5-2014

Capturing and Analyzing Network Traffic from Common Mobile Devices for Security and Privacy

Billy Overton

Capturing and Analyzing Network Traffic from Common Mobile Devices for Security and Privacy

Thesis submitted in partial fulfillment of Honors

By

Billy Overton
The Honors College
University Honors Program
East Tennessee State University

February 28, 2014

Dr. Michael Lehrfeld, Faculty Mentor

Billy Overton, Student

ABSTRACT

Mobile devices such as tablets and smartphones are becoming more common, and they are holding more information. This includes private information such as contacts, financial data, and passwords. At the same time these devices have network capability with access to the Internet being a prime feature. Little research has been done in observing the network traffic produced by these mobile devices. To determine if private information was being transmitted without user knowledge, the mobile capture lab and a set of procedures have been created to observe, capture and analyze the network traffic produced by mobile devices. The effectiveness of the lab and procedures has been evaluated with the analysis of four common mobile devices. The data analyzed from the case studies indicates that, contrary to popular opinion, very little private information is transmitted in clear text by mobile devices without the user's knowledge.

TABLE OF CONTENTS

INTRODUCTION

Overview

The security and privacy of mobile devices is becoming more of a concern as the use of these devices increases. Users now use mobile devices to store sensitive private information including emails, passwords, bank account details, and contacts. Because of the number of different device types and mobile operating systems, analyzing the security of mobile devices is difficult. In order to make the task easier, the Mobile Capture Lab was created along with a simple procedure for testing mobile devices. The Mobile Capture Lab is a small, portable device that can capture and analyze network data produced by mobile devices over Wi-Fi.

User Perception of Mobile Security

The Mobile Testing Lab was created to assess the validity of end users' perceptions about the insecurity of mobile devices. In their paper *Measuring user confidence in smartphone security and privacy*, Chin, Felt, Sekar, and Wagner used surveys and interviews to discover if people had concerns about using their mobile devices for sensitive work. In order to determine users' confidence in their smartphones' security and privacy, Chin et al interviewed people who owned both a smartphone and a laptop. Volunteers were solicited via an advertisement posted to the “Et cetera jobs” board of the San Francisco Bay Area Craigslist. Volunteers were required to be 18 or older, have a personal smartphone, and have a personal laptop. Thirty men and 30 women were selected from 282 volunteers, including 19 who were 18-27 years old, 14 who were 28-37 years old, 14 who were 38-47 years old, and 13 who were 48 years old

or older. Participants were also selected to balance the demographics between the different laptop and smartphone operating systems (Chin, Felt, Sekar, & Wagner, 2012, p. 3).

The data was collected during a 50-90 minute session. During this session each participant completed surveys concerned with the participant's laptop usage, laptop applications, smart phone usage, and smartphone applications. Participants also ranked the factors they consider when selecting smartphone and laptop applications. Following the session, participants were interviewed about their willingness to do nine tasks on their laptops and smartphones: using location aware services, using applications or websites that charge them money, accessing their bank accounts, managing financial data, making purchases, reading work related email, entering their Social Security Number (SSN), managing health documents, and sharing photos. To avoid bias, security and privacy were not mentioned explicitly until the end of the interview, where participants were asked how they compared security and privacy on their smartphones to their laptops (Chin, Felt, Sekar, & Wagner, 2012, pp. 3-4).

Chin et al. found statistically significant differences between laptop and smartphone usage for entering SSNs, examining health data, accessing bank accounts, and shopping. Four participants refused to enter their SSN on their laptops while 41 refused to enter their SSN on their smartphone, 36 of whom cited security as their main concern. Eight of the 60 refused to use their laptops to access health information while 19 participants refused to use smartphones for this purpose, 9 of whom cited security and privacy as a concern. One of the 60 refused, for security reasons, to use their laptop to access bank account information, while 10 refused to access this information

on their phone, 8 citing security reasons and 2 the inability to do so. For shopping, all participants were willing to make purchases with their laptops while 17 would not shop on their phones, 11 citing concerns about security (Chin 2012, pp. 4-6).

