Smartphone Pentest Framework v0.1

User Guide

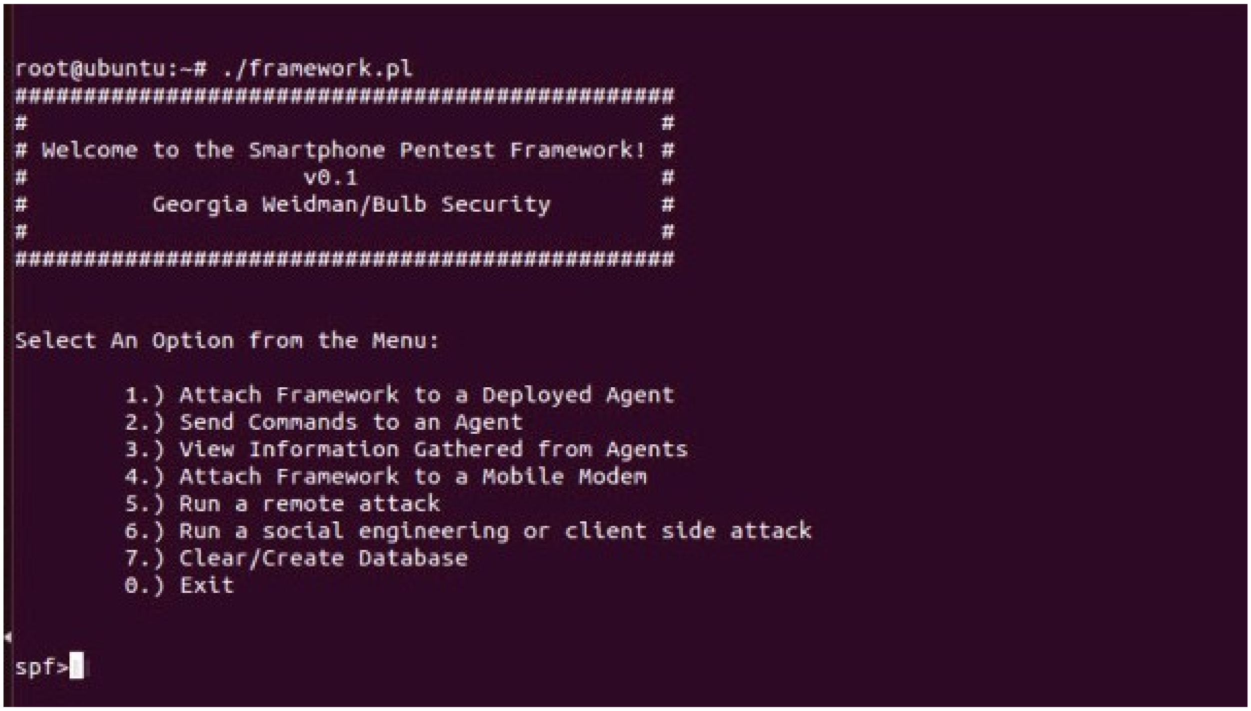
Introduction:

The Smartphone Pentest Framework (SPF) is an open source tool designed to allow users' to assess the security posture of the smartphones deployed in an environment. The tool allows for assessment of remote vulnerabilities, client side attacks, social engineering attacks, post exploitation and local privilege escalation. This is an initial release, with a subset of features from each section. SPF is the product of DARPA Cyber Fast Track grant.

SPF is made up of several parts that may be mixed and matched to meet users' needs. SPF v0.1 includes the following:

- SPF Console
- SPF Web based GUI
- SPF Android App
- SPF Android Agent

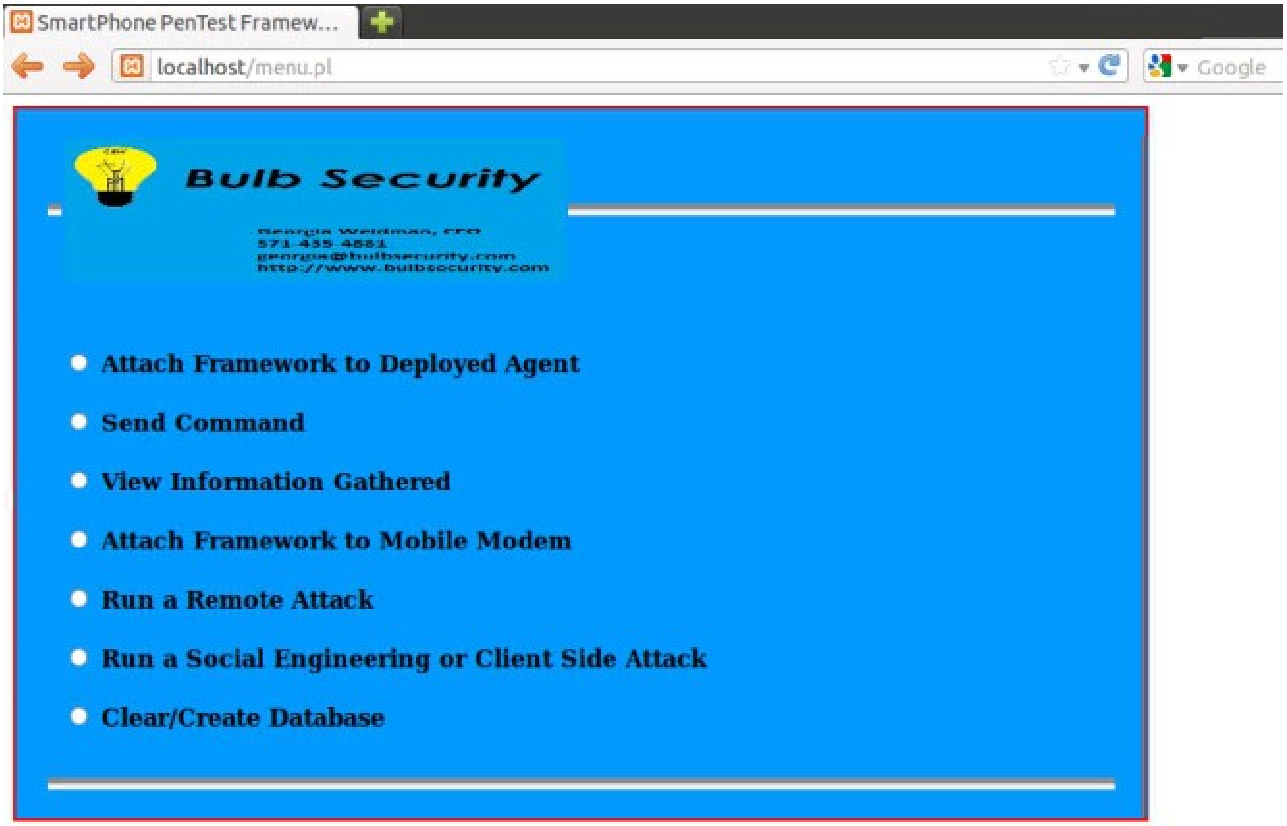
SPF Console:



```
root@ubuntu:~# ./framework.pl
###################################################
#                                                 #
# Welcome to the Smartphone Pentest Framework! #
#                      v0.1                       #
#         Georgia Weidman/Bulb Security           #
#                                                 #
###################################################


Select An Option from the Menu:

        1.) Attach Framework to a Deployed Agent
        2.) Send Commands to an Agent
        3.) View Information Gathered from Agents
        4.) Attach Framework to a Mobile Modem
        5.) Run a remote attack
        6.) Run a social engineering or client side attack
        7.) Clear/Create Database
        0.) Exit


spf>
```

