All Your Mobile Applications Are Belong To Us

Itzik Kotler, Chief Technology Officer

Hello Motto, Hello Hacker

- Mobile phones are no longer only for making calls and sending SMS's
- There are vulnerabilities in Mobile phone operating systems and Applications, just like in every other software/hardware
- No, “Closed Garden” stores do not guarantee securer applications, just limited availability
- No, Not jailbreaking/rooting your Smartphone will not stop hackers from attacking you

Say App, Not Website

- No longer a window in a browser
- Manage it's own SSL stack
- Writes and reads files
- Calls OS/SDK API

Know Your Mobile Application

- Client / Mobile Handset
  - Application
  - OS
- Client-Server Communication
  - Application
- Server / Mobile Application Server
  - Application
  - Server Software
  - OS

Mobile App Penetration Testing

- Testing the application on each platform for:
  - Client-side attacks
    - Server Responses
  - Server-side attacks
    - SQL Injections, Fuzzing, Buffer overflows
  - Network attacks
    - Man-in-the-middle, SSL attacks
  - Local storage attacks
    - Evil CA, Passwords stored in plain text and etc.

Where is the Penetration Tester?

- Remotely, simulating an attacker who is on the same network and executes a man-in-the-middle attack
- Locally, simulating an attacker who is on the same mobile handset and reads/writes files

Server-Side Mobile App Attack

- Web Application Security
  - Lack of Input Validation
  - Fuzzing
- Network Security / SSL
  - Weak Ciphers
  - Legacy Versions (SSLv2)
- Server Application Security
  - Buffer overflow
  - Heap overflow

This is (almost always) not 'www.'

- For many reasons, mobile apps are usually not talking with 'www.' but rather with a different server (sometimes known as 'mobile.')
- These servers are configured differently and somewhat under the assumption that the attacker won't be able to find/access them
- The result is web servers which are poorly configured (e.g. default files, loose SSL settings and etc.)

Client-Side Mobile App Attacks

- Network attacks
  - Spoof server using DNS
  - Spoof server using HTTP redirection (30X, Meta tag and etc.)
- SSL attacks
  - Spoof server using Self-signed Certificate
- Server replies
  - Change application data
  - Change application behavior
  - Write data to mobile handset storage
  - Read data from mobile handset storage
- Local attacks
  - Spoof server using Evil Certificate Authority
  - Read data from to application-related files
  - Write data to application-related files

Luke, I Am Your Server!

Basic Server Spoofing Methods

- Remote:
  - Man-in-the-middle
  - DNS spoofing
  - Application level redirections
- Local:
  - Pharming (i.e. hosts file)

Mobile App GUI helps Hackers

- Full blown applications (as oppose to web browsers) do not include server and connection details in their GUI
- Attacker can freely switch from HTTPS to HTTP
- Attacker can freely use application level redirections such as HTTP 301, 302, Refresh Meta Tag and etc. to switch between servers

Luke, I Am Your SSL Server!

SSL Self-Signed Certificate Attack

- Attacker setup SSL server with a self-signed certificate and redirect the application
- Browser doesn't know better then to display a pop-up dialog (trusts the user to make the call)
- Application should know better then to connect to a server with a self-signed server
- Shows the difference between developing a web application and mobile application

SSL Evil Certificate Authority Attack

- Attacker installs a Certificate Authority on mobile handset (Social Engineering, Malware) and uses pharming
- Application connects to attacker SSL server without any warnings or errors
- Application should verify certificate serial number and issue to match known good
- Hard to detect, easy to pull.

To Wi-Fi, or not to Wi-Fi?

- Public Wi-Fi Hotspots such as in Coffee shops and Airports are security hazards
- No encryption? Free for all Man-in-the-middle!
- Applications don't do risk management and thus allows the same functionally and parameters over any type of Internet access (e.g. 3G, Encrypted Wi-Fi and Public Wi-Fi)

Simple Banking Mobile App Risk Management Algorithm

- IF NetworkType EQUAL 3G/2G/GPRS
  - SET LimitTransfer = UNLIMITED
- IF NetworkType EQUAL Wi-Fi Protected Access
  - SET LimitTransfer = 100k
- IF NetworkType EQUAL Public Wi-Fi
  - SET LimitTransfer = 50k

LimitTrasnfer reflects the risk of fraud/attack

UI Attacks

- Not necessarily an application attack but rather a design vulnerability in OS/browser level
- Small screen and displayed data gets truncated
- What happens if it's the URL?
- A combination of DNS spoofing and URL truncating can result in a sophisticated phishing attack

Israel's AppStore Survey July 2011

- Top 10-20 Apps from the following categories:
  - Books
  - Education
  - Entrainment
  - Finance
  - Games
  - Health and Fitness
  - Medical
  - Music
  - Navigation
  - News
  - Photography
  - Productivity
  - Reference
  - Social Networking
  - Sports
  - Travel
  - Utilities
  - Weather

Total of 280 applications were tested

Survey App Testing Methodology

- Mostly Client-side tests (Non-intrusive)
- Focus on Network attacks (Not local)
- iPhone 3GS and iPod as testing devices
- Wi-Fi (w/ Man-in-the-middle) for Internet Access

Survey – SSL Figures

- 42.5% (119) of the apps were Client-Server
- 36% of the Client-Server apps were using SSL
- 18% of the SSL Client-Server were not enforcing SSL (same URL, HTTP, works)
- 4% of the SSL Client-Server were accepting self-signed certificates

Survey – Client-Side Attacks

- 42.5% (119) of the apps were Client-Server
- 26% of the apps had exploitable response
- 75% of the exploitable response allowed changing one of the following parameters:
  - Price
  - Balance
  - Time/dates
  - Queries results

Executive Summary

- We have found:
  - App that sends CC details over HTTP
  - App that sends login credentials over HTTP
  - Apps that accept self-signed certificates
  - Apps that do not enforce SSL (HTTPS)
  - Apps that server response changes prices
  - Apps that have servers who speaks SSLv2

Security for Mobile Applications

- Mobile application needs to undergo security tests just like any other software
- Mobile application servers should not assume that they can not be accessed directly
- Mobile applications GUI should inform the user of switching between HTTP and HTTPS, as well as display the connection/server details
- Mobile applications need to do risk management and differentiate between different connectivity methods and their risk

Questions?

Thank you!

Security-Art: info@security-art.com
My Twitter: @itzikkotler
My Email: ik@ikotler.org