Overall, participants were less willing to enter private information on their phones due to perceived security risks. Other factors that contributed to this included perceived usability issues with mobile applications and a lack of discussion regarding the security of private information on mobile devices (Chin 2012, p. 6).

While little difference was found between users' concerns about the platforms' security, Chin et al. found that significantly more participants were more concerned about privacy on their smartphones than on their laptops. Those who were more concerned about the security and privacy of their laptops stated that they do more personal activities on their laptop than on their phones. Overall there was a perception that 3G and Wi-Fi were less secure on smartphones than on laptops (Chin 2012, pp. 6-7).

Participants were asked broad-based questions about their primary concerns regarding their smartphones, so as not to bias responses toward security and privacy concerns. Concerns cited included physical damage, data loss, and phone loss. Of those that identified phone loss, most participants expressed concerns about breaches of security and privacy that could result from a loss instead of just financial losses. Participants also mentioned a distrust of installed applications, with at least one participant providing false information to applications to ensure a sense of privacy (Chin 2012, pp. 6-7).

Chin et al. found that participants were more willing to experiment with applications on their phones than on their laptops. Participants mostly installed smartphone applications from official application markets designed for their devices such as the Apple App Store. While participants considered factors such as brand name, price, and existing security features when installing applications on phones and laptops, they were much more likely to install free applications that were not associated with a known brand on their smartphones. For both platforms the participants rated an application's popularity and the number of positive reviews as the most important factors for installing new applications. Current security and privacy features such as the EULA and privacy agreements were rated as less important for most participants (Chin 2012, pp. 7-10).

Participants' security and privacy concerns were found to be correlated to the demographic groups. Chin et al. found that younger participants were more likely to enter their SSN on their smartphone and were more concerned with phone privacy. The authors also found that the longer participants owned their phone, the less they considered price a factor in application installation and the less they worried about mobile security (Chin 2012, pp. 10-11).

MOBILE CAPTURE LAB CREATION

Overview

The mobile capture lab was inspired by a similar device created by Bruce Nikkel in his 2006 paper *A Portable Network Forensic Evidence Collector*. The Portable Network Forensic Evidence Collector (PNFEC) device was a small embedded system designed to serve as a low cost, network forensic evidence collection device for individuals without forensic training (Nikkel, 2006, pp. 127-129). The PNFEC, when placed in-line on a wired network, stores all network traffic passing through it on a dedicated hard drive. When a capture was completed, a cryptographic hash would be computed on the capture file to ensure no modification of the data occurred (Nikkel, 2006, pp. 129-130).

Like the PNFEC device, the mobile capture lab was designed with the goals of being self-contained, based on consumer level hardware, and as automated as possible. In addition to these goals, the mobile capture lab was also designed to perform some network capture analysis on the device itself. In order to meet these goals, a consumer level wireless access point/router was selected to serve as the mobile capture lab's hardware base. This provided Wi-Fi access to mobile devices, the ability to capture and store produced network traffic, and the ability analyze the captured data all on single device.

The mobile capture lab was designed to store all network traffic that passed through its wireless interface to a dedicated USB drive. Like PNFEC, it computes cryptographic hashes on all captured data to allow the detection of modifications to

captured data. Due to hardware and legal limitations, the mobile lab was designed to only capture network traffic transmitted via Wi-Fi. This means devices that send network traffic through other means, such as cellular data, cannot be completely analyzed with the current lab configuration.

Hardware

A BUFFALO AirStation HighPower N300 Gigabit Wireless Router (Model WZR-300HP) was chosen for the lab's base hardware. This router was selected primarily due to its support of the OpenWrt Linux distribution and its inclusion of an onboard USB port. OpenWrt is a Linux distribution designed to be installed on embedded devices such as commercial routers (OpenWrt Index). The existence of an onboard USB port was necessary to provide a location to store network captures. The router provides an Atheros AR7242 400MHz CPU which is used for capturing and analyzing the traffic generated by connected mobile devices. In addition, the N300 router provides Wi-Fi 802.11n support for modern devices.