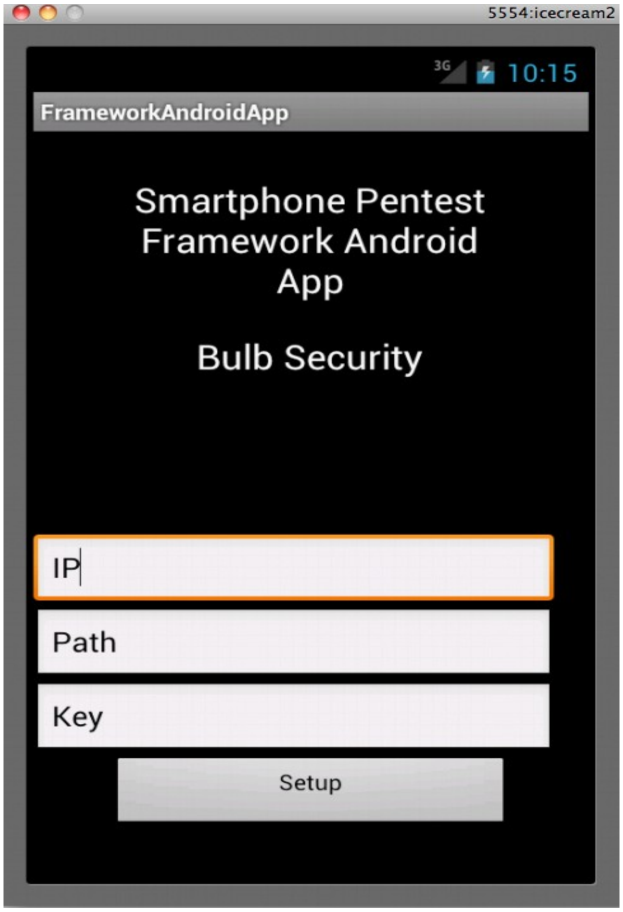
The console is a text based Perl program that allows SPF users to perform all the server functionality of SPF.

SPF GUI:


SmartPhone PenTest Framew...
localhost/menu.pl
Google
Bulb Security
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

The GUI is a web based front end for SPF that allows users' to perform all the server functionality. It is a set of Perl based webpages.

SPF Android App:



Smartphone Pentest Framework Android App

Bulb Security

GEW Bulb Security

The SPF Android App allows users to use the mobile modem of the Android smartphone with SPF to send SMS messages, gather information, etc. Users can also perform server functionality directly from Android smartphones using this app.

SPF Android Agent:

The SPF Android Agent is one of SPF's post exploitation options. It is transparent to the user and allows SPF users to perform post exploitation tasks such as privilege escalation, information gathering, and remote control on Android phones with the agent installed. Agents for iPhone and Blackberry platforms are currently in development.

Prerequisites:

Webserver

Mysql database

Perl

Perl packages:

    Bundle::Expect

    DBD::MYSQL

Webserver Perl support (required by the SPF GUI)

Android smartphone (required by the SPF Android app)

Android development environment (if modification of agents and app to meet users' custom needs is desired)

Installation:

SPF is currently supported on the Linux operating system. Support for Windows and Mac is a near term planned feature.

Installing Prerequisites:

Perl is probably already present on your system. If not type **apt-get install perl** (or platform equivalent) at the command line. Two dependencies for the SPF console (Expect and Mysql) are not met in the base install. Expect installs easily Linux systems using Perl's CPAN tool.

```
perl -MCPAN -e 'install Bundle::Expect'
```

The syntax to install Mysql is:

```
perl -MCPAN -e 'install DBD::mysql'
```

however this is prone to errors and may fail.

`apt-get install libdbd-mysql-perl` (or your platforms equivalent)

is a better solution.

A Mysql database server is required by SPF. The database server need not be on the same physical machine if so desired. SPF will use a database called “framework.” Details about the Mysql server will be read by SPF from its config file, so keep those handy.

A webserver is required by SPF. The webserver does need to be on the same physically machine as the SPF console and/GUI at this time. The webserver will require Perl support to use the SPF GUI. Webserver Perl support is not required to exclusively use the console.

If you are having trouble installing the prerequisites, XAMPP includes both the necessary Mysql server and webserver (including Perl support).

Installing SPF Console:

The SPF console is packaged as frameworkconsole.tar. Untar it with the command **tar xvf frameworkconsole.tar**. Then change the permissions on all the files with **chmod 777 *** in the framework console directory. Make changes to the config file (discussed later in this section), and you are ready to go.

Installing SPF GUI:

The SPF GUI is packaged as frameworkgui.tar. Untar it in the webserver root with the command **tar xvf frameworkgui.tar**. Then change the permissions on all the files with **chmod 777 *** in the framework console directory. Make changes to the config file (discussed later in this section), and you are ready to go.

Installing SPF Android App:

The full source code of the SPF Android App as well as an APK is available in FrameworkAndroidApp.7z. Unzip it with 7zip. To make changes to the source code, open up the project in your Android development environment. To have the Framework Android App catch SMS based responses from a deployed agent (discussed later in this document), open the SMSReceiver.java file and change the values of **key** and **controlnumber** to the control key and phone number of the deployed agent respectively. Compile the APK and install it on the desired smartphone. All other variables will be set during setup.

Config File:

SPF GUI and SPF console include an identical text based config file. This file tells SPF important information such as the location and login information of the Mysql server, location of the webserver, etc. At the current time the config file reads as follows:

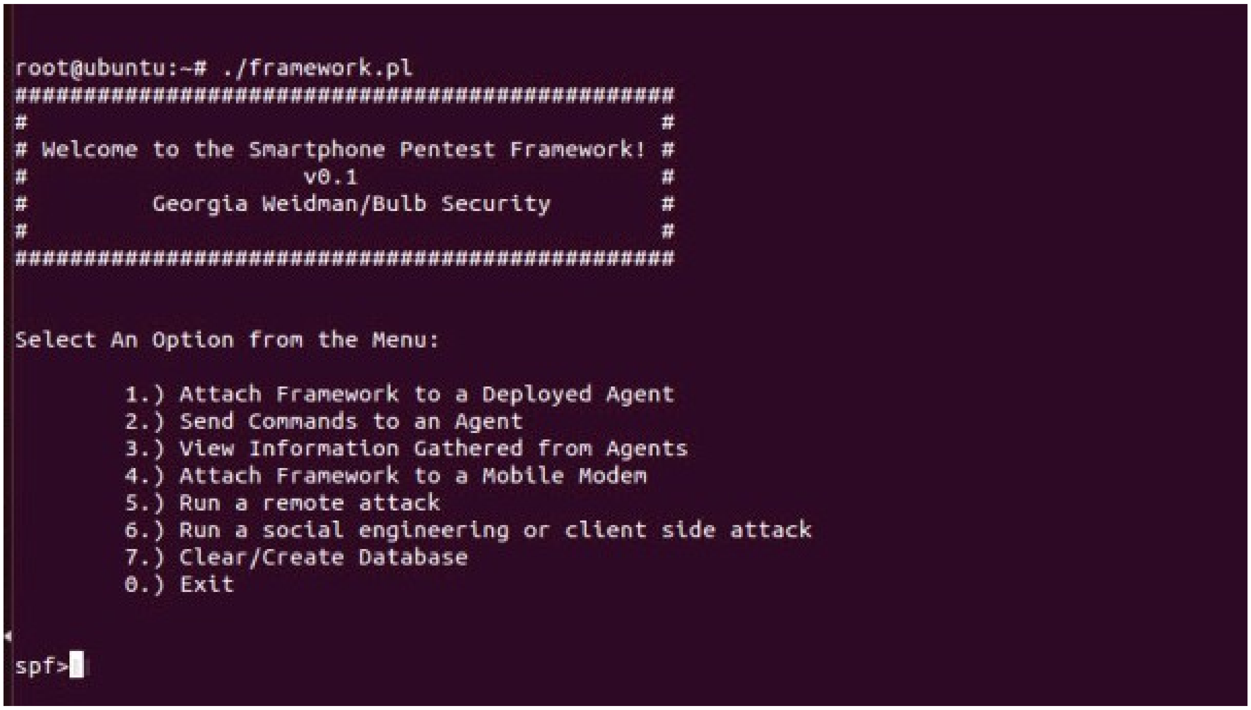
```
#SMARTPHONE PENTEST FRAMEWORK CONFIG FILE
#ROOT DIRECTORY FOR THE WEBSERVER THAT WILL HOST OUR FILES
WEBSERVER = /opt/lampp/htdocs
#IPADDRESS TO LISTEN ON
IPADDRESS = 192.168.0.108
#IP ADDRESS OF SQLSERVER 127.0.0.1 IF LOCALHOST
MYSQLSERVER = 192.168.0.107
#USERNAME OF THE MYSQL USER TO USE
MYSQLUSER = root
#PASSWORD OF THE MYSQL USER TO USE
MYSQLPASS = password
#PORT MYSQL IS RUNNING ON (3306 IS DEFAULT)
MYSQLPORT = 3306
#LOCATION OF ANDROID APK FOR AGENT DROP
ANDROIDAGENT = /root/AndroidAgent.apk
```

```
#LOCATION OF IPHONE DEB FOR AGENT DROP
IPHONEAGENT = /root/iphone.deb
```

Change any values necessary to match your own environment. It SPF console or GUI fails a bad configuration option is the most likely culprit.

