Mobile Platform Security (finish)

Fall 2016

Ada (Adam) Lerner
lerner@cs.washington.edu

Thanks to Franzi Roesner, Dan Boneh, Dieter Gollmann, Dan Halperin, Yoshi Kohno, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...
Security Mindset: Customs

- Exchange on Reddit comment thread
- Started with an observation about the world:
  - “I tried to ship something to Venezuela, but it would have cost $80 shipping and $1420 in taxes and duty import fees!”
Security Mindset: Customs

• Problem: Extremely high customs fees.

• Solution?

Lie about the value of the item, or, better, claim it’s broken!
“That won’t make it past the customs inspection. They snatch it up in a heartbeat then throw the recipient in jail for fraud.”
“That can’t be right. Otherwise I could just send packages of people I don’t like in other countries with fake packing slips to have them arrested.”
Mobile Malware Attack Vectors

• Unique to phones:
  – Premium SMS messages
  – Identify location
  – Record phone calls
  – Log SMS

• Similar to desktop/PCs:
  – Connects to botmasters
  – Steal data
  – Phishing
  – Malvertising
Mobile Malware Examples

“ikee is never going to give you up”
(Android) Malware in the Wild

What does it do?

<table>
<thead>
<tr>
<th># Families</th>
<th>Root Exploit</th>
<th>Remote Control</th>
<th>Financial Charges</th>
<th>Information Stealing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Net</td>
<td>SMS</td>
<td>Phone Call</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>27</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>1204</td>
<td></td>
<td>1171</td>
<td>1</td>
<td>256</td>
</tr>
</tbody>
</table>

[Zhou et al.]
What’s Different about Mobile Platforms?

• Applications are isolated
  – Each runs in a separate execution context
  – No default access to file system, devices, etc.
  – Different than traditional OSes where multiple applications run with the same user permissions!

• **App Store**: approval process for applications
  – Market: Vendor controlled/Open
  – App signing: Vendor-issued/self-signed
  – User approval of permissions
Two Types of App We Want to Defend Against

• Malware

• Legit, but privacy invasive
(1) Permission Granting Problem

Smartphones (and other modern OSes) try to prevent such attacks by limiting applications’ access to:

- System Resources (clipboard, file system).
- Devices (camera, GPS, phone, ...).

How should operating system grant permissions to applications?
State of the Art

Prompts (time-of-use)
State of the Art

**Prompts** (time-of-use)

*Disruptive*, which leads to prompt-fatigue.

**Manifests** (install-time)
State of the Art

Prompts (time-of-use)

Disruptive, which leads to prompt-fatigue.

Manifests (install-time)

Out of context; not understood by users.

In practice, both are overly permissive: Once granted permissions, apps can misuse them.
Are Manifests Usable?

Do users pay attention to permissions?

24 observed installations

- 42% Didn’t look, but aware
- 42% Unaware of permissions
- 17% Looked at permissions

... but 88% of users looked at reviews.
Are Manifests Usable?

Do users understand the warnings?

<table>
<thead>
<tr>
<th>Permission</th>
<th>n</th>
<th>Correct Answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>READCALENDAR</td>
<td>101</td>
<td>46</td>
</tr>
<tr>
<td>CHANGENETWORKSTATE</td>
<td>66</td>
<td>26</td>
</tr>
<tr>
<td>READSMS1</td>
<td>77</td>
<td>24</td>
</tr>
<tr>
<td>CALLOPHONE</td>
<td>83</td>
<td>16</td>
</tr>
<tr>
<td>WAKELOCK</td>
<td>81</td>
<td>27</td>
</tr>
<tr>
<td>WRITEEXTERNALSTORAGE</td>
<td>92</td>
<td>14</td>
</tr>
<tr>
<td>READCONTACTS</td>
<td>86</td>
<td>11</td>
</tr>
<tr>
<td>INTERNET</td>
<td>109</td>
<td>12</td>
</tr>
<tr>
<td>READPHONESTATE</td>
<td>85</td>
<td>4</td>
</tr>
<tr>
<td>READSMS2</td>
<td>54</td>
<td>12</td>
</tr>
<tr>
<td>CAMERA</td>
<td>72</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 4: The number of people who correctly answered a question. Questions are grouped by the number of correct choices. \( n \) is the number of respondents. (Internet Survey, \( n = 302 \)
Do users act on permission information?

