game or program, but instead they were giving the phone "permission" to send a text message costing about £3.

Google has confirmed to the BBC that the additional apps have now been removed.
Two researchers at TippingPoint's Digital Vaccine Group have duped thousands of iPhone and Android smartphone users into joining a mobile botnet by spreading a seemingly innocuous weather application.
Trends

• Two attack vectors: web browser, installed apps.
  – Both are increasing in prevalence and sophistication

• Android malware is increasing:
  – Estimated 500,000 to 1 million users affected by Android malware in the first half of 2011
  – Approx 2.5x increase in second half of 2011
  – Three in ten will experience web-based threat /year
  – Approx 80 apps infected with malware in Jan 2011
  – Increased to over 400 apps cumulative in June 2011

• Attackers deploying increasingly sophisticated techniques to take control of the phone, personal data, and money

source: https://www.mylookout.com/mobile-threat-report
Web-based threats

- **Phishing**: mobile users more likely to fall to attack
- **Drive-by-downloads**: circumvent markets
- **Direct exploitation**
- **Mobile sites** may lag in https, other security measures

source: https://www.mylookout.com/mobile-threat-report
Malware-infected Apps

Leading categories: games, utilities, adult entertainment

source: https://www.mylookout.com/mobile-threat-report
Malicious App Creation

1. Developer creates a game called Monkey Jump.
2. Developer uploads game to Android Market.
3. Malicious Developer takes legitimate game and repackages it with malware.
4. Malicious Developer uploads game to 3rd party app store.
5. User downloads game with malware.
6. Malicious Developer can control the phone remotely and access users' private information.

Source: https://www.mylookout.com/mobile-threat-report
Research Sample:

Selective Authorization

Elie Bursztein, Jason Bau, Baptiste Goudin, John C. Mitchell
Stanford University
91% of users are concerned about their phone privacy
To prevent a **single** physical illegitimate access a pin code is enforced for **every** use.
Protect Applications Selectively?
User Study

- Browse Internet
- Install an App
- Make a Call
- Look at picture
- Listen to a Song
- Read Email
- Send a SMS
- Navigation
- Play a Game
- Take a Picture
- Access Social Profile

Categories:
- Family
- Friends
- Stranger
Android Application Permissions

• Each application has its own space (uid)
• Application capabilities model (gid)
• Try to enforce a “least privilege” idea
Alternative?

- Instead of asking at launch time do it at capability time
- **Intercept** and **block/warm** for specific phone permission access
Our “Selective Auth” Framework

- Password Schemes
- Configuration Interfaces
- Authentication Provider
- Database
- Alternative Pass Schemes
- Alternative Configuration Interfaces

Activity Manager Service
Cache

Android Application Framework
Dalvik Virtual Machine

Selective Authentication Framework
Third Party Applications
Android Framework

Specific Permission Needed
Configuration options

**Security Wizard**
Welcome to the phone security wizard. This wizard will help you set up two kinds of protection for your phone. You can tell this wizard to control:
- How users have access to different types of phone applications, and
- How phone applications have access to potentially sensitive phone behaviors, such as reporting your location.

**Email**
How do you want to protect:
- Read and write emails
Choose one unlock scheme

**Location**
When an application tries to access your location, do you want the phone to

Wizard Start  Usage Question  Permissions Question
Research Sample:

Secure Web Login From an Untrusted Terminal Using Session Hijacking

Elie Bursztein, Chinmay Soman, Dan Boneh, John C. Mitchell
Stanford University
SessionJuggler
SessionJuggler

1. Facebook Login Screen
2. SessionJuggler Start Screen
3. SessionJuggler Confirmation Dialog
4. Facebook Login Screen
5. Facebook Profile Page
6. Facebook Profile Page
### Anti-Hijacking Defenses

<table>
<thead>
<tr>
<th>Defense</th>
<th>% of Alexa100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using HTTPS</td>
<td>83%</td>
</tr>
<tr>
<td>Using Secure Cookies</td>
<td>52%</td>
</tr>
<tr>
<td>Separating Mobile and Desktop Sessions</td>
<td>6%</td>
</tr>
<tr>
<td>Binding Session to IP Address</td>
<td>8%</td>
</tr>
<tr>
<td>Checking Local Time</td>
<td>1%</td>
</tr>
<tr>
<td>Binding Session to User-Agent Header</td>
<td>0%</td>
</tr>
<tr>
<td>Binding Session to Local language</td>
<td>0%</td>
</tr>
<tr>
<td>Logout Over HTTPS</td>
<td>1%</td>
</tr>
</tbody>
</table>