Using the SPF Console:

To start the SPF console type **./framework.pl**.



```
root@ubuntu:~# ./framework.pl
################################################
#                                              #
# Welcome to the Smartphone Pentest Framework! #
#                     v0.1                     #
#         Georgia Weidman/Bulb Security        #
#                                              #
################################################


Select An Option from the Menu:

        1.) Attach Framework to a Deployed Agent
        2.) Send Commands to an Agent
        3.) View Information Gathered from Agents
        4.) Attach Framework to a Mobile Modem
        5.) Run a remote attack
        6.) Run a social engineering or client side attack
        7.) Clear/Create Database
        0.) Exit


spf>
```

Creating or Clearing the Database:

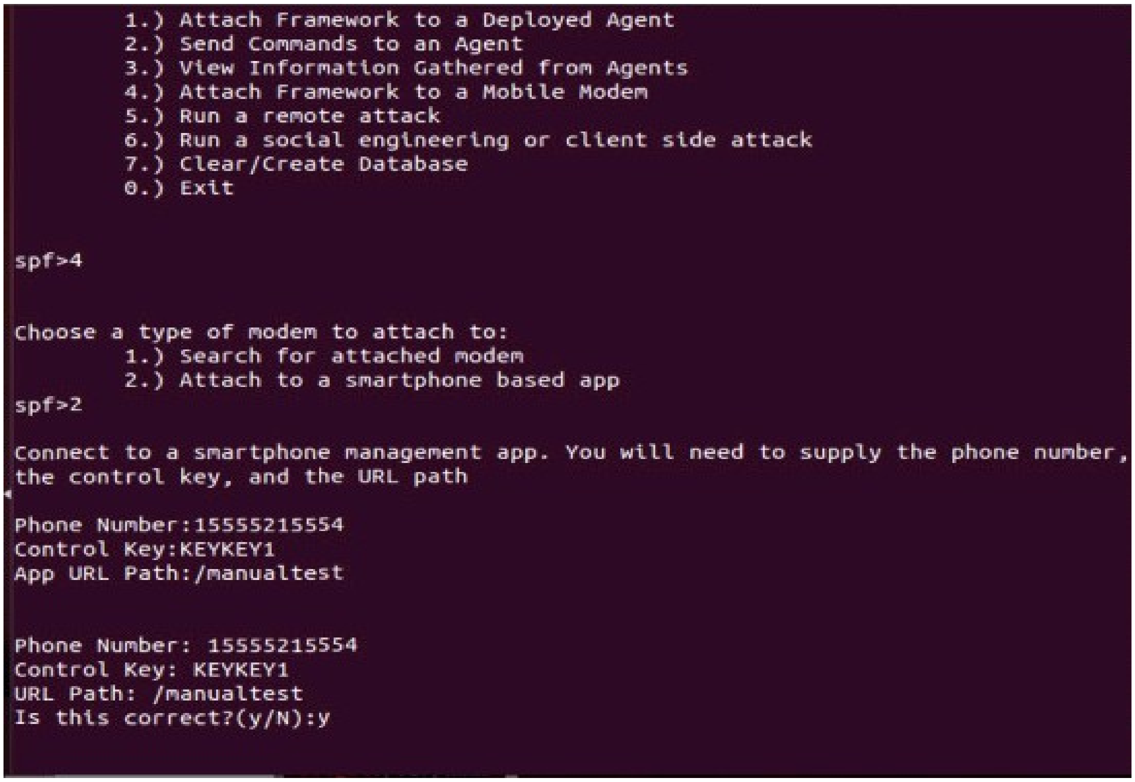
If it is your first time using SPF, the first thing to do is set up the database. If things just get too complicated and you want to start over, using this option is also a good idea. However, be forewarned that this will destroy all your data and detach all modems, agents, apps, etc.

To clear/create the database choose option **7**. You will be warned that this will destroy all your data. Type **y** to agree to this. All tables in the framework database will be dropped if they exist and recreated with no data.

Attaching to an Android App

Currently the Android app is the only way to connect a mobile modem to SPF. Support for USB mobile modems and Google Voice is currently in development. In order to use the mobile network to send attacks, communicate with deployed agents, etc. you will need to deploy the SPF Android app on an Android smartphone (any Android phone with a mobile modem and Android version 1.6 and above works, so burn phones are acceptable). The SPF console will use the modem on the Android device to communicate over the mobile network.

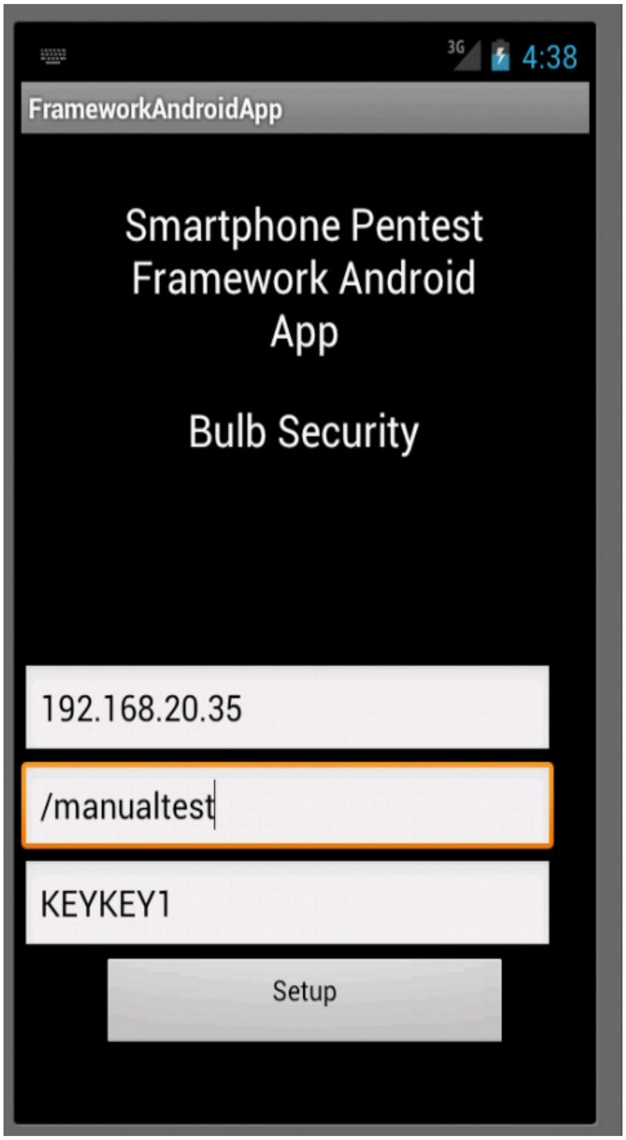
To attach to an SPF Android app choose option **4**. Choose option **2** to connect to a smartphone based app (option 1 for an attached mobile modem is still in development). You will be asked to supply the phone number of the smartphone where the app is installed, a path on the webserver to host control files, and the 7 digit key that will be used to control the app. Fill in this data and then choose **y** if the data is correct.



```
        1.) Attach Framework to a Deployed Agent
        2.) Send Commands to an Agent
        3.) View Information Gathered from Agents
        4.) Attach Framework to a Mobile Modem
        5.) Run a remote attack
        6.) Run a social engineering or client side attack
        7.) Clear/Create Database
        0.) Exit


spf>4


Choose a type of modem to attach to:
        1.) Search for attached modem
        2.) Attach to a smartphone based app
spf>2

Connect to a smartphone management app. You will need to supply the phone number,
the control key, and the URL path

Phone Number:15555215554
Control Key:KEYKEY1
App URL Path:/manualtest


Phone Number: 15555215554
Control Key: KEYKEY1
URL Path: /manualtest
Is this correct?(y/N):y
```

The SPF console will appear to hang. It is waiting for a SPF app to connect. Open the SPF Android app. You will be asked to fill in the IP address of the webserver, the control key, and the path on the webserver to the control files. Make sure that the key and path are the same as you entered in the SPF console.