“Have you ever not installed an app because of permissions?”

- 72% responded Yes
- 20% responded No
- 8% responded Probably

25 interview responses
Over-Permissioning

- Android permissions are badly documented.
- Researchers have mapped APIs \(\rightarrow\) permissions.

www.android-permissions.org (Felt et al.), http://pscout.csl.toronto.edu (Au et al.)
Why is Over-Permissioning Bad?

• **Over-permissioning:** app has permission to access resources but never accesses them.

• If the app never uses the extra permissions, why is it bad that it has them?
Manifests rely on the user to make good choices at install time

- It’s not clear that users know how to make the right choice – or that there IS a right choice.

- I don’t want ANY app to access my camera at all times. I just want apps to access my camera when they need to for legitimate purposes!
Android 6.0: Prompts!

- **First-use prompts** for sensitive permission (like iOS).
- **Big change!** Now app developers need to check for permissions or catch exceptions.
Promps rely on the user to make good choices at use time

• It’s not clear that users know how to make the right choice at use time either.

• Still only checks on first use – the app can still use the resource for any reason it wants, at any time now or in the future.
Improving Permissions: AppFence

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

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.

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
  - 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.
Let this application access my location now.

**Insight:**
A user’s natural UI actions within an application implicitly carry permission-granting semantics.
Let this application access my location now.

Study shows: Many users already believe (52% of 186) – and/or desire (68%) – that resource access follows the user-driven access control model.

Improving Permissions: User-Driven Access Control

[Insight: A user’s natural UI actions within an application implicitly carry permission-granting semantics.]
New OS Primitive: Access Control Gadgets (ACGs)

**Approach:** Make resource-related UI elements first-class operating system objects (access control gadgets).

- To receive resource access, applications must embed a system-provided ACG.
- ACGs allow the OS to capture the user’s permission granting intent in application-agnostic way.
Misc Thoughts From Mobile Security
Permission Re-Delegation

- An application without a permission gains additional privileges through another application.
- Settings application is a *deputy*: has permissions, and accidentally exposes APIs that use those permissions.
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

[https://developer.android.com/about/dashboards/index.html]
Poor Usability Causes Problems
Importance in Security

• Why is usability important?
  – People are the critical element of any computer system
    • People are the real reason computers exist in the first place
  – Even if it is possible for a system to protect against an adversary, people may use the system in other, less secure ways
Today

• 3 case studies
  – Phishing
  – SSL warnings
  – Password managers

• Step back: root causes of usability problems, and how to address
Case Study #1: Phishing
A Typical Phishing Page

Weird URL
http instead of https
Safe to Type Your Password?
Safe to Type Your Password?

Bank of the West

Gives me you pa55w0rds!

User name:

Password:

Login

Done
Safe to Type Your Password?
Safe to Type Your Password?

“Picture-in-picture attacks”
Trained users are more likely to fall victim to this!
Experiments at Indiana University

- Reconstructed the social network by crawling sites like Facebook, MySpace, LinkedIn and Friendster
- Sent 921 Indiana University students a spoofed email that appeared to come from their friend
- Email redirected to a spoofed site inviting the user to enter his/her secure university credentials
  - Domain name clearly distinct from indiana.edu
- 72% of students entered their real credentials into the spoofed site
More Details

• Control group: 15 of 94 (16%) entered personal information
• Social group: 349 of 487 (72%) entered personal information

• 70% of responses within first 12 hours
• Adversary wins by gaining users’ trust

• Also: If a site looks “professional”, people likely to believe that it is legitimate
Phishing Warnings

Passive (IE)

Active (IE)

Active (Firefox)
Are Phishing Warnings Effective?

- CMU study of 60 users
- Asked to make eBay and Amazon purchases
- All were sent phishing messages in addition to the real purchase confirmations
- Goal: compare active and passive warnings
Active vs. Passive Warnings

- Active warnings significantly more effective
  - Passive (IE): 100% clicked, 90% phished
  - Active (IE): 95% clicked, 45% phished
  - Active (Firefox): 100% clicked, 0% phished
User Response to Warnings

• Some fail to notice warnings entirely
  – Passive warning takes a couple of seconds to appear; if user starts typing, his keystrokes dismiss the warning

• Some saw the warning, closed the window, went back to email, clicked links again, were presented with the same warnings… repeated 4-5 times
  – Conclusion: “website is not working”
  – Users never bothered to read the warnings, but were still prevented from visiting the phishing site

  – Active warnings work!
Why Do Users Ignore Warnings?