Firmware Installation

To provide a modifiable software platform on which to base the mobile capture lab, OpenWrt was installed on the router. The version installed and supported by the BUFFALO N300 router was OpenWrt Attitude Adjustment (12.09-rc1, r34185). Installation instructions can be found in the OpenWrt documentation on the hardware support page for the BUFFALO N300 router (Buffalo WZR-HP-G300NH2 / WZR-300HP).

Once installation is completed, the router becomes manageable by connecting as root using Secure Shell (SSH). Additional router software can then be installed using OpenWrt's opkg package manager. Before installing any packages, the available package list needs to be downloaded to the device using the command **opkg update**.

In order to enable the onboard USB port the following packages were installed using the command **opkg install <package name>**:

- kmod-usb-storage
- kmod-usb-storage-extras
- block-mount
- kmod-fs-vfat
- kmod-nls-cp437
- kmod-nls-iso8859-1

Once these packages were installed, the router was able to access and use FAT32 formatted USB devices. This was done to provide a location to store captured network traffic and log files produced during mobile device testing. In addition to the above packages, the packages tcpdump and ngrep were installed.

Internal Network Architecture

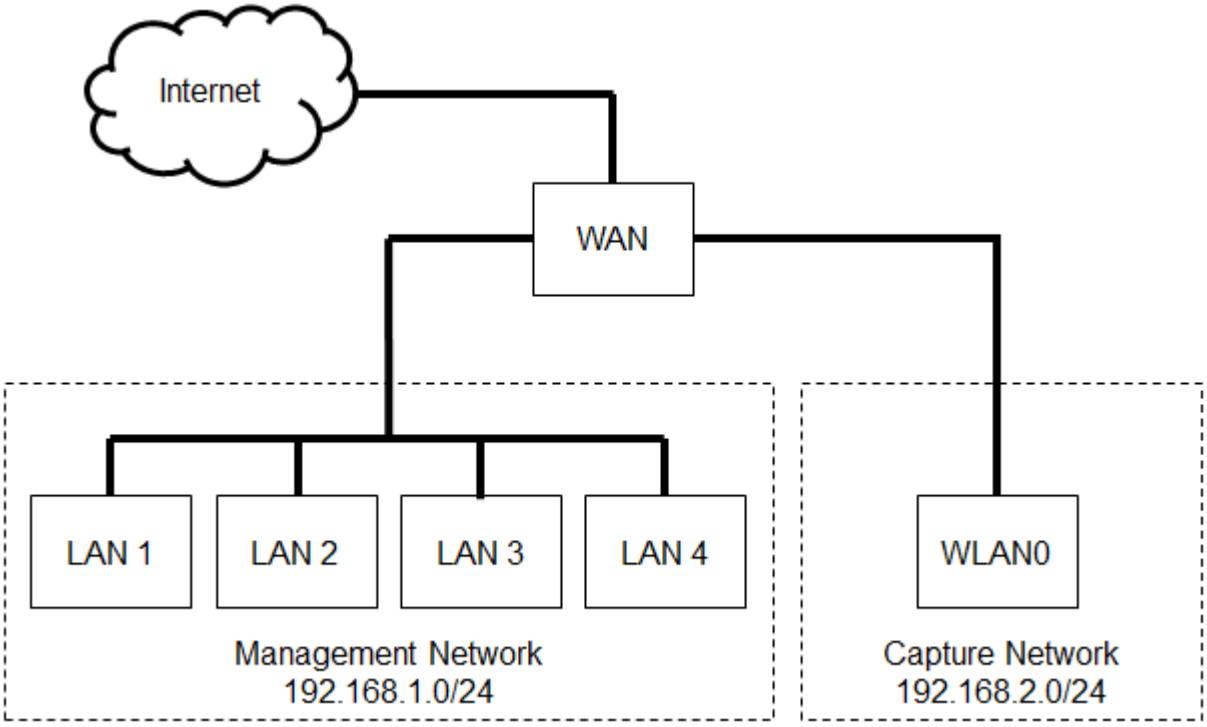

Figure 1: Network Architecture Diagram

The router was configured to have two separate internal networks: a capture network and a management network. As seen in Figure 1, the capture network consists of the wireless interface WLAN0. The management network consists of the four physical Local Area Network (LAN) ports on the back of the router. Both the management interface and the capture interface can reach the Internet via the Wide Area Network (WAN). No communication between the management network and the capture network was allowed to avoid contamination of captured data. Network captures are performed on the WLAN0 interface, and thus the capture network, only. See APPENDIX A for complete configuration settings.