Click Setup and both the app and the console will return to the menu screens.



Smartphone Pentest Framework Android App

Bulb Security

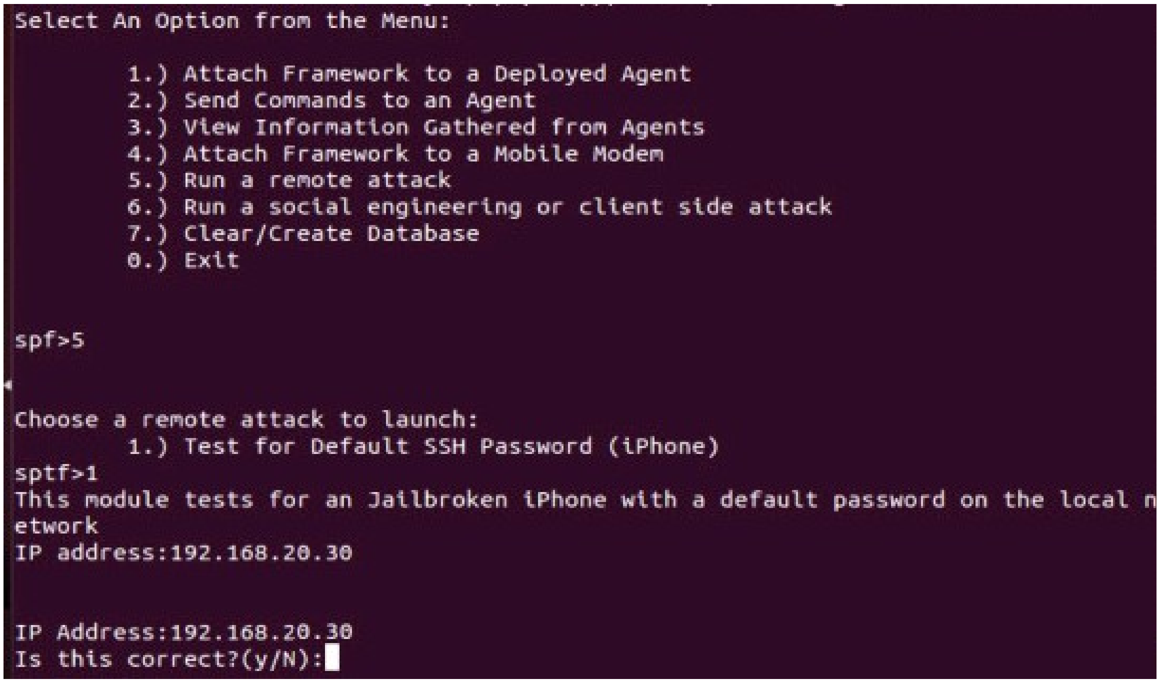
192.168.20.35

/manualtest

KEYKEY1

Remote Attacks:

SPF allows you to run remote attacks against smartphone platforms, via HTTP or SMS. To run a remote attack choose option **2**. In version 0.1 a remote attack is implemented to check for the default SSH password (“alpine”) on jailbroken iPhones. Select option **1**. You will be prompted for the IP address to test. Enter this value and type **y** when asked if this is correct.



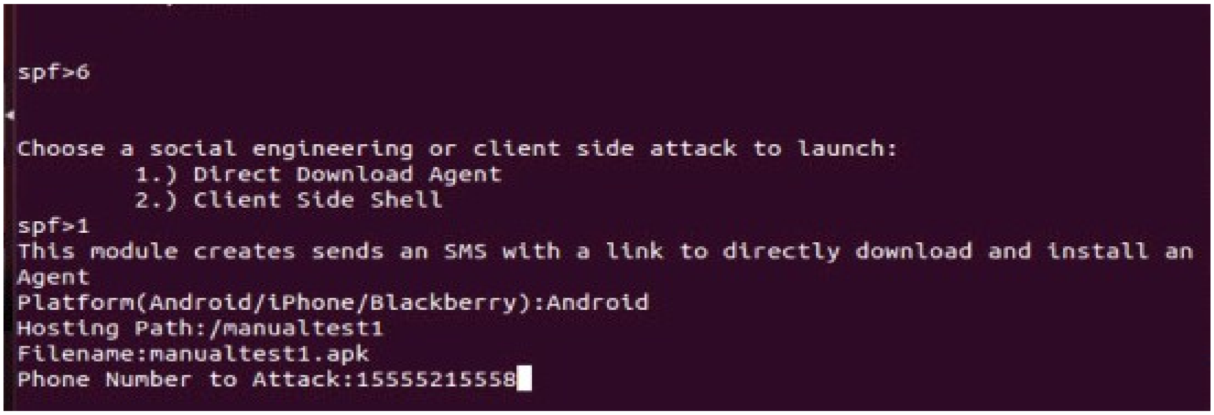
```
Select An Option from the Menu:

        1.) Attach Framework to a Deployed Agent
        2.) Send Commands to an Agent
        3.) View Information Gathered from Agents
        4.) Attach Framework to a Mobile Modem
        5.) Run a remote attack
        6.) Run a social engineering or client side attack
        7.) Clear/Create Database
        0.) Exit


spf>5


Choose a remote attack to launch:
        1.) Test for Default SSH Password (iPhone)
sptf>1
This module tests for an Jailbroken iPhone with a default password on the local n
etwork
IP address:192.168.20.30


IP Address:192.168.20.30
Is this correct?(y/N):
```

Three things may happen, either there will be no SSH server present at the IP address given in which case the SSH connection will time out, there will be an SSH server but its password will not be the default, in which case the login will fail, or the iPhone will be vulnerable. If the iPhone is vulnerable, SPF will log in and install the iPhone agent (currently a stub). Successes and failures will be recorded in the database.

Client-Side and Social Engineering Attacks:

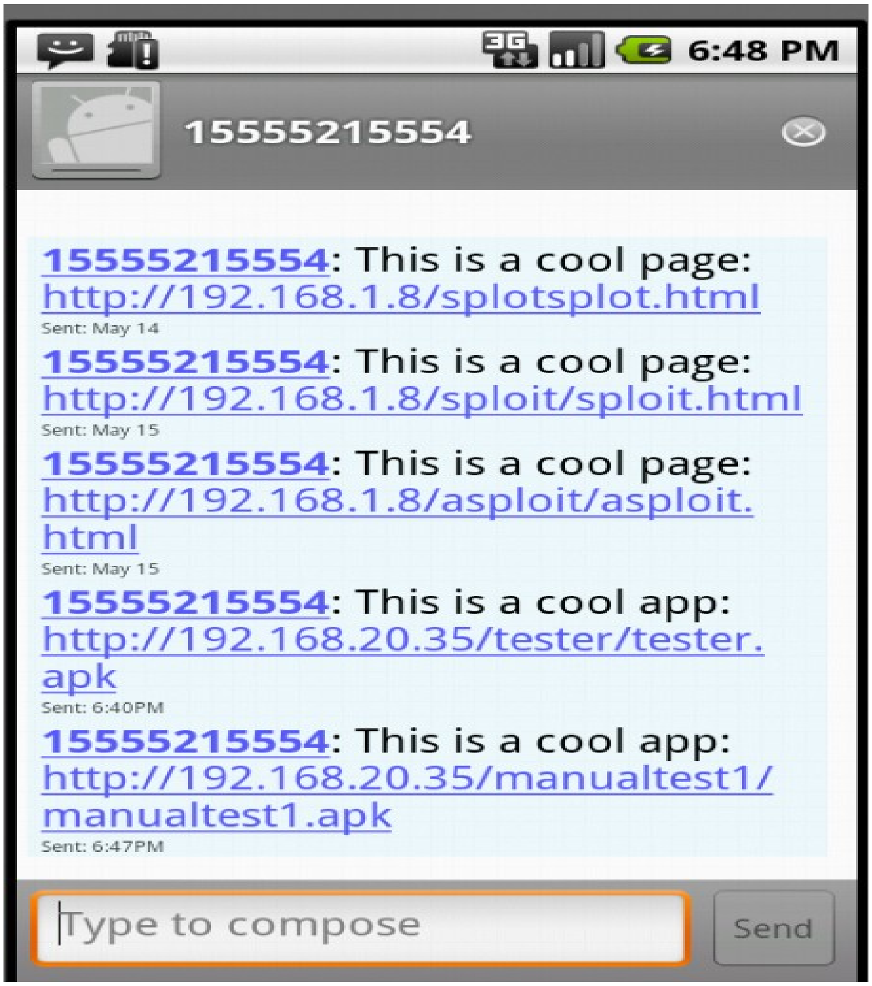
SPF allows you to create and host malicious webpages, PDFs etc. to launch client side attacks against smartphone platforms. Additionally, SPF tests smartphone users' susceptibility to SMS based social engineering attacks. To launch such attacks select option **6**.