• Don’t trust the warning
  – “Since it gave me the option of still proceeding to the website, I figured it couldn’t be that bad”

• Ignore warning because it’s familiar (IE users)
  – “Oh, I always ignore those”
  – “Looked like warnings I see at work which I know to ignore”
  – “I thought that the warnings were some usual ones displayed by IE”
  – “My own PC constantly bombards me with similar messages”
The Lock Icon

- Goal: identify secure connection
  - SSL/TLS is being used between client and server to protect against active network attacker
- Lock icon should only be shown when the page is secure against network attacker
  - Semantics subtle and not widely understood by users
  - Whose certificate is it??
  - Problem in user interface design
Will You Notice?

Clever favicon inserted by network attacker
If you don’t recognize your personalized SiteKey, don’t enter your Passcode.
Do These Indicators Help?

• “The Emperor’s New Security Indicators”

Users don’t notice the absence of indicators!
Case Study #2: Browser SSL Warnings

• Design question: How to alert the user if a site’s SSL certificate is untrusted?
Firefox vs. Chrome Warning

33% vs. 70% clickthrough rate
Experimenting w/ Warning Design

<table>
<thead>
<tr>
<th>#</th>
<th>Condition</th>
<th>CTR</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Control (default Chrome warning)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Chrome warning with policeman</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Chrome warning with criminal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Chrome warning with traffic light</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Mock Firefox</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Mock Firefox, no image</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Mock Firefox with corporate styling</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Click-through rates and sample size for conditions.
Experimenting w/ Warning Design

<table>
<thead>
<tr>
<th>#</th>
<th>Condition</th>
<th>CTR</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Control (default Chrome warning)</td>
<td>67.9%</td>
<td>17,479</td>
</tr>
<tr>
<td>2</td>
<td>Chrome warning with policeman</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Chrome warning with criminal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Chrome warning with traffic light</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Mock Firefox</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Mock Firefox, no image</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Mock Firefox with corporate styling</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Click-through rates and sample size for conditions.

Figure 1. The default Chrome SSL warning (Condition 1).
Experimenting w/ Warning Design

<table>
<thead>
<tr>
<th>#</th>
<th>Condition</th>
<th>CTR</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Control (default Chrome warning)</td>
<td>67.9%</td>
<td>17,479</td>
</tr>
<tr>
<td>2</td>
<td>Chrome warning with policeman</td>
<td>68.9%</td>
<td>17,977</td>
</tr>
<tr>
<td>3</td>
<td>Chrome warning with criminal</td>
<td>66.5%</td>
<td>18,049</td>
</tr>
<tr>
<td>4</td>
<td>Chrome warning with traffic light</td>
<td>68.8%</td>
<td>18,084</td>
</tr>
<tr>
<td>5</td>
<td>Mock Firefox</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Mock Firefox, no image</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Mock Firefox with corporate styling</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Click-through rates and sample size for conditions.

Figure 1. The default Chrome SSL warning (Condition 1).

Figure 4. The three images used in Conditions 2-4.
Experimenting w/ Warning Design

<table>
<thead>
<tr>
<th>#</th>
<th>Condition</th>
<th>CTR</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Control (default Chrome warning)</td>
<td>67.9%</td>
<td>17,479</td>
</tr>
<tr>
<td>2</td>
<td>Chrome warning with policeman</td>
<td>68.9%</td>
<td>17,977</td>
</tr>
<tr>
<td>3</td>
<td>Chrome warning with criminal</td>
<td>66.5%</td>
<td>18,049</td>
</tr>
<tr>
<td>4</td>
<td>Chrome warning with traffic light</td>
<td>68.8%</td>
<td>18,084</td>
</tr>
<tr>
<td>5</td>
<td>Mock Firefox</td>
<td>56.1%</td>
<td>20,023</td>
</tr>
<tr>
<td>6</td>
<td>Mock Firefox, no image</td>
<td>55.9%</td>
<td>19,297</td>
</tr>
<tr>
<td>7</td>
<td>Mock Firefox with corporate styling</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Click-through rates and sample size for conditions.