MOBILE LAB PROCEDURES

Testing Procedure

A test is a single task that is performed on a mobile device to generate network traffic. In order to relate a set of captured data to a test on a mobile device, a new network capture is started at the beginning of a test and stopped at the end. Captures are started and stopped using the capture.sh shell script (Appendix B). The script takes a single string as a single command-line argument. This string is appended to the produced log file and serves as a label for the captured data. Capture.sh uses tcpdump on the router's WLAN0 interface to generate the network dump and md5sum to compute the cryptographic hashes after the test is complete. Currently capture.sh is launched manually via a SSH connection on the management network.

Capture Analysis

Captures were analyzed using the nscan.sh script (Appendix C). Nscan.sh uses ngrep to search network capture files produced by capture.sh for a series of keywords. Ngrep is command-line string searching tool that can search the pcap files produced by tcpdump and capture.sh (Ritter, 2006). Keywords are placed in a text file, with each keyword on a separate line. When provided a keyword and pcap file, nscan.sh produces a report cataloging found instances of each keyword. Keywords for analysis were selected based on the information used on each mobile device selected. Keywords used include full name, username, email address, phone number, and plain-text passwords. In addition to the device-specific keywords, two generic keywords were used the word “email” and the word “password.”

CASE STUDIES

Four devices (iPad3, Kindle Fire, Kindle PaperWhite, and Nexus 7) were used as case studies for developing and using the Mobile Capture Lab and the testing procedures.

Apple iPad3

Tests Performed

Performing Initial Device Setup.

The initial device configuration steps provided by the iOS prompts were followed, using common settings for an English speaking user. During this test, the device was connected to the mobile capture lab for the first time. Information such as account logins for email and Apple IDs were entered post WiFi connection to ensure the maximum amount of data captured that could potentially show the transmission of a user's private data.

Syncing of iOS's Built-in Mail Application with Gmail

The device was configured to connect and sync with the Gmail service by following the settings guided provided by Google on their support site. (Google)

Installing Facebook application

The Facebook application was installed by opening the Apple App Store, searching for the word "Facebook", selecting install for the official Facebook application,

and waiting for the installation to be complete. The App Store application was then closed.

Setting up of Facebook Application

The application was launched and the username and password created for the device were entered. After the initial login, the Facebook app was terminated.

Posting with Facebook Application

The Facebook app was launched and the text “First post” was entered as a wall post. The application was then terminated.

Installing Twitter Application

The Twitter application was installed by opening the Apple App Store, searching for the word “Twitter”, selecting install for the Twitter application, and waiting for the installation to complete. After the application was launched, the username and password created for the device were entered, and the application was terminated.

Setting up of Twitter Application

After launching the application, the username and password created for the device were entered, and the application was terminated.

Posting with Twitter Application

The Twitter app was launched and the text “First post” was sent out as a tweet.

Results

Apple iPad 3	
Test(s)	**Sensitive Data Transmitted Without Encryption**
Performing initial device setup Syncing of iOS's built-in mail application with Gmail Installing Facebook application Posting with Facebook application Installing Twitter application Setting up Twitter application Posting with twitter application.	None
Setting up Facebook application	User's email address

Table 1: Apple iPad 3 Summary

As show in Table 1, of the tests performed only the setting up of the Facebook application sent any of the keywords to the Internet without encryption. During this test, the mobile device sent the email address used to sign in to Facebook publicly.

