ANDROID LAB EXERCISE

Material Produced in the framework of:

DECAMP PROJECT

University of Padova
February 20, 2016
Contents

1 Activity 1: SSL MITM Attack using SSL Strip ............................................. 1
   1.1 Purpose ................................................................................................. 1
   1.2 Disclaimer ........................................................................................... 1
   1.3 Background ......................................................................................... 1
   1.4 Setup ................................................................................................. 3
       1.4.1 Setting up IP Forwarding ............................................................. 4
       1.4.2 Setting the victim to use a proxy server ..................................... 4
       1.4.3 Setting up port redirection using Iptables ................................. 6
       1.4.4 Starting the SSLstrip tool ......................................................... 7
       1.4.5 Dumping SSLstrip Log ............................................................... 8

2 Activity 2: Mobile Network Exploit .............................................................. 10
   2.1 Purpose ............................................................................................... 10
   2.2 Disclaimer ........................................................................................... 10
   2.3 Mobile Network Exploit Attack via Interceptor-NG ........................... 10
       2.3.1 Setup .......................................................................................... 11
   2.4 Mobile Network Exploit Attack via Droidsheep ............................... 22
       2.4.1 Setup .......................................................................................... 22
   2.5 Protection for the Network Exploits via NOROOT FIREWALL .......... 30
       2.5.1 Setup .......................................................................................... 32

3 Reference ......................................................................................................... 46
1 Activity 1: SSL MITM Attack using SSL Strip

What You Need for This Experiment

A PC or Laptop with any OS (preferable Ubuntu 14.04), Kali-Linux, Genymotion Virtual Machine, Virtualbox installed and high-speed Internet connection.

1.1 Purpose

To convert secure HTTPS sessions to insecure HTTP ones.

1.2 Disclaimer

All data and information provided on this experiment are for informational purposes only. The program described in this lab might be unethical and is considered a form of hacking. The University of Padua and the author of this experiment have no responsibility for any sort of consequence faced in the court of law for any sort of misuse.

1.3 Background

Brief Introduction: HTTP and HTTPS

HTTP and HTTPS are the application-layer protocols in TCP/IP model. HTTP stands for Hypertext Transfer Protocol. HTTPS uses a secure tunnel to transfer and receive data. This secure tunnel is commonly called as SSL (Secure Socket Layer) and therefore the suffix “S” is added to HTTPS.

http

<table>
<thead>
<tr>
<th>HTTP</th>
<th>TCP</th>
<th>IP</th>
</tr>
</thead>
</table>

https

| HTTP | SSL or TLS | TCP | IP |

Figure 1: HTTP vs. HTTPS.

What is SSLstrip?

In short SSLstrip is a type of Man-In-The-Middle attack that forces a victim into communicating with an adversary in plain-text over HTTP, and the adversary modifies content from an HTTPS server. To do this, SSLstrip is “stripping” https:// URLs and turning them into http:// URLs.
SSLstrip is used for sniff or steal password from target. SSLstrip is known in hijacking http traffic in the network.

SSLstrip was designed to hijack the SSL session and replace the SSL encryption with plain text. This is done by monitoring all connection between the host system and the server. SSLStrip then replaces https:// request with http:// requests. The browser is faked into thinking it is still directly communicating directly with the server.

![Diagram of SSL Connection Intercepted by SSLStrip](image1)

**Figure 2:** A SSL Connection Intercepted by SSLStrip.

In the SSLstrip attack, the data is intercepted by using a MitM attack and the secure requests to the server are replaced with unsecured requests. Data that is then transmitted and can be viewed in plain text.

![Diagram of Normal SSL Connection](image2)

**Figure 3:** A Normal SSL Connection.
1.4 Setup

In this experiment, we are going to perform SSLstrip attack. We use Kali Virtual Machine as an attacker and Genymption Virtual Machine as a victim. Before we actually start hacking using SSLstrip, we have to setup the MITM attack and packet redirection/forwarding mechanism. In the following, the required steps to take will be described.