Two attacks are currently implemented in this category. The first one is a social engineering attack that sends an SMS to a smartphone and requests that they download and install a “cool app.” The app will of course be an SPF agent. To launch this attack select option **1**. You will be prompted for the webserver path to host the app, the filename, the smartphone platform, and the phone number to attack.



```
spf>6

Choose a social engineering or client side attack to launch:
        1.) Direct Download Agent
        2.) Client Side Shell
spf>1
This module creates sends an SMS with a link to directly download and install an
Agent
Platform(Android/iPhone/Blackberry):Android
Hosting Path:/manualtest1
Filename:manualtest1.apk
Phone Number to Attack:15555215558
```

Using this data SPF will host the agent for the selected platform on the webserver at the path specified (this is currently only useful for Android victims as the iPhone and Blackberry agents are still in development). If more than one mobile modem is currently attached to SPF you will be prompted to choose which modem you would like to use to send the SMS. An SMS is then sent to the phone number provided with a link to download the app.



6:48 PM

15555215554

15555215554: This is a cool page: http://192.168.1.8/splotsplot.html
Sent: May 14

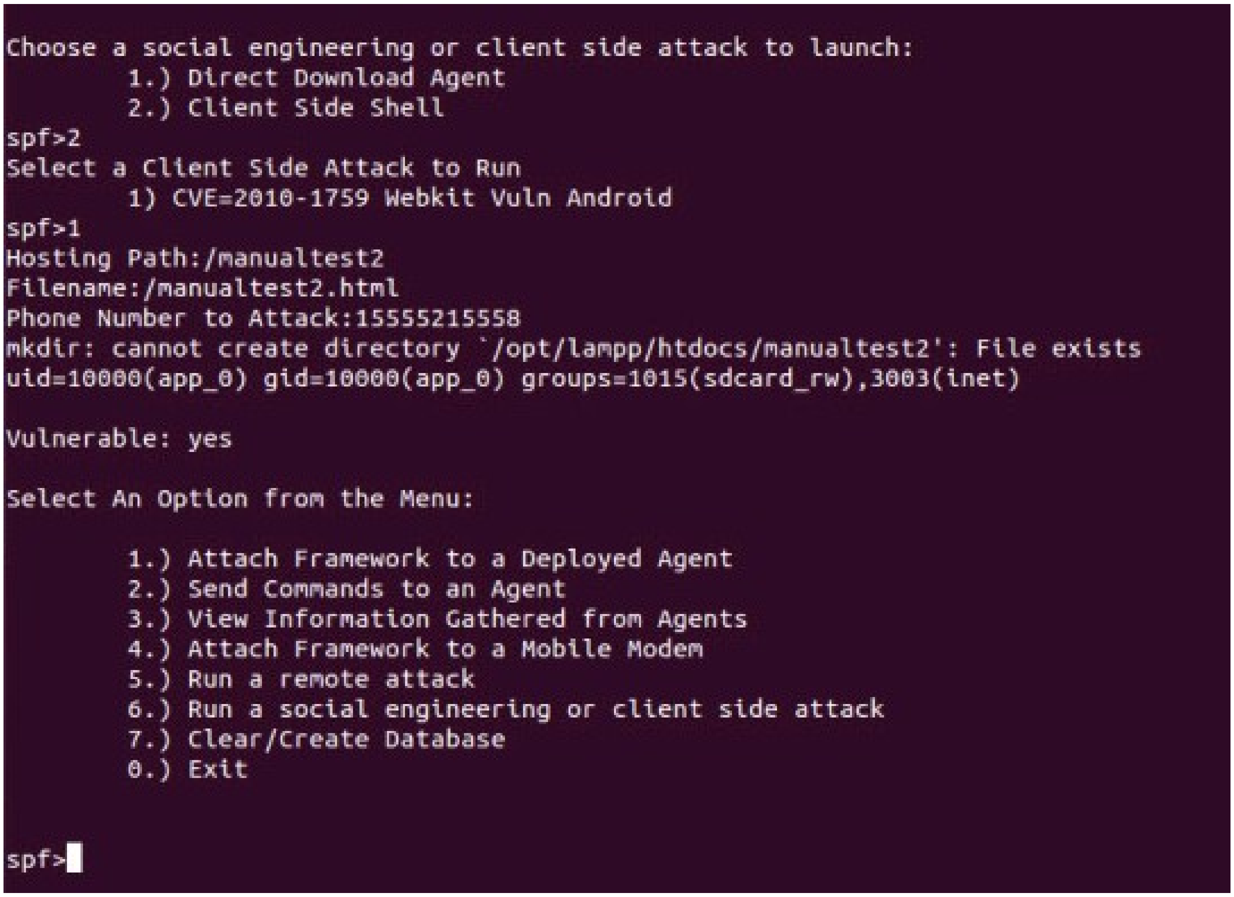
15555215554: This is a cool page: http://192.168.1.8/sploit/sploit.html
Sent: May 15

15555215554: This is a cool page: http://192.168.1.8/asploit/asploit.html
Sent: May 15

15555215554: This is a cool app: http://192.168.20.35/tester/tester.apk
Sent: 6:40PM

15555215554: This is a cool app: http://192.168.20.35/manualtest1/manualtest1.apk
Sent: 6:47PM

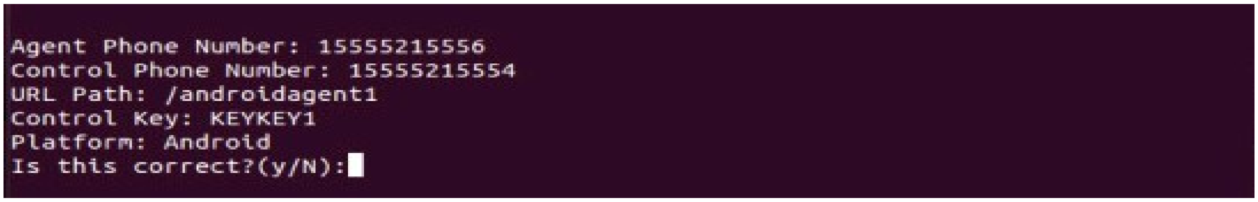
The other currently implemented option is an attack against the mobile browser on Android (CVE-2010-1759 Webkit). Instead of selecting option 1 for direct download, select option **2**. You will be prompted for the hosting path, file name, and phone number to attack. A malicious page that attempts to exploit the vulnerability is crafted and hosted on the webserver. A socket listening for a remote connection from an exploited is started. If more than one mobile modem is currently attached to SPF you will be prompted to choose which modem you would like to use to send the SMS. An SMS is then sent to the phone number provided with a link to a "cool page." If the smartphone user opens the link in the mobile browser, the page will attempt to exploit the Webkit vulnerability. If it is successful, the smartphone will connect to the listening socket. SPF will run a command on the exploited smartphone, close the connection, and record the results. Keeping a shell open for SPF user interaction is a planned feature. If the user does not click on the link or the browser is not vulnerable the socket will timeout.



```
Choose a social engineering or client side attack to launch:
        1.) Direct Download Agent
        2.) Client Side Shell
spf>2
Select a Client Side Attack to Run
        1) CVE=2010-1759 Webkit Vuln Android
spf>1
Hosting Path:/manualtest2
Filename:/manualtest2.html
Phone Number to Attack:15555215558
mkdir: cannot create directory `/opt/lampp/htdocs/manualtest2': File exists
uid=10000(app_0) gid=10000(app_0) groups=1015(sdcard_rw),3003(inet)

Vulnerable: yes

Select An Option from the Menu:

        1.) Attach Framework to a Deployed Agent
        2.) Send Commands to an Agent
        3.) View Information Gathered from Agents
        4.) Attach Framework to a Mobile Modem
        5.) Run a remote attack
        6.) Run a social engineering or client side attack
        7.) Clear/Create Database
        0.) Exit


spf>
```

Attach to a Deployed Agent:

You can attach SPF to a manually deployed agent. This basically just adds it to the database so you can interact with it using SPF. Choose option **1** at the main menu. You will be asked for information about the agent including the phone number, control phone number (modem used to receive results via the mobile network), platform, 7 digit control key, and HTTP control URL. Most of this is currently hardcoded in the agent itself, so just make sure it matches.



