

Introducing the Smartphone Pentesting Framework

Georgia Weidman
Bulb Security LLC

Approved for Public Release, Distribution Unlimited

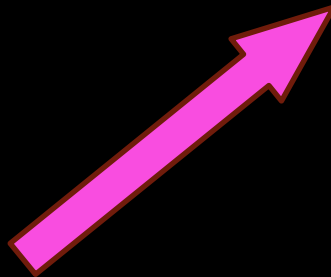
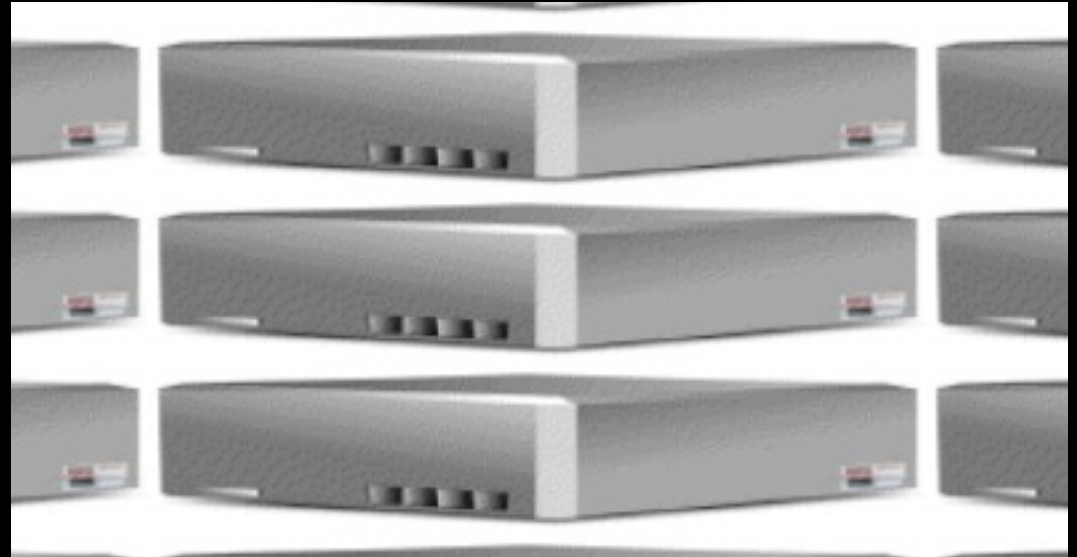
Disclaimer

“The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.” This is in accordance with DoDI 5230.29, January 8, 2009.

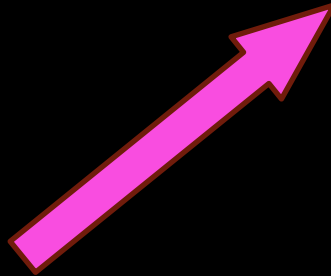
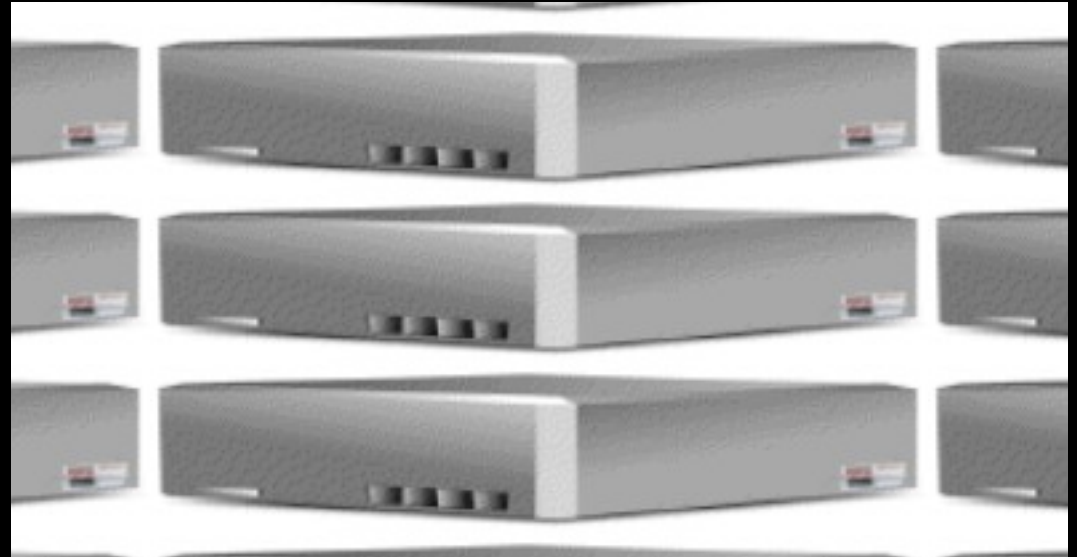
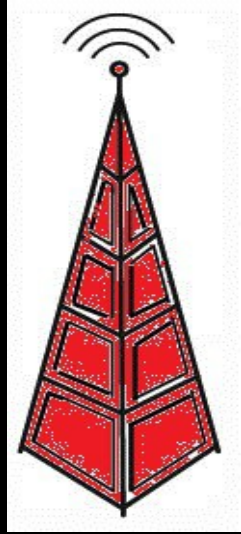
<3 to DARPA

