Mobile Platform Security
[continued]

Spring 2015

Franziska (Franzi) Roesner
franzi@cs.washington.edu

Thanks to Dan Boneh, Dieter Gollmann, Dan Halperin, Yoshi Kohno, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...
Roadmap

• Today: Finish mobile platform security
• Remaining topics:
  – 5/29: Usable security
  – 6/1: Anonymity
  – 6/3: Social engineering, physical security
  – 6/5: Side channels, emerging technologies

• Homework #3 – due 5pm on 5/29 (Friday)
• Lab #3 – due 5pm on 6/5 (next Friday)
Reminder: Challenges with Isolated Apps

So mobile platforms isolate applications for security, but...

1. **Permissions**: How can applications access sensitive resources?
2. **Communication**: How can applications communicate with each other?
Improving Permissions: AppFence

Today, ultimatums give app developers an unfair edge in obtaining permissions.

AppFence can enable new interfaces that give users control over the use of their info.

The App that User Wishes to Install

**This application has access to the following:**

- **Network Communication**: full Internet access
- **Your Personal Information**: read contact data
- **Phone Calls**: read phone state and identity

I’d rather not share all that information just to try this app, but it looks like I have no choice.

OK

The App that User Wishes to Install

**This application has access to the following:**

- **Network Communication**: block Internet access if data will be sent to any servers known advertisers any third parties
- **Your Personal Information**: allow access to all contact data
- **Device Information**: allow access to anonymized device ID report my phone # as 650 555 1212

I’ll start by giving out only the information I think this app actually needs.

OK
Improving Permissions: User-Driven Access Control

Insight:
A user’s natural UI actions within an application implicitly carry permission-granting semantics.

Let this application access my location now.
Access Control Gadgets (ACGs)

• Special UI elements that carry permission-granting semantics: When user clicks, grant access.

• ACGs are owned by system and embedded by apps: need to secure them!
  – No clickjacking, no programmatic clicking, etc.
(2) Inter-Process Communication

• Primary mechanism in Android: **Intents**
  – Sent between application components
    • e.g., with `startActivity(intent)`

  – **Explicit:** specify component name
    • e.g., `com.example.testApp MainActivity`

  – **Implicit:** specify action (e.g., `ACTION_VIEW`) and/or data (URI and MIME type)
    • Apps specify **Intent Filters** for their components.
Unauthorized Intent Receipt

- **Attack #1:** Eavesdropping / Broadcast Thefts
  - Implicit intents make intra-app messages public.

- **Attack #2:** Activity Hijacking
  - May not always work:

- **Attack #3:** Service Hijacking
  - Android picks one at random upon conflict!

[Chin et al.] 5/27/15
Intent Spoofing

• **Attack #1:** General intent spoofing
  – Receiving implicit intents makes component public.
  – Allows data injection.

• **Attack #2:** System intent spoofing
  – Can’t directly spoof, but victim apps often don’t check specific “action” in intent.
Aside: Incomplete Isolation

Embedded UIs and libraries always run with the host application’s permissions! *(No same-origin policy here...)*

[Shekhar et al.]
More on Android...
Security-Enhanced Linux (SELinux)

• Added in Android 4.3 to strengthen app isolation
• Mandatory access control (MAC): central system authority makes all access control decisions
  – In addition to standard Linux discretionary access control (DAC), in which objects have owners that make access control decisions
  – Result: Even processes running as root can be limited by explicit policy (example: only system server should modify system files)

Android Application Signing

• Apps are signed
  – Often with self-signed certificates
  – Signed application certificate defines which user ID is associated with which applications
  – Different apps run under different UIDs

• Shared UID feature
  – Shared Application Sandbox possible, where two or more apps signed with same developer key can declare a shared UID in their manifest
Shared UIDs

• App 1: Requests GPS / camera access
• App 2: Requests Network capabilities

• Generally:
  – First app can’t exfiltrate information
  – Second app can’t exfiltrate anything interesting

• With Shared UIDs (signed with same private key)
  – Permissions are a superset of permissions for each app
  – App 1 can now exfiltrate; App 2 can now access GPS / camera
File Permissions