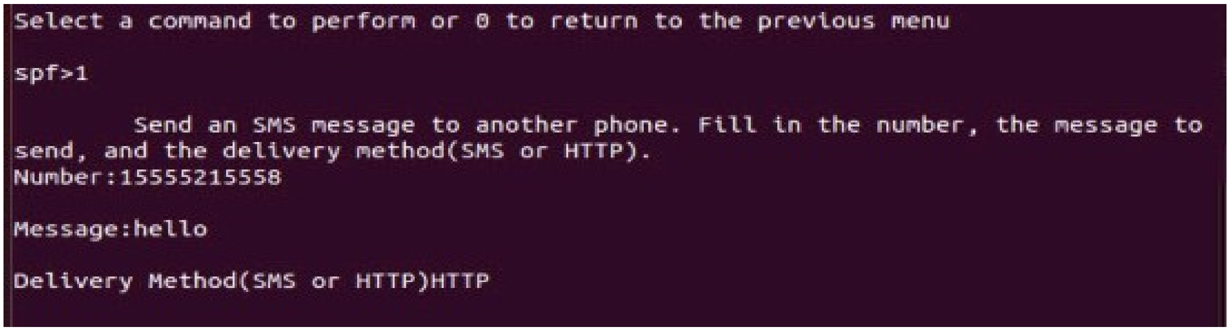
```
Agent Phone Number: 15555215556
Control Phone Number: 15555215554
URL Path: /androidagent1
Control Key: KEYKEY1
Platform: Android
Is this correct?(y/N):
```

Send Commands to Agents:

By choosing option **2** at the main menu you can send commands to agents installed on remote smartphones. The command currently supported are:

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

All commands may be delivered to agents via SMS or HTTP. Some commands return data while others do not. Of those that return data, you will be prompted to choose between SMS and HTTP for the return method if both are available. You will first be asked to select an agent to interact with. Currently only Android agents are available, but iPhone and Blackberry agents are in production.



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

spf>1

        Send an SMS message to another phone. Fill in the number, the message to
send, and the delivery method(SMS or HTTP).
Number:15555215558

Message:hello

Delivery Method(SMS or HTTP)HTTP
```

The Send SMS function allows you to send an SMS to another phone using the mobile modem from the agent infected smartphone.

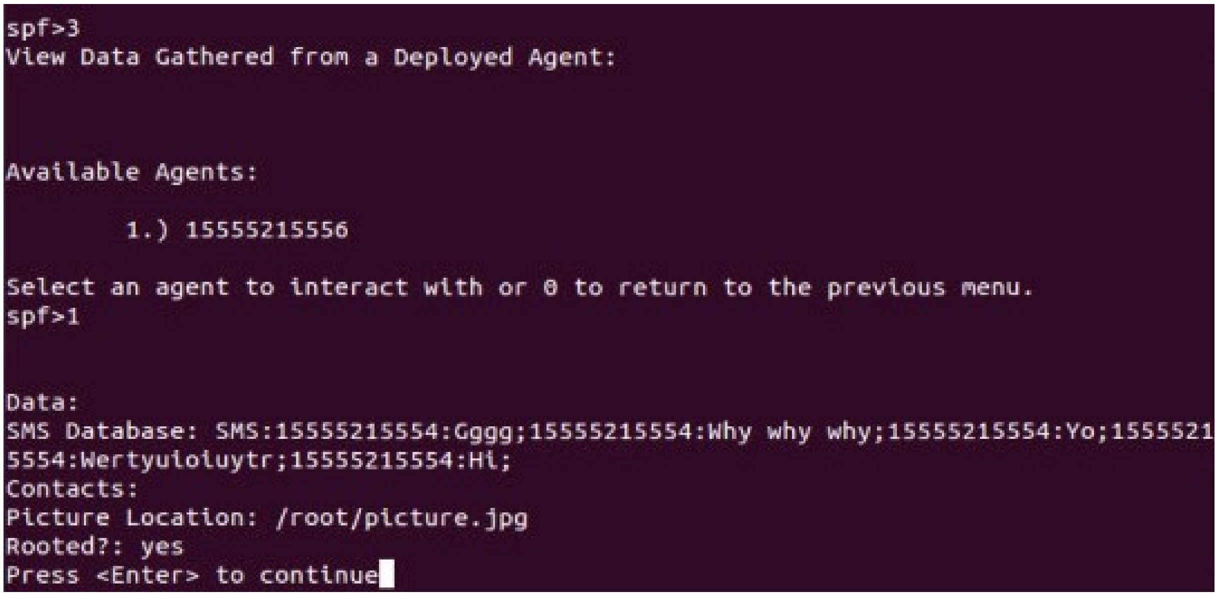
Take picture takes a picture if possible and uploads it to the webserver.

Get Contacts and Get SMS Database return the contacts and the last 10 SMSs received by the agent infected smartphone.

Privilege Escalation attempts to root the infected smartphone. Currently this functionality attempts to use the rageagainstthecage exploit against Android phones. If successful it will return that it worked and then drop the root privileges. Further use of root privileges is in development.

View Data Gathered from Agents:

Use option **3** to see the results of agent commands that return data. You will be prompted to choose an agent.

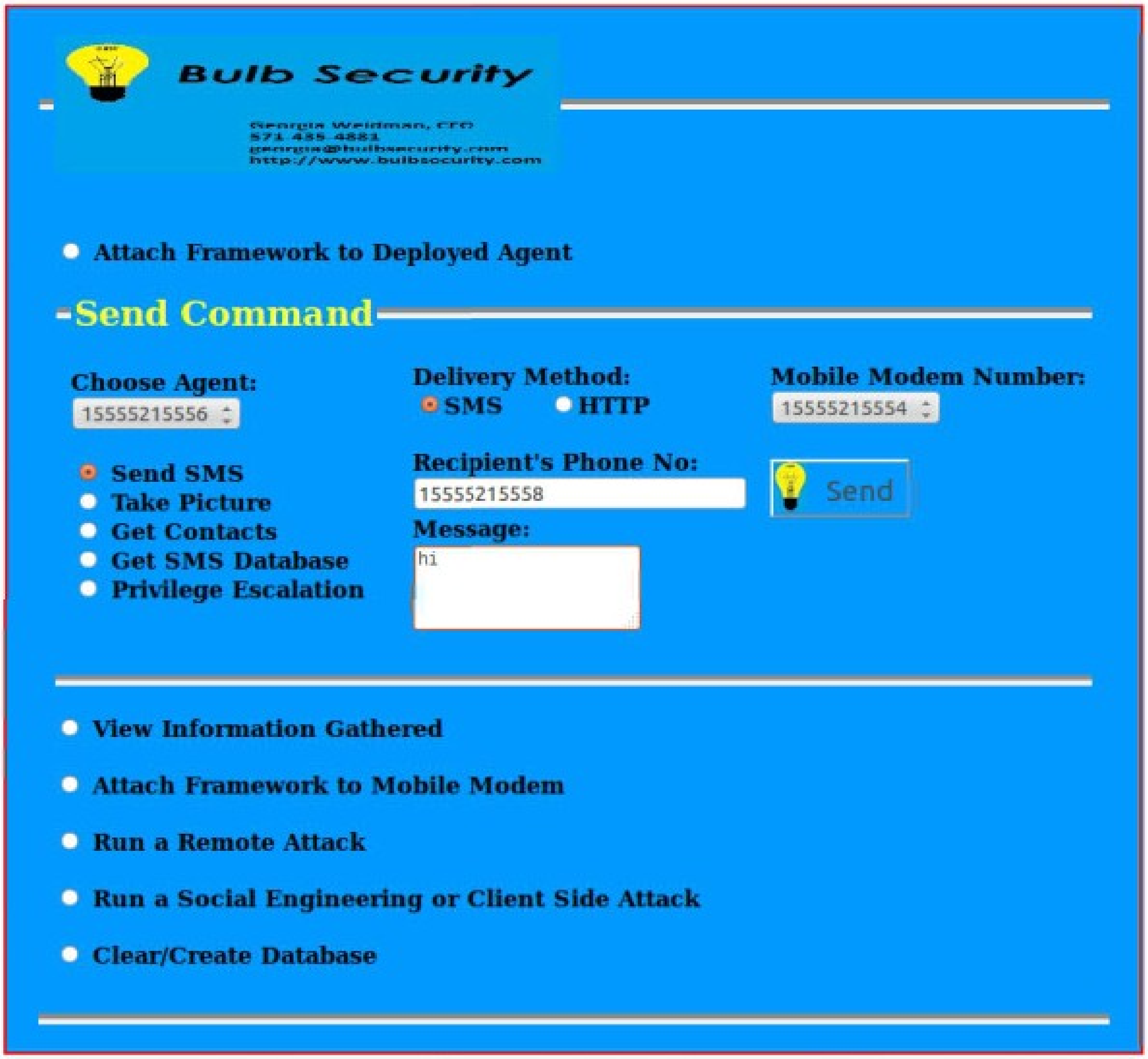