Figure 2. The mock Firefox SSL warning (Condition 5).
# Experimenting w/ Warning Design

<table>
<thead>
<tr>
<th>#</th>
<th>Condition</th>
<th>CTR</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Control (default Chrome warning)</td>
<td>67.9%</td>
<td>17,479</td>
</tr>
<tr>
<td>2</td>
<td>Chrome warning with policeman</td>
<td>68.9%</td>
<td>17,977</td>
</tr>
<tr>
<td>3</td>
<td>Chrome warning with criminal</td>
<td>66.5%</td>
<td>18,049</td>
</tr>
<tr>
<td>4</td>
<td>Chrome warning with traffic light</td>
<td>68.8%</td>
<td>18,084</td>
</tr>
<tr>
<td>5</td>
<td>Mock Firefox</td>
<td>56.1%</td>
<td>20,023</td>
</tr>
<tr>
<td>6</td>
<td>Mock Firefox, no image</td>
<td>55.9%</td>
<td>19,297</td>
</tr>
<tr>
<td>7</td>
<td>Mock Firefox with corporate styling</td>
<td>55.8%</td>
<td>19,845</td>
</tr>
</tbody>
</table>

Table 1. Click-through rates and sample size for conditions.

![Firefox SSL warning](image.png)

Figure 3. The Firefox SSL warning with Google styling (Condition 7).
Opinionated Design Helps!

The site's security certificate is not trusted!

You attempted to reach 192.168.17.129, but the server presented a certificate issued by an entity that is not trusted by your computer's operating system. This may mean that the server has generated its own security credentials, which Chrome cannot rely on for identity information, or an attacker may be trying to intercept your communications.

You should not proceed, especially if you have never seen this warning before for this site.

Proceed anyway  Back to safety

Felt et al.

<table>
<thead>
<tr>
<th>Adherence</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>30.9%</td>
<td>4,551</td>
</tr>
</tbody>
</table>
Opinionated Design Helps!

The site's security certificate is not trusted!

You attempted to reach 192.168.17.129, but the server presented a certificate that is not trusted by your computer's operating system. This may mean that the server cannot be verified by your computer, or that the certificate has expired. You should not proceed, especially if you have never seen this warning before.

Adherence

<table>
<thead>
<tr>
<th>Percentage</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>30.9%</td>
<td>4,551</td>
</tr>
<tr>
<td>32.1%</td>
<td>4,075</td>
</tr>
<tr>
<td>58.3%</td>
<td>4,644</td>
</tr>
</tbody>
</table>
Challenge: Meaningful Warnings

- a248.e.akamai.net
- Client missing root certificate
- Captive portal
- Expired certificate
- Client clock wrong
- Anti-virus software
- School or employer
- Gov't content filter
- ISP adding advertisements
- Certificate mis-issuance
- Malware
- State attacks

11/30/16

CSE 484 / CSE M 584 - Spring 2016
Password Managers

• Separate application and/or extension in your browser.

• Remembers and automatically enters passwords on your behalf.

• Seems possibly easier than remembering all your passwords. Is it more secure?
• Q. What are the root causes of usability issues in computer security?
Issue #1: Complexities, Lack of Intuition

Real World

We can see, understand, relate to.

Electronic World

Too complex, hidden, no intuition.

SSL/TLS  XSS  RSA
Buffer overflows  Phishing  Spyware
Issue #1: Complexities, Lack of Intuition

• Mismatch between perception of technology and what really happens
  – Public keys?
  – Signatures?
  – Encryption?
  – Message integrity?
  – Chosen-plaintext attacks?
  – Chosen-ciphertext attacks?
  – Password management?
  – ...

11/30/16
Issue #2: Who’s in Charge?

Real World

Complex, hidden, but doctors manage

Electronic World

SSL/TLS  XSS
RSA
Buffer overflows
Phishing  Spyware

Users want to feel like they’re in control.

Where analogy breaks down: Adversaries in the electronic world can be intelligent, sneaky, and malicious.

Complex, hidden, and users manage
Issue #2: Who’s in Charge?