- DARPA Cyber Fast Track program funded this project
- Without them I'd still be a junior pentester at some company
- Now I'm CEO!
- <3 <3 <3 <3 <3

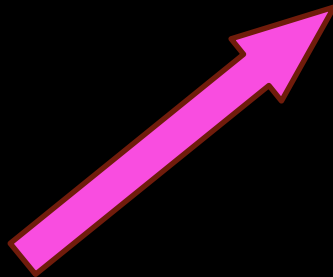
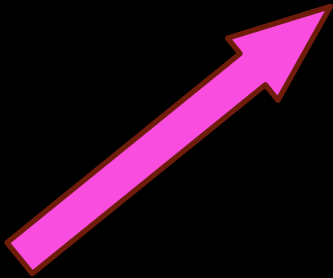
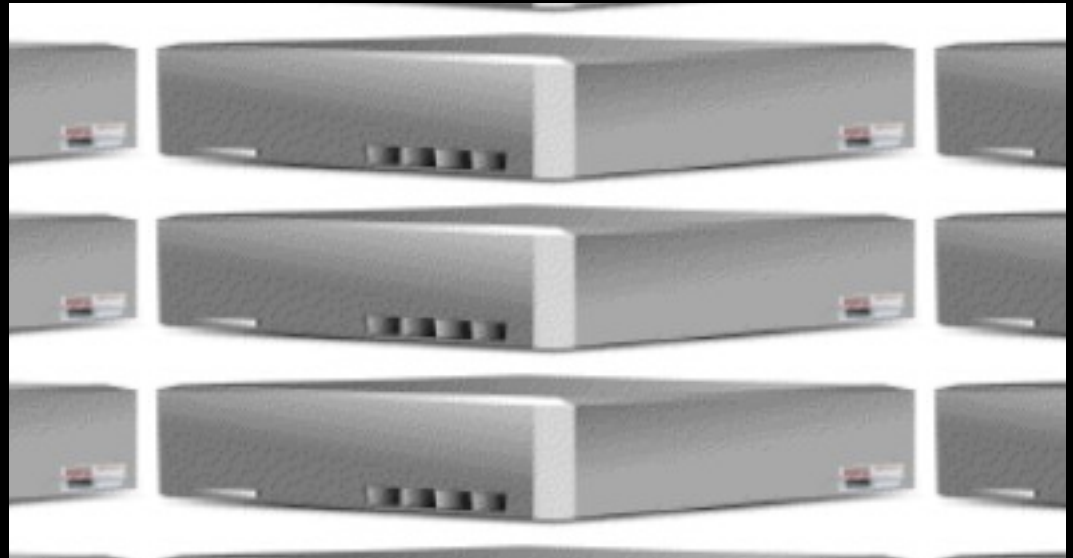
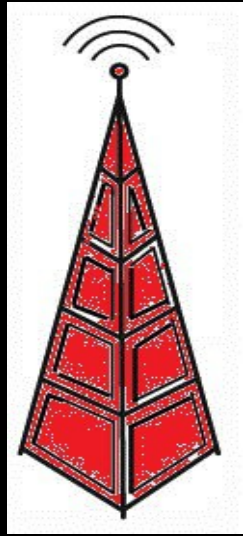
The Problem: Smartphones in the Workplace



The Problem: Smartphones in the Workplace



The Problem: Smartphones in the Workplace



Smartphones in the workplace

- Access your data
- Store company emails
- Connect to VPNs
- Generate 1 time passwords

Threats against smartphones: Apps

- Malicious apps steal your data, remotely control your phone, etc.
- Happens on all platforms. Some easier than others.
- If your employees have a malicious angry birds add-on what is it doing with your data?

Threats against smartphones: software bugs

- Browsers have bugs
- Apps have bugs
- Kernels have bugs
- Malicious apps, webpages, etc. can exploit these and gain access to data

Threats against smartphones: social engineering

- Users can be tricked into opening malicious links
- Downloading malicious apps

Threats against smartphones: jailbreaking

- Smartphones can be jailbroken
- Giving a program expressed permission to exploit your phone
- Once it is exploited, what else does the jailbreaking program do?