```
spf>3
View Data Gathered from a Deployed Agent:



Available Agents:

        1.) 15555215556

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


Data:
SMS Database: SMS:15555215554:Gggg;15555215554:Why why why;15555215554:Yo;1555521
5554:Wertyuioiuytr;15555215554:Hi;
Contacts:
Picture Location: /root/picture.jpg
Rooted?: yes
Press <Enter> to continue
```

Using the SPF GUI

The SPF GUI includes all the same functionality as the SPF console. As its name suggests it is graphical rather than text based. If you have not already done so, edit the config file in the SPF GUI folder to suit your environment. Browse to http://localhost/menu.pl

For information about the functionality see the SPF console section. The functionality is identical here. However, instead of choosing menu options we use text boxes, radio buttons, etc. Functionality is performed in the background only, whereas from the console, some functionality may be performed in the foreground. After functionality is run, the page refreshes and the database is queried for the most up to date data.



○ Attach Framework to Deployed Agent

Send Command

Choose Agent: 15555215556

Delivery Method: ● SMS ○ HTTP

Mobile Modem Number: 15555215554

● Send SMS
○ Take Picture
○ Get Contacts
○ Get SMS Database
○ Privilege Escalation

Recipient's Phone No: 15555215558

Message: hi

○ View Information Gathered

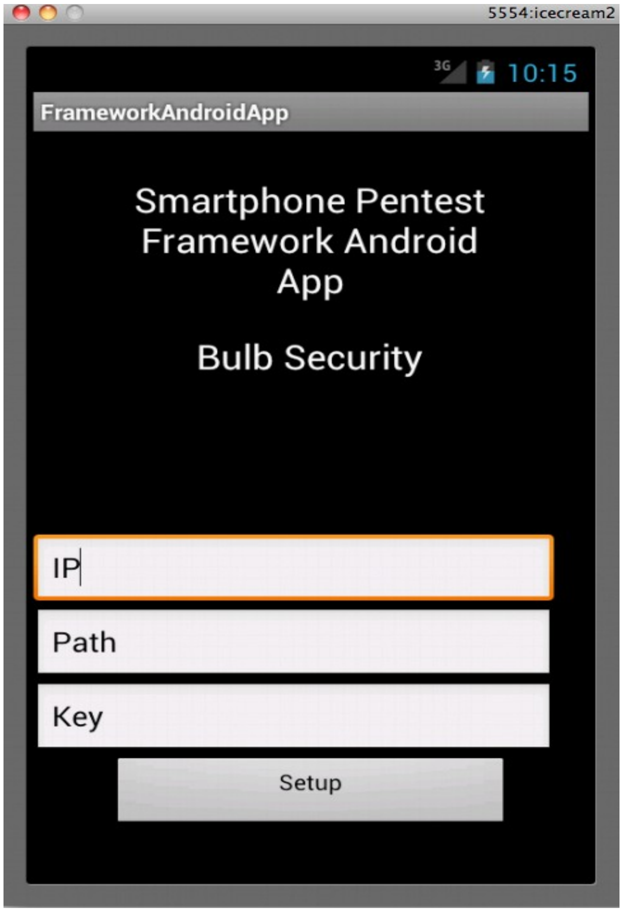
○ Attach Framework to Mobile Modem

○ Run a Remote Attack

○ Run a Social Engineering or Client Side Attack

○ Clear/Create Database

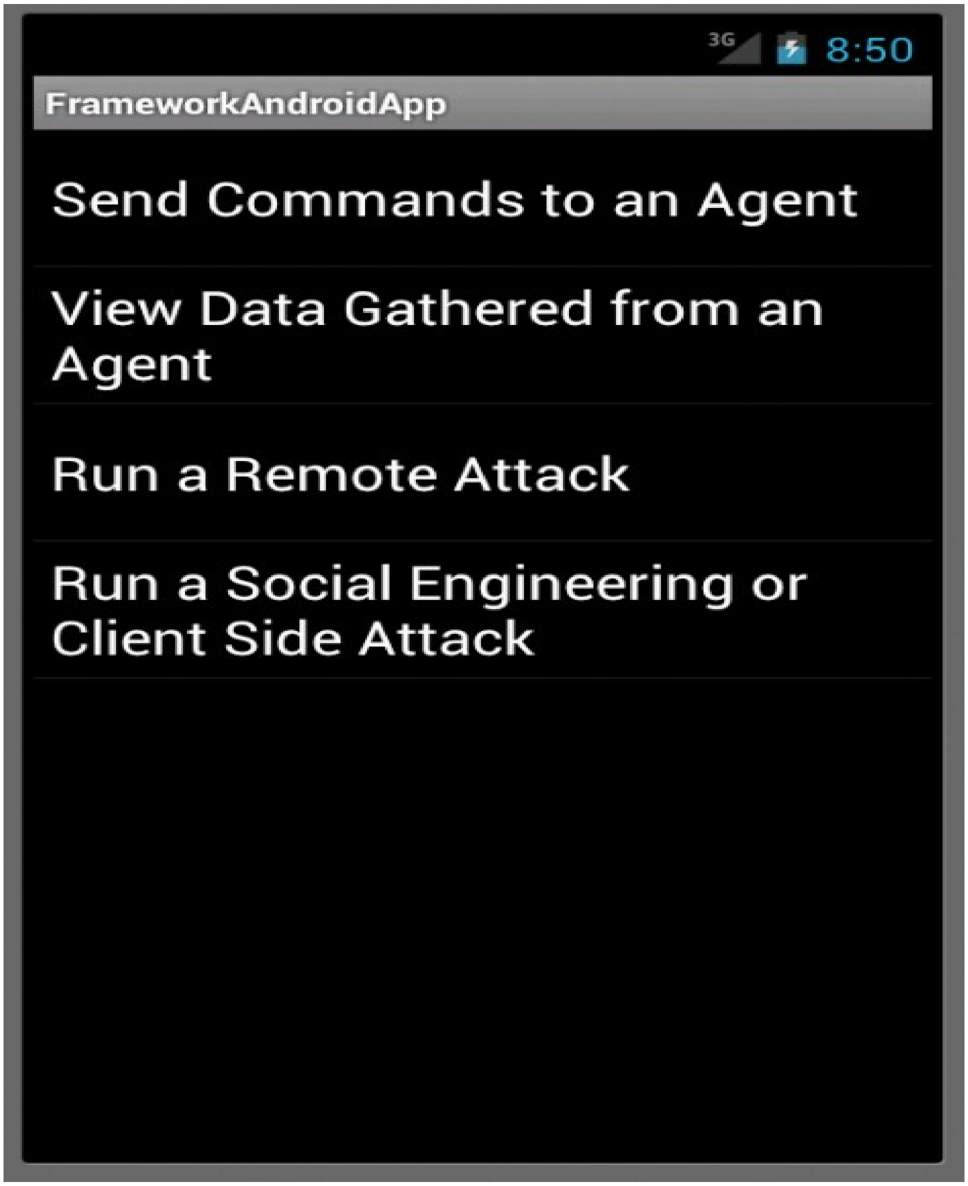
Using the SPF Android App



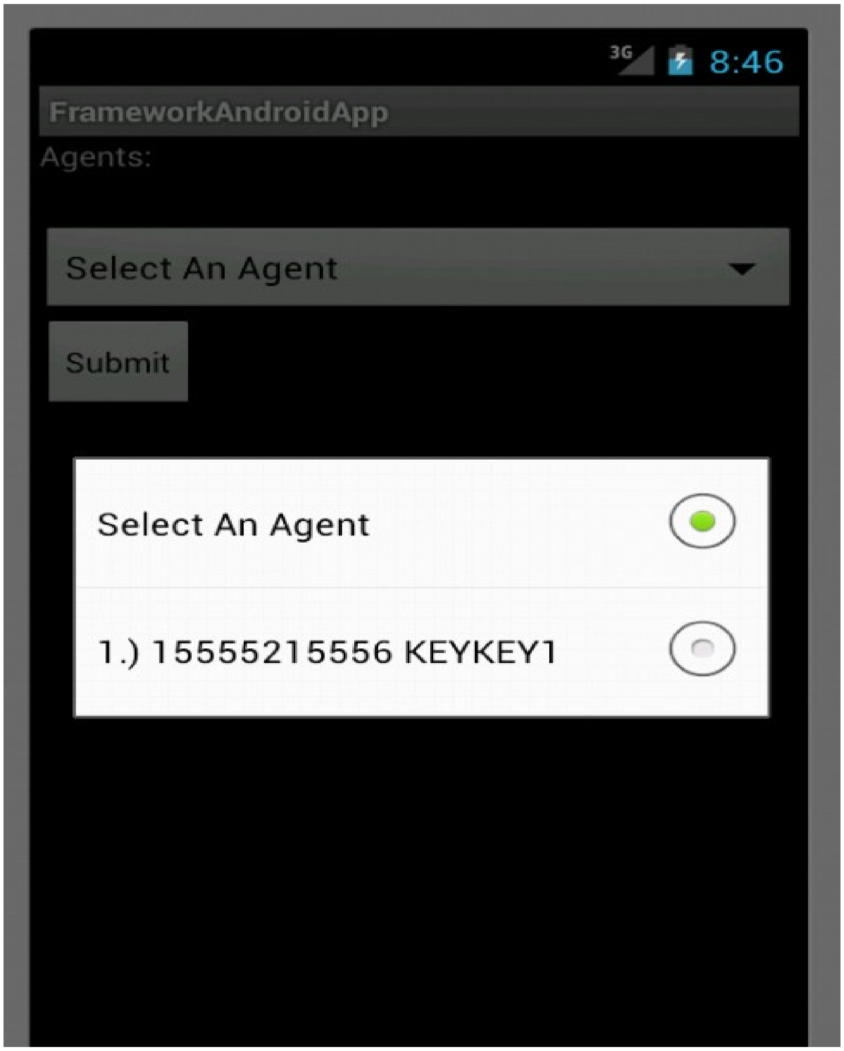
Smartphone Pentest Framework Android App

Bulb Security

To attach the SPF Android app to a SPF console or GUI server, use the process described in the SPF console section. After attachment, you will be taken to the functionality menu. Additionally, the mobile modem of the device can now be used by SPF for communication.


3G
8:50
FrameworkAndroidApp
Send Commands to an Agent
View Data Gathered from an Agent
Run a Remote Attack
Run a Social Engineering or Client Side Attack