Amazon Kindle Fire

Tests Performed

Performing Initial Device Setup

Information was entered into the device as prompted by the Kindle Fire setup application. This was the first time the device was connected to the mobile capture lab. All information was entered into the setup application following language selection and WiFi setup.

Linking to Twitter

The Kindle Fire settings menu was opened and the social media section was selected. After selecting to link to Twitter, the username and password for this device's twitter account were entered.

Linking to Facebook

The Kindle Fire settings menu was opened and the social media section was selected. After selecting to link to Facebook, the username and password for this device's Facebook account were entered.

Adding a Contact

The built in contact management application was opened and a new contact was created. The information from the iPad case study was used to create this contact, and the appropriate keywords (username, real name, etc) were added to the list to scan for.

Highlighting a Section of Text in a Book

By default, the Kindle fire came with an eBook version of its user manual. This manual was opened and a randomly selected sentence was highlighted using the built in highlighting tool. This test was performed to detect if a user's highlighting habits were transmitted in clear text.

Browsing the Book Store

The Amazon book store application was opened. Each of the common genres of books were opened and browsed until at least two pages of book results were displayed for each. The store application was then closed.

Downloading a Book Sample

The Amazon book store was opened and a random book was selected from the main page. A sample was downloaded for the book, and then opened and browsed till 50% of the pages were read. The book's title and page location were added to the keyword file for this device's analysis step.

Results

Amazon Kindle Fire	
Test(s)	**Sensitive Data Transmitted Without Encryption**
Performing initial device setup Linking to Twitter Linking to Facebook Adding a contact Highlighting a section of text in a book Browsing the book store Downloading a book sample	None

Table 2: Amazon Kindle Fire Summary

As shown in Table 2, none of the tests resulted in a clear-text keyword being sent to the Internet.

Amazon Kindle Paperwhite

Similar Tests

The following tests were performed in the same fashion as the Amazon Kindle Fire's tests. An additional test of forcing network sync was also performed by selecting Sync from the menu options.

- Performing initial device setup
- Browsing the bookstore
- Downloading a book sample
- Forcing a network sync

Results

Amazon Kindle Paperwhite	
Test(s)	**Sensitive Data Transmitted Without Encryption**
Performing initial device setup Browsing the book store Downloading a book sample Forcing a network sync	None

Table 3: Amazon Kindle Paperwhite Summary

As shown in table 3, none of the tests resulted in a clear-text keyword being sent to the Internet.

Google Nexus 7

Tests Performed

Similar Tests

The steps preformed for the Nexus 7 tests closely matches the Apple iPad3 tests for the following except the Google app store was used instead of the iOS app store:

- Performing initial device setup
- Installing Facebook application
- Setting up of Facebook application

- Posting with Facebook application
- Installing Twitter application
- Setting up of Twitter application
- Posting with Twitter application

In addition to the similar tests above, additional tests were performed.

Adding a Contact

The built in contact application (which syncs with the Gmail service) was opened, and the information from the iPad3 device was entered as a contact. The relevant information was added to the keyword file for the analysis step.

Opening a Book

One of the default books provided with the device was opened. With the book open, several pages were navigated by turning the pages manually, jumping to a specified page, and using the table of contents. The same book was used for the following test.

Highlighting Text in a Book

While the book for the “Opening a Book” test was open (but after the test was ended), a random sentence was highlighted in the book.

Results

Google Nexus 7	
Test(s)	**Sensitive Data Transmitted Without Encryption**
Performing initial device setup Installing Facebook application Posting with Facebook application Installing Twitter application Setting up of Twitter application Posting with twitter application Adding a contact Opening a book Highlighting text in a book.	None
Setting up of Facebook application	User's email address

Table 4: Google Nexus 7 Summary

Like the Facebook application on iOS, the Nexus Facebook application frequently sent the user's email address in clear-text. Nscan.sh produced a large number of false-positives for the keyword "password," but no sensitive information regarding the password was sent. The false positives were due to several sections of page information containing the word "password" to indicate a password should be entered.