First of all, open the browser in Android Genymotion VM and navigate to arbitrary HTTPS website (e.g., a bank website).

Notice that the page is using HTTPS, as shown in the following figure.
Let’s start SSLstrip attack.

1.4.1 Setting up IP Forwarding

Open up a Terminal in Kali Linux VM and type:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Or you can do this by typing these commands:

```
cp /etc/sysctl.conf /etc/sysctl.conf.bak
nano /etc/sysctl.conf
```

Then in Uncomment the this line on order that IP forwarding takes effect.

```
net.ipv4.ip_forward=1
```

Save the changes and exit.

1.4.2 Setting the victim to use a proxy server

In a real attack, we would redirect traffic by ARP poisoning (ARP cache poisoning).

**In ARP spoofing:** we constantly send the victim computer ARP answers telling him that the MAC address belonging to the IP of the gateway machine (router) is our MAC address. After some time the victim computer will believe us and makes a wrong entry in his ARP cache. Next time the victim wants to send an IP packet to the gateway he sends the Ethernet frame to our MAC address so actually we get the IP packet. We do the same thing with the gateway machine just the other way round.

Before doing ARP spoofing, Do not forget to enable IP forwarding on your host so that the traffic goes through your host. Otherwise victim will loose connectivity.
**Note:** in this experiment, we set the proxy within Genymotion VM. to make the experiment easier to do. In this case, it will not affect other machines in the Lab.

Open up a Terminal in Kali Linux and run this command:
`ifconfig`

![Figure 8: Finding IP address of Kali Linux.](image)

Take a note of your Kali Linux IP address.

In Genymotion VM, go to Setting then WiFi and long press the available WiredSSID to configure a proxy.

![Figure 9: Setting the victim to use a proxy server(1).](image)
1.4.3 Setting up port redirection using Iptables

First, remove all iptables rules by opening a Terminal and typing this command:

```
iptables -t nat -F
```

Then, check the iptables:

```
iptables -t nat -L
```
To do port forwarding execute this command:

```
iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-ports 8080
```

Open up a Terminal in Kali Linux and type in:

```
sslstrip -l 8080
```

This starts the SSLstrip tool and makes it listen to port 8080.

1.4.4 Starting the SSLstrip tool

Open up a Terminal in Kali Linux and type in:

```
sslstrip -l 8080
```

This starts the SSLstrip tool and makes it listen to port 8080.
Do not close this Terminal.

**Note:** by default Kali Linux has SSLstrip installed, but if you have no SSLstrip on the Attacker Linux machine, in a Terminal type these commands, pressing Enter after each one:

```
curl http://www.thoughtcrime.org/software/sslstrip/sslstrip-0.9.tar.gz > sslstrip-0.9.tar.gz
tar xzf sslstrip-0.9.tar.gz
cd sslstrip-0.9
```

Then, on the Attacker Linux machine, in a Terminal type this commands:
```
python sslstrip.py -h
```
A help message appears, showing the options.
Open up a Terminal and type in:
```
python sslstrip.py -p -l 8080
```
This starts sslstrip collecting data. It writes the data into a file named `sslstrip.log`.

### 1.4.5 Dumping SSLstrip Log

Once the setup is up and running perfectly, all of our victim’s traffic will be routed through the attacking machine. In particular, HTTP traffic will be redirected to the port 8080, where SSLstrip is listening. After this you will be able to eavesdrop and steal all of the victim’s passwords sent supposedly over “SSL”.

Let’s start the Logfile Scanner:
```
tail -f sslstrip.log
```
Do not close this Terminal.

Now, open up the browser in Android Genymotion VM and navigate to the same HTTPS website you already were. SURPRISE, the website is not HTTPS any more!