The Question

A client wants to know if the environment is secure

I as a pentester am charged with finding out

There are smartphones in the environment

How to I assess the threat of these smartphones?

What's out there now?

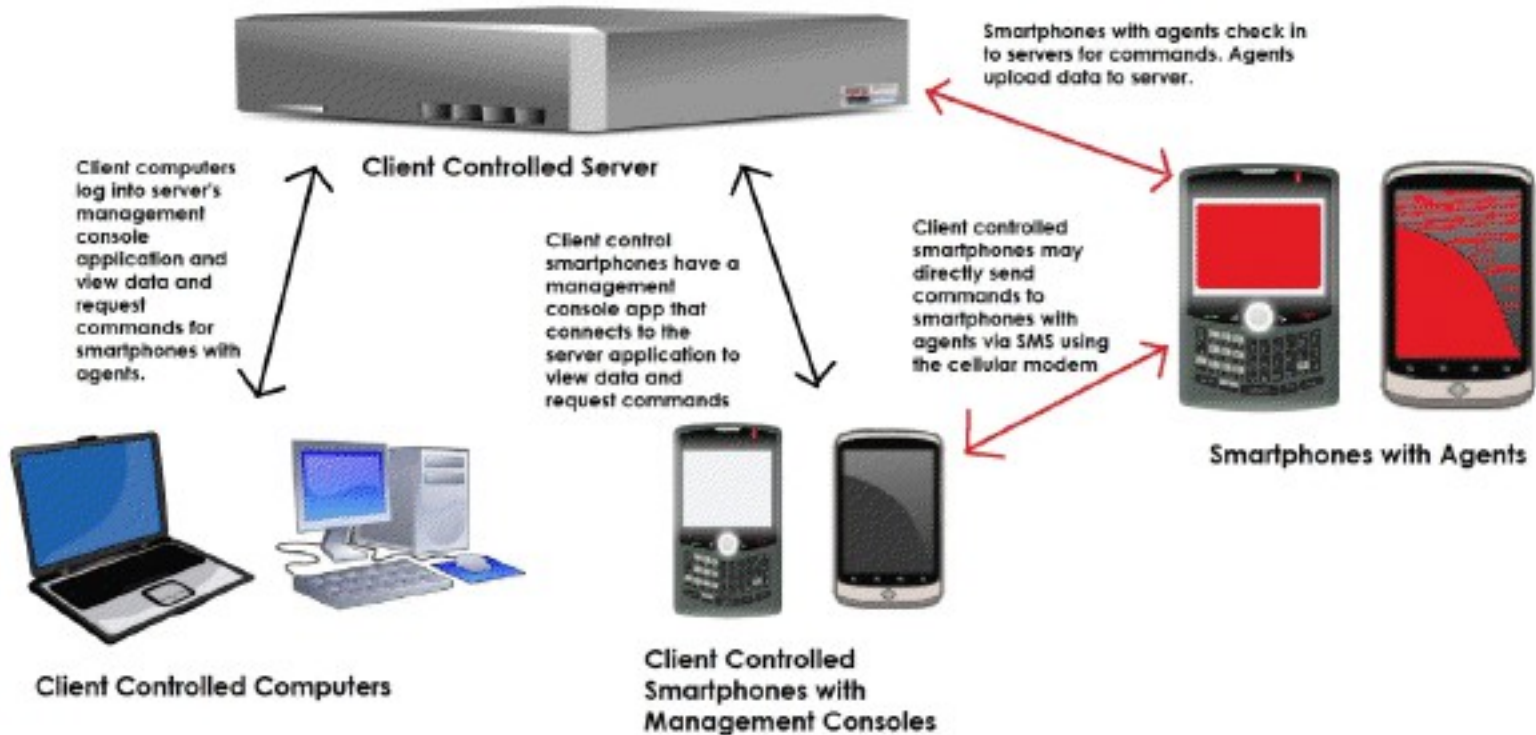
Pentesting from Smartphones: zAnti

Smartphone tool live cds: MobiSec (another DARPA project)

Pentesting smartphone apps: Mercury

Pentesting smartphone devices: ??