Anti-hijacking defenses at the Alexa top 100 sites
# Sites with Improper Logout

<table>
<thead>
<tr>
<th>Website</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>health.google.com</td>
<td>View and edit record</td>
</tr>
<tr>
<td>healthvault.com</td>
<td>View and edit health record</td>
</tr>
<tr>
<td>Linkedin</td>
<td>Editing and saving profile</td>
</tr>
<tr>
<td>Yahoo</td>
<td>Accessing and sending emails</td>
</tr>
<tr>
<td>Hotmail/MSN</td>
<td>Accessing and sending emails</td>
</tr>
<tr>
<td>blogger.com</td>
<td>Posting a blog post</td>
</tr>
<tr>
<td>Ebay</td>
<td>Bidding on an auction</td>
</tr>
<tr>
<td>Flicker</td>
<td>Uploading photos</td>
</tr>
<tr>
<td>wordpress.com</td>
<td>Posting a blog post</td>
</tr>
<tr>
<td>IMDB</td>
<td>Editing and saving profile</td>
</tr>
<tr>
<td>ask.com</td>
<td>Editing and saving profile</td>
</tr>
<tr>
<td>cnn.com</td>
<td>Editing and saving profile</td>
</tr>
<tr>
<td>conduit.com</td>
<td>Editing and saving profile</td>
</tr>
<tr>
<td>megaupload.com</td>
<td>Uploading files</td>
</tr>
<tr>
<td>mediafire.com</td>
<td>Uploading files</td>
</tr>
<tr>
<td>4shared.com</td>
<td>Uploading files</td>
</tr>
<tr>
<td>cnet.com</td>
<td>Editing and saving profile</td>
</tr>
<tr>
<td>weather.com</td>
<td>Editing and saving profile</td>
</tr>
<tr>
<td>imageshack.com</td>
<td>Uploading photos</td>
</tr>
<tr>
<td>OpenMR</td>
<td>Accessing, changing medical records</td>
</tr>
</tbody>
</table>
Research Sample:

Android malware certification

Alex Aiken, John C. Mitchell, Mayur Naik, Isil Dillig, Thomas Dillig
Stanford, Georgia Tech, College of William and Mary
Idea #1

STAMP’s static analysis will infer the spec from the app’s code
An Example

- Consider a malicious barcode scanner that transmits phone's unique IMEI and IMSI numbers to a malicious server.

- STAMP infers the specs:
Idea #2

- Auditor inspects spec to decide if app is malware
False Alarms

• There will be false alarms
  – Code that is OK, but STAMP cannot prove is OK

• This is a huge problem in practice
  – One of the major issues in using static analysis
Idea #3

- Work with the auditor to refine specifications
Summary
STAMP: Static Analysis of Mobile Programs

Key problem:
Optimize auditor’s time

Solution:
Allow fine-grain interaction between STAMP and the auditor. At each step, the auditor can refine the specification, ask a question of the analysis, or have the analysis pose a question to the auditor.

 Novel features:
• plug-ins for modeling high-level API semantics
  • containers, database interface, etc.
• STAMP asks questions of the auditor to aid verification
  • cost model picks best question to ask
• auditor can ask detailed questions of STAMP
  • answered by dynamic symbolic analysis

Impact:
• path to scale to arbitrary apps through API extensibility
• unified model for the auditor
  • develop specification and verification simultaneously
• verification in hours, not days
Risks

- Are the semantics of Android well understood?
- Are source-sink flows sufficient as specifications?
- Can we make developing specifications productive?
- Spending time trying to understand the programming model
- No. But they are the first and perhaps most important class.
- Eat our dog food. Build a prototype, verify apps, and find out what gets in the way.
Perspective:

Why Mobile Application Security Risks Will Continue Growing for Enterprise and Consumer Mobile Applications
Why Enterprise Mobile Apps

Average number of applications per brand:

- 1-10: 70-80%
- 11-20: 20-30%
- 21-30: 0-10%
- 31-40: 40-50%
- 41-50: 10-15%
- 51-60: 15-20%
- 61-70: 15-20%
- 71-80: 10-15%
- 81-90: 10-15%
- 91-100: 0-5%

Over 40% of brands have deployed more than 30 applications.

Branded apps over time:

- Mar - 2010: 891
- Sept - 2010: 1091
- Mar - 2011: 1693
- Sept - 2011: 2343

Branded apps have increased by 263% in the period between March, 2010 to September, 2011.

Source: [x]cube Labs
Mobile Applications: Server Threats

- Concentration of risk on back-end services – that’s where the data is
- Enterprises not checking services used by mobile devices
- Threats are evolving even as mobile applications are downloaded
- Lots of hype around endpoint protection and source code scanning - these solutions cannot detect
  - Privacy escalation
  - Boundary authentication
Mobile App Vulnerability Trends

Vulnerability Category

- **Input Validation**
  - 50% of apps

- **Session (Authentication, Authorization)**
  - 60% of apps

- **Sensitive Information Disclosure**
  - 70% of apps

- **Infrastructure**
  - 60% of apps

**Percentage of Apps**

- **Typical Web application security stats**
  - 80% input validation injection attacks, 20% session authentication

- **Mobile applications from large enterprises tested via managed service**
  - 60% input validation attacks, 40% authentication related
  - 50 to 75% of apps had vulnerabilities that cannot be detected by endpoint analysis or source code analysis

source: Cenzic Application Security Intelligence: Mobile App Testing Results - February 2012
Conclusions

• Mobile platform and apps widely used
• Several threats
  – Browser threats, malicious apps, back-end services
• Research progress
  – Improving front-end device security
  – Web defenses useful against browser attack
  – New malware characteristics and platforms
  – Enterprise apps widely used
    • Need to protect both end point and apps that access backend systems
    • Significant threats to back-end services—bigger payoff
  – Auto update: app security important throughout lifecycle
• Likely continued job security for security professionals!