Insert the fake login credential in login page:

The login credential was sniffed. The username and password are visible to the attacker!!!
Activity 2: Mobile Network Exploit

In this activity, we will be doing a study of mobile network exploits. This will be done using the android applications such as Intercepter-NG and Droidsheep.

What You Need for This Experiment
An Android-powered smartphone and Internet connection.

2.1 Purpose
To show some Mobile Network Exploit such as Session Hijacking and Password Sniffing using Android apps in real-world scenarios.

2.2 Disclaimer
All data and information provided on this experiment are for informational purposes only. The program described in this lab might be unethical and is considered a form of hacking. The University of Padua and the author of this experiment have no responsibility for any sort of consequence faced in the court of law for any sort of misuse.

2.3 Mobile Network Exploit Attack via Intercepter-NG
Intercepter-NG is a multifunctional network toolkit for various types of IT specialists. Some of its features are:
- Network Traffic analysis
- Sniffing Passwords (Non-SSL HTTP traffic)
- Session Hijacking and etc.

Note: Runs on Android >= 2.3.3 with root+busybox
This app will only work on rooted devices.

Interceptor-NG (A packet sniffer Android app) enables users to grab cookies from other user(s) by mounting itself onto a local Wi-Fi network.

How it works:
In order to start, the phone must be rooted so that Intercepter-NG has superuser access. When the spoofing function is started, Intercepter-NG will obtain an IP address from the current Wi-Fi router, and will then proceed to obtain cookies that are sent through that Wi-Fi router. The user of this mobile device can then choose to view webpage data that was embedded in a cookie sent through the Wi-Fi.

Since public Wi-Fi is often unrestricted and easy to access, the potential for a mobile network attack is relatively high. Thus, it is crucial to be informed about how this attack could take place, and more importantly, how it can be prevented.
2.3.1 Setup

First, go to the Play Store and install “Interceptor-NG” as shown below:

![Interceptor-NG (ROOT)](image)

Figure 18: Interceptor-NG (1).
Run the application (In case of need, grant the required root permission to this app). The first screen will let you scan for all clients on the network and let you select the users whose traffic you want to sniff. The first result is usually the WiFi router, the results below that are connected users (known as targets or victims) on the network.

Figure 19: Intercepter-NG (2).
Select the victim and continue.

Figure 20: Interceptor-NG (3).
Then head over to the settings tab which is the “cog” symbol. There are a number of options here but you can just use the settings shown in the picture to get started.

![Figure 21: Interceptor-NG (4).](image)
Then click on the “radiation” symbol and hit the play button to start ARP poisoning and capturing data sent over the network. You will see some information here but not everything.

Figure 22: Interceptor-NG (5).
Cookies tab. This feature is really interesting as you can see which websites have been visited.

Figure 23: Interceptor-NG (6).
Here, there some cookies are sent through the Network and captured by Intercepter-NG.

Figure 24: Intercepter-NG (7).
If you click on one, you can actually view and navigate the webpage the user is viewing. Here you can see the website who user has visited.

![Intercepter-NG](image)

Figure 25: Intercepter-NG (8).
You might want to hijack more sessions :)

Figure 26: Intercepter-NG (9).
Back to the radiation symbol tab, you can also see what the users are doing on the network and if they visit a site you can sniff usernames and passwords. Here, you can see a user visited this website: www.lrb.co.uk/login.

![Image of a web browser showing www.lrb.co.uk](image)

This site uses cookies. By continuing to browse this site you are agreeing to our use of cookies.

**London Review of Books**

JAN-WERNER MÜLLER

**The Problems of the Eurozone**

We can expect more Greek drama before too long: the real struggle over the Eurozone – and the EU more broadly – is just beginning.

AVIES PLATT

**An Encounter with Yeats**

One evening in the spring of 1937 I was in London, at the Grafton Galleries. The occasion was an open meeting of the Sex Education Society.

OWEN BENNETT-JONES

**Suburban Jihadis**

The fact that more British Muslims are fighting for Islamic State than for the British army demands an explanation.