• Files written by one application cannot be read/written by other applications
  – Not true for files stored on the SD card
  – SD card changes in Android 4.4: limited write ability

• Full file system encryption
  – Encryption key is protected with AES128 using key derived from user password (salted/hashed)
  – Root access not sufficient to break – need password
  – Enabled by default in Android 5.0
Memory Management

• Address Space Layout Randomization to randomize addresses on stack
• Hardware-based No eXecute (NX) to prevent code execution on stack/heap
• Stack guard derivative
• Some defenses against double free bugs (based on OpenBSD’s dmalloc() function)
• etc.

[See http://source.android.com/tech/security/index.html]
Android Fragmentation

• Many different variants of Android (unlike iOS)
  – Motorola, HTC, Samsung, ...

• Less secure ecosystem
  – Inconsistent or incorrect implementations
  – Slow to propagate kernel updates and new versions

  • “At Google I/O 2011, many of the largest OHA partners committed to providing updates to devices for 18 months after initial shipment.”
CSE 484 / CSE M 584: Computer Security and Privacy

[And now for something completely different...] CAPTCHA

Spring 2015

Franziska (Franzi) Roesner

franzi@cs.washington.edu

Thanks to Dan Boneh, Dieter Gollmann, Dan Halperin, Yoshi Kohno, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials...
Human Verification

• Problem:
  – Want to make it hard for spammers to automatically create many free email accounts
  – Want to make it difficult for computers to automatically crawl some data repository

• Need a method for servers to distinguish between human users and machine users

• Approach: CAPTCHA
  – Completely Automated Public Turing Test to Tell Computers and Humans Apart
CAPTCHAs

Figure 1: Examples of CAPTCHAs from various Internet properties.

Figure 2: Examples of CAPTCHAs downloaded directly from reCaptcha at different time periods.

Questions

Q1: What do you like/dislike about CAPTCHAs?
Q2: What properties of CAPTCHAs are valuable?
Q3: What properties of CAPTCHAs are “problematic”?
Q4: Should web sites use CAPTCHAs?

Q5: Suppose you are a spammer and want to create free accounts on Webmail Provider X, and Webmail Provider X uses CAPTCHAs during enrollment. How would you go about breaking those CAPTCHAs?
Re: \textsc{Captcha} – Understanding \textsc{Captcha}-Solving Services in an Economic Context

\textit{Marti Motoyama, Kirill Levchenko, Chris Kanich, Damon McCoy, Geoffrey M. Voelker and Stefan Savage}  
\textit{University of California, San Diego}  
{mmotoyam, klevchen, ckanich, dlmccoy, voelker, savage}@cs.ucsd.edu

\textbf{Abstract}

Reverse Turing tests, or \textsc{captcha}s, have become an ubiquitous defense used to protect open Web resources from being exploited at scale. An effective \textsc{captcha} resists existing mechanistic software solving, yet can be solved with high probability by a human being. In response, a robust solving ecosystem has emerged, re-selling both automated solving technology and real-time human labor to bypass these protections. Thus, \textsc{captcha}s can increasingly be understood and evaluated in purely economic terms; the market price of a solution \textit{vs} the monetizable value of the asset being protected. We examine the market-side of this question in depth, analyzing the behavior and dynamics of \textsc{captcha}-solving service providers, their price performance, and the underlying labor markets driving this economy.

Alphanumeric characters that are distorted in such a way that available computer vision algorithms have difficulty segmenting and recognizing the text. At the same time, humans, with some effort, have the ability to decipher the text and thus respond to the challenge correctly. Today, \textsc{captcha}s of various kinds are ubiquitously deployed for guarding account registration, comment posting, and so on.