CONCLUSIONS

Summary

Despite user concerns regarding the security and privacy of mobile devices as found by Chin et. al. (2012), preforming standard mobile actions on modern devices does not present a significant privacy concern. Of the devices and actions tested, only the Facebook application on the iOS and Kindle Fire platform leaked any user information. See tables 1-4 for a summary of the results. A mobile device's security and privacy does depend on the applications a user chooses to run on the device. The Mobile Capture Lab provides a quick way to evaluate the privacy and security of mobile applications.

Further Work

Improvements to the capture analysis can be made by searching network capture files for other non-encrypted resources besides textual information. These resources could include images and compressed files. Mapping where information is sent, regardless of the information’s encryption status, could reveal information being sent without the user’s consent.

Further work on automating the capture lab can also be done by removing the need to manually start and stop a capture. While this will reduce the ability to correlate generated traffic to specific tasks, it would provide a more holistic view of a particular mobile device’s usage.

WORKS CITED

Buffalo WZR-HP-G300NH2 / WZR-300HP. (n.d.). Retrieved December 2012, from OpenWrt: http://wiki.openwrt.org/toh/buffalo/wzr-hp-g300nh2

Chin, E., Felt, A., Sekar, V., & Wagner, D. (2012). Measuring user confidence in smartphone security and privacy. *SOUPS '12 Proceedings of the Eighth Symposium on Usable Privacy and Security* , 1-16.

Google. (n.d.). *Set up Google Sync with your iOS device*. Retrieved 02 15, 2013, from Google Support: https://support.google.com/a/users/answer/138740

Nikkel, B. J. (2006, September). A portable network forensic evidence collector. *Digital Investigation: The International Journal of Digital Forensics & Incident Response, 3*(3), 127-135.

OpenWrt Index. (n.d.). Retrieved 2 15, 2013, from OpenWrt: http://openwrt.org/

Ritter, J. (2006, 11 18). *ngrep - network grep*. Retrieved from http://ngrep.sourceforge.net/

APPENDIX A

Mobile Capture Lab OpenWRT Configuration

Install the following packages using the command **opkg install <package name>**:

- kmod-usb-storage
- kmod-usb-storage-extras
- block-mount
- kmod-fs-vfat
- kmod-nls-cp437
- kmod-nls-iso8859-1
- tcpdump
- ngrep

Enable the automated mounting of a FAT or FAT32 USB device at boot time by adding the following section to the /etc/config/fstab file.

```
config mount
    option  target /mnt/captures
    option device /dev/sda1
    option fstype vfat
    option options rw,sync
    option enabled 1
    option enabled_fsck 0
```

Creation of Network Separation

Add the follow section to the /etc/config/network file.

```
config interface 'wifi'
```

```
    option 'proto' 'static'
    option 'ipaddr' '192.168.2.1'
    option 'netmask' '255.255.255.0'
```

Modify the current wifi-iface entry in /etc/config/wireless file to match the configuration below. This puts the wifi interface on the newly created network and enables the WPA2-PSK encryption protocol.

```
config wifi-iface
    option device radio0
    option network wifi
    option mode ap
    option ssid ETSUMobileLab
    option encryption psk
    option key 'ETSUMobileLab1'
```

Add the following lines to the /etc/config/dhcp file to enable DHCP for the newly created network.

```
config dhcp wifi
    option interface wifi
    option start 100
    option limit 150
    option leasetime 12h
```

Finally, add the following to the /etc/config/firewall file to enable devices on the new network to communicate with the network via the WAN connection.

```
config zone
    option name wifi
    option network wifi
    option input ACCEPT
    option output ACCEPT
    option forward REJECT

config forwarding
    option src wifi
    option dest wan
```