STEPHEN SEDLEY

**The Right to Die**

Figure 27: Interception-NG (10).
The victim has logged in with his login credential.

**Figure 28: Interceptor-NG (11).**
Now, you can see the username and password of the user in the last two lines.

![Intercepter-NG](image)

**Figure 29: Intercepter-NG (12).**

**Note:** this will work on any site that only uses “HTTP”. So always be careful when using free or public WiFi.

### 2.4 Mobile Network Exploit Attack via DroidSheep

DroidSheep is a simple Android tool for web session hijacking (side-jacking). It listens for HTTP packets sent via a wireless network connection and extracts the session id from these packets in order to reuse them.

DroidSheep takes a number of steps to accomplish this task.

- The phone must be rooted so that DroidSheep has superuser access.
- DroidSheep obtains an IP address from the current Wi-Fi router
- DroidSheep proceeds to obtain cookies that are sent through that Wi-Fi router.

As cookies are sent through the Wi-Fi router, a list of data will appear on the mobile device. The attacker can then choose to view webpage data that was embedded in a cookie sent through the Wi-Fi.

#### 2.4.1 Setup

Since DroidSheep accomplishes tasks that are categorized as hacking, the application cannot be found on the Android Market.
First, download the Droidsheep app from the below link:
https://github.com/jgworks/droidsheep/tree/master/%20droidsheep
Install the app on your device.

Figure 30: Droidsheep (1).
Figure 31: Droidsheep (2).
Fire up the Droidsheep app on your device and check mark the disclaimer.

Figure 32: Droidsheep (3).
Connect to any Wi-Fi Network.

Figure 33: Droidsheep (4).
Check mark the “ARP Spoofing” and “generic mode” at the bottom of the screen to scan for all cookies passing through the Wi-Fi. Lastly, push the “start” button to start the spoofing activity.

Figure 34: Droidsheep (5).
When cookies are passed through the Wi-Fi, Droidsheep captures them and shows them as a list. The list displays the name of the site and the ID number.

Figure 35: Droidsheep (6).
Select a specific cookie. When a cookie is clicked, a menu of options pops up, enabling the user to perform multiple functions with that cookie. In order to perform Session Hijacking attack, select the “Open site”.

Figure 36: Droidsheep (7).
Now, you hijacked the session belonging to another user!

2.5 Protection for the Network Exploits via NOROOT FIREWALL

Most of the apps on Android devices probably require a data connection, and the user may not even know when they are sending or receiving data. NoRoot Firewall puts the user in
control of Internet access for apps, as well as how they can access it. This way, the user can choose which traffic he wants to allow and which he wants to stop.

With Firewall applications we can keep a tab on the external IPs that an app connects to in real-time.

NoRoot Firewall app:

- Lets the user control outgoing network connections on his Android-powered smartphone.
- Can be installed on a non-rooted device (none of the system files are modified by this app).

When any app tries to access the Internet, the user will receive an alert where he can allow or deny it and save the setting for the future. This level of control can also let the user see when a certain app is accessing data for ads and block it.

The user can specify if an app can access the Internet only over Wi-Fi, only over 3G/Data, or both.

**Note:** for 4G users though as NoRoot Firewall does not work well with 4G at the moment.
2.5.1 Setup

Install NoRoot Firewall app from Play Store and run the app.
Once the app initializes, tap on the Start button to start the firewall service on your device.

Figure 39: NoRoot Firewall (2).
The app will ask you to grant VPN access and you will have to tap the OK button when it asks you the permission to connect to the VPN.

Figure 40: NoRoot Firewall (3).
No actual VPN connection is made. The VPN connection is actually between the NoRoot Firewall app and the device, then Firewall app decides whether or not requesting app should be allowed to connect to the Internet.

Figure 41: NoRoot Firewall (4).
Under “APPS” tab, there exist list of all application installed on the device.

