A View To A Kill

WebView Exploitation

Matthias Neugschwandtner
Martina Lindorfer
Christian Platzer

International Secure Systems Lab
Vienna University of Technology
Web - Views

- Consumption of web content shifts to mobile devices
- Typically not through browser but standalone app
WebView Library

• Browser library for mobile devices
• Available on all popular Smartphone OS
• Allows quick development of web-based apps
  – HTML, JavaScript, CSS
  – Also targeted at inexperienced developers
  – Third party frameworks (Apache Cordova) require no native code at all
  – Updates just require change of web content
WebView vs. Browser

• Provides access to device functionality via JavaScript
  – Hardware buttons
  – Persistent storage
  – Contacts
  – SMS
  – Location
  – …

• Allows development of more streamlined and capable apps

• No containment of web content (sandbox)
Threat Scenario
Server Compromise

Victim

② GET foo.html

Attacker

③ <\>
④ Data Leak

Webserver

① Malicious Script
Threat Scenario
Traffic Compromise

1. GET foo.html
2. Webserver
3. Data Leak
4. Attacker

Usenix LEET 2013
6th Usenix Workshop on Large-Scale Exploits and Emergent Threats
## Threat Scenario Comparison

<table>
<thead>
<tr>
<th></th>
<th>Server Compromise</th>
<th>Traffic Compromise</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attack leverage</strong></td>
<td>Large (all installations of a single app are affected)</td>
<td>Smaller (depends on number and location of rogue AP)</td>
</tr>
<tr>
<td><strong>Encryption</strong></td>
<td>Server takes care of encryption</td>
<td>Only possible with apps that use plain text or don’t handle encryption properly</td>
</tr>
<tr>
<td><strong>Feasibility</strong></td>
<td>Server dependent</td>
<td>Traffic dependent</td>
</tr>
</tbody>
</table>
Case Study
“Take Weather”

- Social weather-photo sharing app
- Available for iOS and Android
  - 10,000-50,000 installs on Android
- Uses plain HTTP
- Based on Cordova
  - Cross-platform access to contacts, call log, location (GPS)
  - Android: full access to Java
WebView on Android

- Provides JavaScript-Java bridge
  - Expose complete Java objects via
    `WebView.setJavascriptEnabled()`
    `WebView.addJavascriptInterface(<object>, <js_object_name>)`
  - Use reflection to create objects & invoke methods
- Requires signed certificate for HTTPS
Case Study
“Jiepang”

• Chinese “Foursquare” – location based social app
• 100,000-500,000 installs
• Permissions to
  – access external storage
  – install packages
• Uses HTTPS, but
  – overwrites default SSL error handler
  – accepts any certificate
Large Scale Evaluation

WebView Prevalence

- 287,512 Android apps submitted to Andrubis
- July 2012 to March 2013
- WebView usage:

<table>
<thead>
<tr>
<th>WebView related method call</th>
<th>Samples</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>loadURL</td>
<td>166,751</td>
<td>55%</td>
</tr>
<tr>
<td>setJavaScriptEnabled</td>
<td>158,042</td>
<td>58%</td>
</tr>
<tr>
<td>addJavaScriptInterface</td>
<td>87,079</td>
<td>30%</td>
</tr>
</tbody>
</table>
### Traffic Type

<table>
<thead>
<tr>
<th>Traffic Type</th>
<th>Samples</th>
<th>Percentage of JS-enabled samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unencrypted HTML or JavaScript</td>
<td>23,048</td>
<td>27%</td>
</tr>
<tr>
<td>Lax SSL handling</td>
<td>6,208</td>
<td>7%</td>
</tr>
</tbody>
</table>

### Permissions

<table>
<thead>
<tr>
<th>Permissions</th>
<th>Samples</th>
<th>Percentage of vulnerable samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMS (receive, read, write, send)</td>
<td>3,124</td>
<td>11%</td>
</tr>
<tr>
<td>Installation (write, install)</td>
<td>16,726</td>
<td>60%</td>
</tr>
<tr>
<td>Privacy (contacts, location)</td>
<td>21,197</td>
<td>76%</td>
</tr>
</tbody>
</table>
Mitigation & Conclusion

• Use of HTTPS and correct certificate handling
  – Signed certificates
  – Certificate pinning
  – WebView targeted at inexperienced developers
• Android 4.2 introduced @JavascriptInterface annotation
  – Will take time until 4.2 is run by a majority of the devices
  – New annotation only prevents reflection attacks
  – Intended functionality is still available