• Systems developers should help protect users
  – Usable authentication systems
  – Usable privacy settings (e.g., on social media)
  – User-driven access control

• Software applications help users manage their applications
  – Anti-virus software
  – Anti-web tracking browser add-ons
  – PwdHash, Keychain for password management
  – Some say: Can we trust software for these tasks?
Issue #3: Hard to Gauge Risks

“It won’t happen to me!” (Sometimes a reasonable assumption, sometimes not.)

Schneier on Security
A weblog covering security and security technology.

May 02, 2005

Users Disabling Security

It's an old story: users disable a security measure because it's annoying, allowing an attacker to bypass the measure.

A [redacted] accused in a deadly courthouse rampage was able to enter the chambers of the judge slain in the attack and hold the occupants hostage because the door was unlocked and a buzzer entry system was not activated, a sheriff's report says.

Security doesn't work unless the users want it to work. This is true on the personal and national scale, with or without technology.


"It won't happen to me!"

Sometimes a reasonable assumption, sometimes not.)
Issue #4: No Accountability

• Issue #3 is amplified when users are not held accountable for their actions
  – E.g., from employers, service providers, etc.
  – (Not all parties will perceive risks the same way)

• Also, recall that a user’s poor security choices may affect other people
  – E.g., compromise account of user with weak password, then exploit a local (rather than remote) vulnerability to get root access
Issue #5: Annoying, Awkward, or Difficult

• Difficult
  – Remembering 50 different, “random” passwords

• Awkward
  – Lock computer screen every time leave the room

• Annoying
  – Browser warnings, virus alerts, forgotten passwords, firewalls

• Consequence:
  – Changing user’s knowledge may not affect their behavior
Issue #6: Social Issues

• Public opinion, self-image
  – Only “nerds” or the “super paranoid” follow security guidelines

• Unfriendly
  – Locking computers suggests distrust of co-workers

• Annoying
  – Sending encrypted emails that say, “what would you like for lunch?”
Issues with Usability

1. Lack of intuition
   - See a safe, understand threats. Not true for computers.

2. Who’s in charge?
   - Doctors keep your medical records safe, you manage your passwords.

3. Hard to gauge risks
   - “It would never happen to me!”

4. No accountability
   - Asset-holder is not the only one you can lose assets.

5. Awkward, annoying, or difficult

6. Social issues
Question

• Q. What approaches can we take to mitigate usability issues in computer security?
Response #1: Education and Training

• Education:
  – Teaching technical concepts, risks

• Training
  – Change behavior through:
    • Drill
    • Monitoring
    • Feedback
    • Reinforcement
    • Punishment

• May be part of the solution – but not the solution
Response #2: Security Should Be Invisible

• Security should happen
  – Naturally
  – By Default
  – Without user input or understanding

• Recognize and stop bad actions

• Starting to see some invisibility
  – SSL/TLS
  – VPNs
  – Automatic Security Updates
  – User-driven access control
Response #2: Security Should Be Invisible

- “Easy” at extremes, or for simple examples
  - Don’t give everyone access to everything

- But hard to generalize

- Leads to things not working for reasons user doesn’t understand

- Users will then try to get the system to work, possibly further reducing security
  - E.g., “dangerous successes” for password managers
Response #3: “3 Word UI”: “Are You Sure?”

• Security should be invisible
  – Except when the user tries something dangerous
  – In which case a warning is given

• But how do users evaluate the warning? Two realistic cases:
  – Always heed warning. But see problems / commonality with Response #2 (“security should be invisible”)
  – Always ignore the warning. If so, then how can it be effective?
Response #4: Focus on Users, Use Metaphors

- Clear, understandable metaphors:
  - Physical analogs; e.g., red-green lights

- User-centered design: Start with user model

- Unified security model across applications
  - User doesn’t need to learn many models, one for each application

- Meaningful, intuitive user input
  - Don’t assume things on user’s behalf
  - Figure out how to ask so that user can answer intelligently
Response #5: Least Resistance

• “Match the most comfortable way to do tasks with the least granting of authority”
  – Ka-Ping Yee, *Security and Usability*

• Should be “easy” to comply with security policy

• “Users value and want security and privacy, but they regard them only as secondary to completing the primary tasks”
  – Karat et al, *Security and Usability*