APPENDIX B

```
#!/bin/ash
#
#--------------------------------------------------------------------
# File Name:                    capture.sh
# Project Name:                 ETSU Mobile Lab
#
#--------------------------------------------------------------------
# Creator's name and email:     Billy Overton overtonb@goldmail.etsu.edu
# Creation Date:                01/25/2013
# Date of Last Modification:    04/02/2013
#--------------------------------------------------------------------
# Purpose: Perform a test capture and generated associated log files
#
# Input:   Purpose (String describing the purpose of this capture)
# Output:  Capture file (pcap format)
#          Log file     (timestamp and output from tcpdump)
#          MD5 file     (md5 hashes of capture and log file)
#

_datetime=$(date +"%Y%m%dT%H%M%S")
_dirbase="/mnt/captures/mobileCapture-$_datetime"
_filebase="$_dirbase/mobileCapture-$_datetime"

_capturefile="$_filebase.pcap"
_md5file="$_filebase.md5"
_logfile="$_filebase.log"

# Make the capture directory
mkdir $_dirbase

# Turn on the Lock LED to indicate a capture has started
echo 1 >  /sys/class/leds/buffalo\:orange\:security/brightness

echo "Press ENTER to stop the capture."

# Add information to the log file for this capture
echo "TIMESTAMP: $_datetime" >> "$_logfile"
echo "Purpose: $@" >> "$_logfile"

# Start tcpdump and store the PID
tcpdump -s 0 -i wlan0 -w "$_capturefile" 1>> "$_logfile" 2>> "$_logfile"&
TCPDUMPPID=$!

# Read in characters, waiting for a newline to stop the capture
read garbagedata
```

```
# Stop tcpdump
kill "$TCPDUMPPID"

# Generate the md5sum of the capture file and the log
echo ""
echo "Generating MD5 Sum of Captured Data and Logs"
md5sum $_capturefile > "$_md5file"
md5sum $_logfile >> "$_md5file"

# Turn off the Lock LED to indicate the capture has been halted
echo 0 > /sys/class/leds/buffalo\:orange\:security/brightness
```

APPENDIX C

```
#!/bin/ash
#
#--------------------------------------------------------------------
# File Name:                    nscan.sh
# Project Name:                 ETSU Mobile Lab
#
#--------------------------------------------------------------------
# Creator's name and email:     Billy Overton overtonb@goldmail.etsu.edu
# Creation Date:                03/25/2013
# Date of Last Modification:    04/11/2013
#--------------------------------------------------------------------
# Purpose: Generated a string search report of a capture file based on
#          a supplied keyfile
#
# Input:   Keyword file  (newline separated file contained keywords)
#          Capture file  (pcap format capture file)
# Output:  Scan Log

NGREP="/usr/bin/ngrep -q -i -t"

_keywordfile="$1"
_capturefile="$2"

_scanlog="$_capturefile.scanlog"


# Create the scan log file
rm -f $_scanlog
touch $_scanlog


#Loop through each line in the keyword file
while read keyword
do
echo "==========================================" >> $_scanlog
$NGREP -I $_capturefile "$keyword"  >> $_scanlog
echo "==========================================" >> $_scanlog
echo >> $_scanlog
echo >> $_scanlog
done < $_keywordfile
```

APPENDIX D

Case Study Results Summary

Apple iPad 3	
Test(s)	**Sensitive Data Transmitted Without Encryption**
Performing initial device setup Syncing of iOS's built-in mail application with Gmail Installing Facebook application Posting with Facebook application Installing Twitter application Setting up Twitter application Posting with twitter application.	None
Setting up Facebook application	User's email address
Amazon Kindle Fire	
Test(s)	**Sensitive Data Transmitted Without Encryption**
Performing initial device setup Linking to Twitter Linking to Facebook Adding a contact Highlighting a section of text in a book Browsing the book store Downloading a book sample	None

Amazon Kindle Paperwhite	
Test(s)	**Sensitive Data Transmitted Without Encryption**
Performing initial device setup Browsing the book store Downloading a book sample Forcing a network sync	None
Google Nexus 7	
Test(s)	**Sensitive Data Transmitted Without Encryption**
Performing initial device setup Installing Facebook application Posting with Facebook application Installing Twitter application Setting up of Twitter application Posting with twitter application Adding a contact Opening a book Highlighting text in a book.	None
Setting up of Facebook application	User's email address