From the SPF Android app you can perform a subset of the functionality available in SPF console and SPF GUI. For instance you cannot recreate the database inside of the app, as that would detach the app. The functionality usage is the same as in SPF GUI and console except in the app it looks a little different of course.



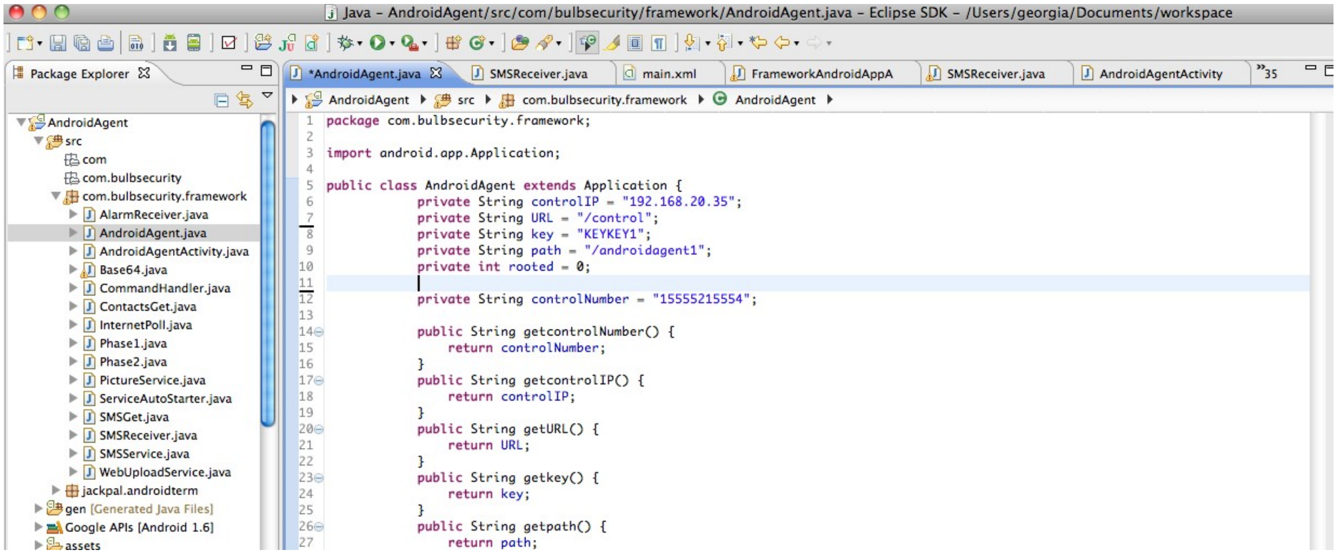
Select An Agent ●

1.) 15555215556 KEYKEY1 ○

Also all agent commands default to SMS delivery and SMS return if available using the mobile modem in the app's host smartphone. To detach the app, just use the back button until you reach the attachment screen you saw when you first started the app.

Building Agents:

Deploying agents is currently not a very simple process. Variable for the agents are hardcoded, so you will need to edit the source code (provided) and recompile them using you Android development environment. Particularly the control key, the path, and the control IP are hardcoded in AndroidAgent.java and the control key and control phone number are hardcoded in SMSReceiver.java. Change these values to fit your needs and build the agents using your Android development environment. Cleaning up this process and the ability to build custom agents from inside SPF is in development.



```java
package com.bulbsecurity.framework;

import android.app.Application;

public class AndroidAgent extends Application {
            private String controlIP = "192.168.20.35";
            private String URL = "/control";
            private String key = "KEYKEY1";
            private String path = "/androidagent1";
            private int rooted = 0;

            private String controlNumber = "15555215554";

            public String getcontrolNumber() {
                return controlNumber;
            }
            public String getcontrolIP() {
                return controlIP;
            }
            public String getURL() {
                return URL;
            }
            public String getkey() {
                return key;
            }
            public String getpath() {
                return path;
```