This innovation has, in turn, attached value to the problem of solving \textsc{captcha}s and created an industrial market. Such commercial \textsc{captcha} solving comes in two varieties: automated solving and human labor. The first approach defines a technical arms race between those developing solving algorithms and those who develop ever more obfuscated \textsc{captcha} challenges in response. However, unlike similar arms races that revolve around spam or malware, we will argue that the underlying...
CAPTCHA-Solving Economies

Figure 3: CAPTCHA-solving market workflow: ① GYC Automator attempts to register a Gmail account and is challenged with a Google CAPTCHA. ② GYC uses the DeCaptcher plug-in to solve the CAPTCHA at $2/1,000. ③ DeCaptcher queues the CAPTCHA for a worker on the affiliated PixProfit back end. ④ PixProfit selects a worker and pays at $1/1,000. ⑤ Worker enters a solution to PixProfit, which ⑥ returns it to the plug-in. ⑦ GYC then enters the solution for the CAPTCHA to Gmail to register the account.

## CAPTCHA-Solving Economies

<table>
<thead>
<tr>
<th>Service</th>
<th>$/1K Bulk</th>
<th>Dates (2009–2010)</th>
<th>Requests</th>
<th>Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antigate (AG)</td>
<td>$1.00</td>
<td>Oct 06 – Feb 01 (118 days)</td>
<td>28,210</td>
<td>27,726 (98.28%)</td>
</tr>
<tr>
<td>BeatCaptchas (BC)</td>
<td>$6.00</td>
<td>Sep 21 – Feb 01 (133 days)</td>
<td>28,303</td>
<td>25,708 (90.83%)</td>
</tr>
<tr>
<td>BypassCaptcha (BY)</td>
<td>$6.50</td>
<td>Sep 23 – Feb 01 (131 days)</td>
<td>28,117</td>
<td>27,729 (98.62%)</td>
</tr>
<tr>
<td>CaptchaBot (CB)</td>
<td>$1.00</td>
<td>Oct 06 – Feb 01 (118 days)</td>
<td>28,187</td>
<td>22,677 (80.45%)</td>
</tr>
<tr>
<td>CaptchaBypass (CP)</td>
<td>$5.00</td>
<td>Sep 23 – Dec 23 (91 days)</td>
<td>17,739</td>
<td>15,869 (89.46%)</td>
</tr>
<tr>
<td>CaptchaGateway (CG)</td>
<td>$6.60</td>
<td>Oct 21 – Nov 03 (13 days)</td>
<td>1,803</td>
<td>1,715 (95.12%)</td>
</tr>
<tr>
<td>DeCaptcher (DC)</td>
<td>$2.00</td>
<td>Sep 21 – Feb 01 (133 days)</td>
<td>28,284</td>
<td>24,411 (86.31%)</td>
</tr>
<tr>
<td>ImageToText (IT)</td>
<td>$20.00</td>
<td>Oct 06 – Feb 01 (118 days)</td>
<td>14,321</td>
<td>13,246 (92.49%)</td>
</tr>
</tbody>
</table>

Table 1: Summary of the customer workload to the CAPTCHA-solving services.