<table>
<thead>
<tr>
<th>PENDING ACCESS</th>
<th>APPS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![NoRoot Firewall screenshot](image)

**Figure 42: NoRoot Firewall (5).**
When an app (e.g., Chrome) tries to connect to the Internet, NoRoot Firewall gives a notification message stating that this app is trying to connect to the Internet. The user can then decide to grant or deny the request.

Figure 43: NoRoot Firewall (6).
Once the user grants or denies the request for an app to go online, NoRoot Firewall will save the settings in its configuration file and the user can review them in the "APPS" tab.

![NoRoot Firewall](image)

<table>
<thead>
<tr>
<th>PENDING ACCESS</th>
<th>APPS</th>
<th>GLOBAL FILTERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android System, Fused Location, Google Backup Transport, Input Devices, Key Chain, Settings Storage, Settings</td>
<td>![Checkbox] ![Checkbox]</td>
<td>![Search Bar]</td>
</tr>
<tr>
<td>Barcode Scanner</td>
<td>![Checkbox] ![Checkbox]</td>
<td>![Search Bar]</td>
</tr>
<tr>
<td>Bluetooth Share</td>
<td>![Checkbox] ![Checkbox]</td>
<td>![Search Bar]</td>
</tr>
<tr>
<td>Browser</td>
<td>![Checkbox] ![Checkbox]</td>
<td>![Search Bar]</td>
</tr>
<tr>
<td>Calendar</td>
<td>![Checkbox] ![Checkbox]</td>
<td>![Search Bar]</td>
</tr>
<tr>
<td>Calendar Storage</td>
<td>![Checkbox] ![Checkbox]</td>
<td>![Search Bar]</td>
</tr>
<tr>
<td>Chrome</td>
<td>![Checkbox] ![Checkbox]</td>
<td>![Search Bar]</td>
</tr>
<tr>
<td>com.advanced.rootchecker</td>
<td>![Checkbox] ![Checkbox]</td>
<td>![Search Bar]</td>
</tr>
<tr>
<td>ConfigUpdater</td>
<td>![Checkbox] ![Checkbox]</td>
<td>![Search Bar]</td>
</tr>
<tr>
<td>Contacts Storage, Contacts, Search Applications Provider, User Dictionary</td>
<td>![Checkbox] ![Checkbox]</td>
<td>![Search Bar]</td>
</tr>
</tbody>
</table>

Figure 44: NoRoot Firewall (7).
In case of denying the app request:

![NoRoot Firewall](image)

Figure 45: NoRoot Firewall (8).
All the pending requests of the apps will be saved under the “Pending Access” tab and the user can review them there.

![NoRoot Firewall](image)

**Figure 46: NoRoot Firewall (9).**
Once a app is granted the permission, it is granted for both Wi-Fi and cellular data. If the user wants to change it to only Wi-Fi or cellular data for some reasons, it can be done under the “APPS” tab. Just the user must uncheck the option he would like to revoke the access from, and save the settings.

Figure 47: NoRoot Firewall (10).
Also, it is possible to create global filters for all the apps that are installed on the user device. If there is any specific IP and Port the user would like to block for all the apps installed on his device, he should tap the option “New Pre-Filter” under the “Global Filter” tab and enter the IP followed by the port information.

Figure 48: NoRoot Firewall (11).
Adding filters can also be done for individual apps and the user can configure it by tapping the app name under the "APPS" tab.

Figure 49: NoRoot Firewall (12).
Figure 50: NoRoot Firewall (13).
Also, the user can see the activities log under “ACCESS LOG” tab.

![NoRoot Firewall](image)

**Figure 51: NoRoot Firewall (14).**

Using NoRoot Firewall, each app can be controlled to only connect to Wi-Fi, 3G, or none. One potential use of this app is to prevent network exploits (through Wi-Fi) by configuring it so that all the applications in the Android phone cannot access Wi-Fi, but only 3G. **This allows for a more secure and less exploitable Internet.**
3 Reference