Structure of the framework



Framework console

```
root@bt: ~/Desktop
File Edit View Terminal Help

1.) 15555215554

Select an agent to interact with or 0 to return to the previous menu
spf>1

Commands:
1.) Send SMS
2.) Take Picture
3.) Get Contacts
4.) Get SMS Database
5.) Privilege Escalation

Select a command to perform or 0 to return to the previous menu
spf>
```

Framework GUI




← → <http://localhost/georgia/menu.pl> SmartPhone PenTest Frame... X

 **Bulb Security**

Georgia Weidman, CTO
871-435-4821
gweidman@bulbsecurity.com
<http://www.bulbsecurity.com>

- **Attach Framework to Deployed Agent**
- **Send Command**
- **View Information Gathered**
- **Attach Framework to Mobile Modem**
- **Run a Remote Attack**
- **Run a Social Engineering or Client Side Attack**
- **Clear/Create Database**

Framework GUI

 **Bulb Security**

Georgia Weidman, CEO
571 425 4881
georgia@bulbsecurity.com
http://www.bulbsecurity.com

Attach Framework to Deployed Agent


Agent Phone Number:

Agent URL Path:

Platform: Android

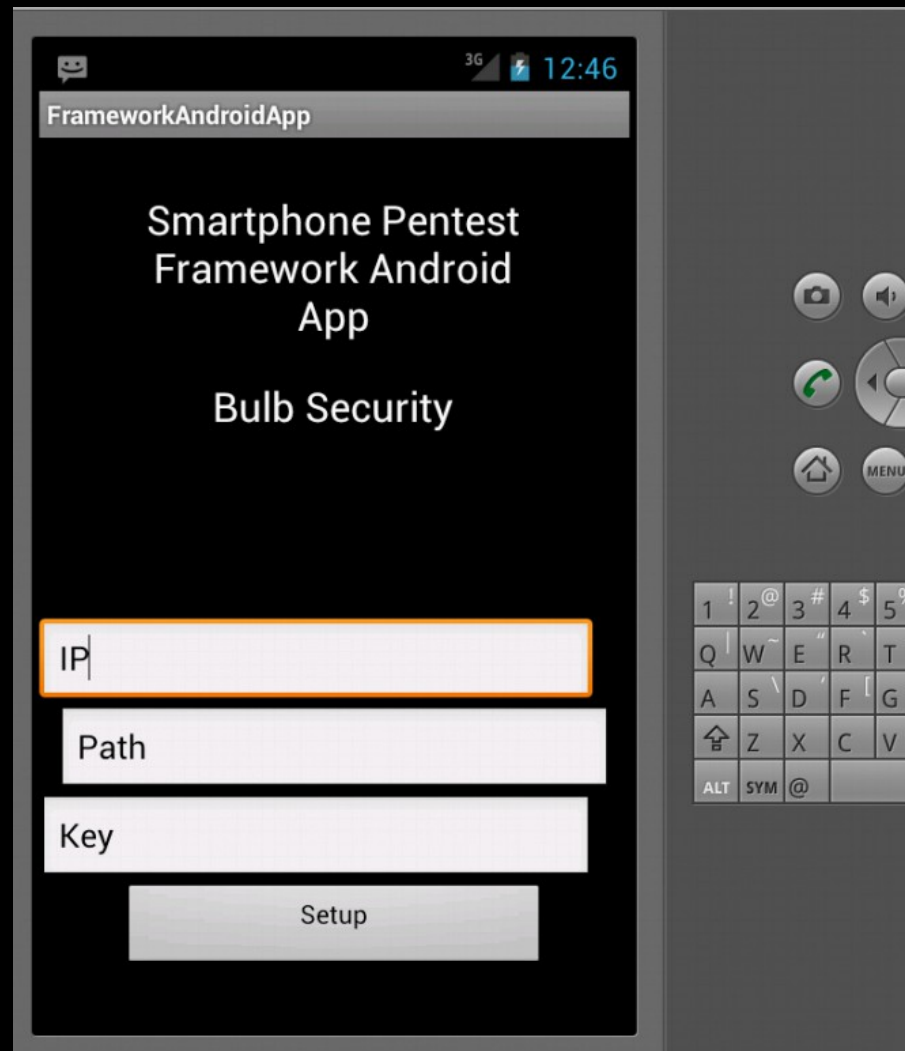
Control Phone Number:

Agent Control Key:

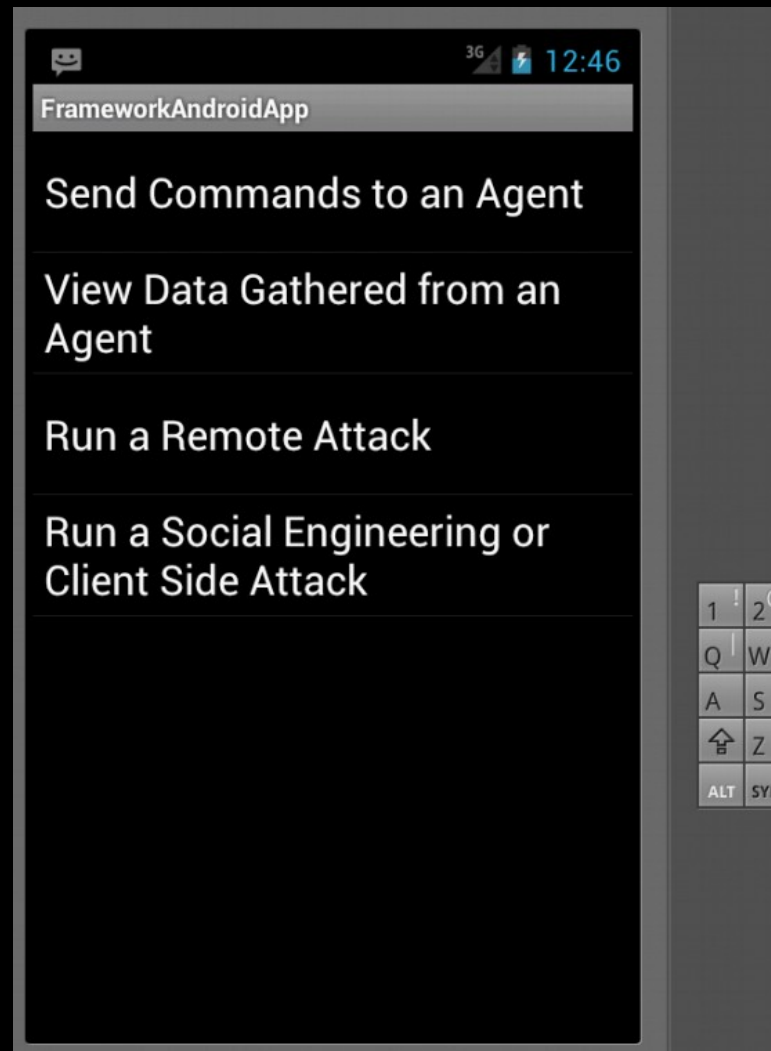
 **Attach**

- Send Command
- I ● View Information Gathered
- Attach Framework to Mobile Modem
- Run a Remote Attack
- Run a Social Engineering or Client Side Attack

Framework Smartphone App



Framework Smartphone App



Framework Smartphone App

The screenshot shows a mobile application interface with a dark background. At the top, the status bar displays a signal strength icon, '3G', a battery icon, and the time '12:46'. Below the status bar is a header area with a smiley face icon on the left, the text 'FrameworkAndroidApp', and the subtitle 'Launch A Social Engineering or Client Side Attack'. The main content area consists of several input fields and a button, all with a light gray background and rounded corners. From top to bottom: a dropdown menu showing 'Browser Exploits', another dropdown menu showing 'Android', a third dropdown menu showing 'CVE-2010-1759 Webkit', a text input field containing 'Path' (highlighted with an orange border), a text input field containing 'Filename', a text input field containing 'Number', and a 'Submit' button. To the right of the app interface, a portion of a virtual keyboard is visible, showing keys for '1 ! 2 @', 'Q W', 'A S', a home key, 'Z', 'ALT', and 'SYM'.

What you can test for

Remote vulnerabilities

Client side vulnerabilities

Social engineering

Local vulnerabilities

Remote Vulnerability Example

Jailbroken iPhones all have the same default SSH password

How many jailbroken iPhones have the default SSH password (anyone can log in as root)?

Client Side Vulnerability Example

Smartphone browsers, etc. are subject to vulnerabilities

If your users surf to a malicious page their browsers may be exploited

Are the smartphone browsers in your organization vulnerable to browser exploits?

Social Engineering Vulnerability Example

SMS is the new email for spam/phishing attacks

“Open this website” “Download this app”

Will your users click on links in text messages?

Will they download apps from 3rd parties?

Local Vulnerability Example

Smartphones have kernel vulnerabilities

Used my jailbreaks and malicious apps

Are the smartphones in your organization subject to local privilege escalation vulnerabilities?

Post exploitation

Command shell

App based agent

 Payloads: information gathering

 local privilege escalation

 remote control

Demos!

- Using the console
- Using the GUI
- Using the app
- Using an agent
- Using a shell
- Remote test
- Client side test
- Local test

Future of the Project

- More modules in each category
- More post exploitation options
- Continued integration with Metasploit and other tools
- Community driven features
- More reporting capabilities

Contact

Georgia Weidman

Bulb Security, LLC

georgia @ bulbsecurity.com

georgiaweidman.com bulbsecurity.com

@georgiaweidman