Image from [http://static.usenix.org/event/sec10 tech/full papers/Motoyama.pdf](http://static.usenix.org/event/sec10 tech/full papers/Motoyama.pdf)
<table>
<thead>
<tr>
<th>Language</th>
<th>Example</th>
<th>AG</th>
<th>BC</th>
<th>BY</th>
<th>CB</th>
<th>DC</th>
<th>IT</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>one two three</td>
<td>51.1</td>
<td>37.6</td>
<td>4.76</td>
<td>40.6</td>
<td>39.0</td>
<td>62.0</td>
<td>39.2</td>
</tr>
<tr>
<td>Chinese (Simp.)</td>
<td>一 二 三</td>
<td>48.4</td>
<td>31.0</td>
<td>0.00</td>
<td>68.9</td>
<td>26.9</td>
<td>35.8</td>
<td>35.2</td>
</tr>
<tr>
<td>Chinese (Trad.)</td>
<td>一 二 三</td>
<td>52.9</td>
<td>24.4</td>
<td>0.00</td>
<td>63.8</td>
<td>30.2</td>
<td>33.0</td>
<td>34.1</td>
</tr>
<tr>
<td>Spanish</td>
<td>uno dos tres</td>
<td>1.81</td>
<td>13.8</td>
<td>0.00</td>
<td>2.90</td>
<td>7.78</td>
<td>56.8</td>
<td>13.9</td>
</tr>
<tr>
<td>Italian</td>
<td>uno due tre</td>
<td>3.65</td>
<td>8.45</td>
<td>0.00</td>
<td>4.65</td>
<td>5.44</td>
<td>57.1</td>
<td>13.2</td>
</tr>
<tr>
<td>Tagalog</td>
<td>isá dalawá tatió</td>
<td>0.00</td>
<td>5.79</td>
<td>0.00</td>
<td>0.00</td>
<td>7.84</td>
<td>57.2</td>
<td>11.8</td>
</tr>
<tr>
<td>Portuguese</td>
<td>um dois três</td>
<td>3.15</td>
<td>10.1</td>
<td>0.00</td>
<td>1.48</td>
<td>3.98</td>
<td>48.9</td>
<td>11.3</td>
</tr>
<tr>
<td>Russian</td>
<td>один два три</td>
<td>24.1</td>
<td>0.00</td>
<td>0.00</td>
<td>11.4</td>
<td>0.55</td>
<td>16.5</td>
<td>8.76</td>
</tr>
<tr>
<td>Tamil</td>
<td>இற சலவா தாளிங்</td>
<td>2.26</td>
<td>21.1</td>
<td>3.26</td>
<td>0.74</td>
<td>12.1</td>
<td>5.36</td>
<td>7.47</td>
</tr>
<tr>
<td>Dutch</td>
<td>een twee drie</td>
<td>4.09</td>
<td>1.36</td>
<td>0.00</td>
<td>1.22</td>
<td>31.1</td>
<td>6.30</td>
<td></td>
</tr>
<tr>
<td>Hindi</td>
<td>एक दो तीन</td>
<td>10.5</td>
<td>5.38</td>
<td>2.47</td>
<td>1.52</td>
<td>6.30</td>
<td>9.49</td>
<td>5.94</td>
</tr>
<tr>
<td>German</td>
<td>eins zwei drei</td>
<td>3.62</td>
<td>0.72</td>
<td>0.00</td>
<td>1.46</td>
<td>0.58</td>
<td>29.1</td>
<td>5.91</td>
</tr>
<tr>
<td>Malay</td>
<td>satu dua tiga</td>
<td>0.00</td>
<td>1.42</td>
<td>0.00</td>
<td>0.00</td>
<td>0.55</td>
<td>29.4</td>
<td>5.23</td>
</tr>
<tr>
<td>Vietnamese</td>
<td>mòt hai ba</td>
<td>0.46</td>
<td>2.07</td>
<td>0.00</td>
<td>0.00</td>
<td>1.74</td>
<td>18.1</td>
<td>3.72</td>
</tr>
<tr>
<td>Korean</td>
<td>일 이 삼</td>
<td>0.00</td>
<td>2.07</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>20.2</td>
<td>3.37</td>
</tr>
<tr>
<td>Greek</td>
<td>ἐνα δύο τρία</td>
<td>0.45</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>15.5</td>
<td>2.65</td>
</tr>
<tr>
<td>Arabic</td>
<td>ثلاثة اثنين واحد</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>15.3</td>
<td>2.56</td>
</tr>
<tr>
<td>Bengali</td>
<td>এক দুই তিন</td>
<td>0.45</td>
<td>0.00</td>
<td>9.89</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.72</td>
</tr>
<tr>
<td>Kannada</td>
<td>ಎರಡು ತೃತೀಯ</td>
<td>0.91</td>
<td>0.00</td>
<td>0.00</td>
<td>0.55</td>
<td>6.14</td>
<td>1.26</td>
<td></td>
</tr>
<tr>
<td>Klingon</td>
<td></td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.12</td>
<td>0.19</td>
<td></td>
</tr>
<tr>
<td>Farsi</td>
<td></td>
<td>0.45</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Table 2: Percentage of responses from the services with correct answers for the language CAPTCHAS.