LAB 2

ANDROID LAB EXERCISE

MATERIAL PRODUCED IN THE FRAMEWORK OF:

DECAMP PROJECT

University of Padova
January 10, 2016
Contents

1 Activity 1: Sniffing Insecure Connections
   1.1 Purpose ........................................... 1
   1.2 Setup ............................................ 1
      1.2.1 Installing the Client App ................... 1
      1.2.2 Installing the Server Component .......... 2
      1.2.3 Logging In .................................. 3
      1.2.4 Stopping the Server App ................... 4
   1.3 Running and Configuring Burp to Redirect Requests .......... 5
      1.3.1 Starting the Server on Port 8081 .......... 6
      1.3.2 Viewing Accounts from the Android Phone ........ 7
      1.3.3 Viewing Traffic in Burp .................... 8
      1.3.4 Starting the Server on HTTPS Port ........ 9
   1.4 Parameter Manipulation ................................ 12
      1.4.1 Performing a Normal Transfer ............... 13
      1.4.2 Stealing Money ................................ 21
   1.5 Countermeasure .................................. 24
   1.6 Source/Reference .................................. 24

2 Activity 2: SSL Man In The Middle Attack - Stealing Credentials
   2.1 Background ...................................... 25
   2.2 Setup ............................................ 26
   2.3 Run and Configure Burp Proxy .................... 26
   2.4 Uninstalling the PortSwigger Certificate ........... 31
   2.5 MITM Attack .................................... 33
      2.5.1 Installing a Vulnerable App ................. 33
      2.5.2 Countermeasure .............................. 37
      2.5.3 Further Study ................................ 37
   2.6 Source/Reference .................................. 38
1 Activity 1: Sniffing Insecure Connections

What You Need for This Experiment
A computer with Android-Studio running, along with the Nexus 5 emulator, and Burp proxy.

1.1 Purpose
To intercept password transmission from the vulnerable app and observe its insecurity.

1.2 Set up
Launch Android Studio and the Nexus 5 emulator.
Swipe up to get to the home screen of the emulated phone.
We will pipe communication between the client and server through a proxy (in reality could be a proxy server to share the Internet or public WiFi access point) and use this to manipulate the data sent.

1.2.1 Installing the Client App
In a Web browser, go to https://github.com/securitycompass/AndroidLabs On the right side, click “Download ZIP”.
Unzip the file you just downloaded.
A folder named AndroidLabs-master appears.
In Android Studio, click File, “Import Project”.
Navigate to the AndroidLabs-master folder.
When the project loads, click Run, “Run app”.
Select the Nexus 5 emulator.
You see a bank login screen, as shown in the following figure:
1.2.2 Installing the Server Component

In a browser, go to https://github.com/securitycompass/LabServer
On the right side, click “Download ZIP”.
Find the LabServer-master.zip file and unzip it.
A LabServer-master folder appears.
Find the complete path to this folder.
Open up an Ubuntu terminal and type in:
cd /home/hossein/Downloads/LabServer-master
python app.py

You should see the message: “Serving HTTP on port 8080”, as shown in the following figure:
Leave this window open.

1.2.3 Logging In

In the Nexus 5 emulator, enter a username of “jdoe” and a password of “password”. Click Done.

You see the message “Please configure a local password”, as shown below:
Enter a password of “P@ssw0rd” in both fields and click Done. You now see the main app page, as shown below:

![Figure 4: Log in in the client app (2)](image)

1.2.4 Stopping the Server App

Click in the Terminal running app.py. Press “Ctrl+C” or “Ctrl+Z” to stop the server and close the terminal.

![Figure 5: Log in in the client app (3)](image)
1.3 Running and Configuring Burp to Redirect Requests

Open up an Ubuntu terminal and go to home directory of Burp proxy. Type:
```
java -jar burpsuite_free_v1.6.01.jar
```
Click the Proxy tab. Go to Options sub-tab. Set IP address to listen to 127.0.0.1 and port number 8080. Burp will listen on 127.0.0.1:8080, as shown in the following:

![Burp configuration (1)](image1)

The emulated Android phone is also using this address and port number to reach the server. In Burp, click the Edit button. In the “Edit” button in proxy listeners box, click the “Request handling” tab.

![Burp configuration (2)](image2)
Insert host and port number as below:
- Redirect to host: 127.0.0.1
- Redirect to port: 8081
Click OK. Burp now shows an Interface of 127.0.0.1:8080 and a Redirect of 127.0.0.1:8081, as shown below:

![Figure 8: Burp configuration (3)](image)

1.3.1 Starting the Server on Port 8081

Open up an Ubuntu terminal and type:

```
python app.py -- port 8081
```
1.3.2 Viewing Accounts from the Android Phone

On the emulated Android phone, open the Accounts item. You will be asked for your password. Enter “Pssw0rd”. You should see your account balances, as shown in the following figure:
1.3.3 Viewing Traffic in Burp

In Burp, on the Proxy tab, click the “HTTP history” tab.
Click a “POST /login” line.
The lower pane shows the data the banking app sent to the server. The username of “jdoe” and password of “password” are visible in clear text, as shown in the following figure:
1.3.4 Starting the Server on HTTPS Port

To run the HTTPS server on port 8443, open up an Ubuntu terminal cd to “LabServer” home directory and type in:

```
python app.py - - ssl - - port 8443
```

To activate HTTPS connection, first click on the most right-hand side button. It seems like “thee dots over each other”. 
Then click on the "Preference" and enable HTTPS mode.
Now, go to the Burp proxy (Proxy tab, options sub-tab) and do required setting and edit proxy listener. In Burp, click the Edit button. In the “Edit” button in proxy listeners box, click the “Request handling” tab. Insert host and port number as below:
-Redirect to host: 127.0.0.1
-Redirect to port: 8443
Check mark the “Force use of SSL”
Click OK. Burp now shows an Interface of 127.0.0.1:8080 and a Redirect of 127.0.0.1:8443, as shown below:

![Edit proxy listener](image)

Figure 16: Starting connection on HTTPS port (3).

![Edit proxy listener](image)

Figure 17: Starting connection on HTTPS port (4).

Make sure that "Intercept" is ON in the Burp proxy. Now, if you do a normal transfer using the app and server, you will not sniff any message via Burp proxy due to the fact that you have used a HTTPS connection that can not be intercepted!
1.4 Parameter Manipulation

In Burp, on the Proxy tab, click the Intercept sub-tab. Click the “Intercept is off” button. Now the button says “Intercept is on” as shown below:

Figure 18: Starting connection on HTTPS port (5).

Figure 19: Burp proxy, Intercept sub-tab (1)

Figure 20: Burp proxy, Intercept sub-tab (2)
1.4.1 Performing a Normal Transfer

In Burp, on the Proxy tab, click the Options sub-tab. Scroll down to the “Intercept Client Requests” section. Check the “Intercept responses based on the following rules” box. In the list of rules, check the item “HTTP method (get | POST)”, as shown below:

Log in the mobile banking app using credential already provided. Click “Done” button. Now in Burp, you can see the user credential sent over the network.
Figure 23: Burp proxy, Intercept client Requests (2)

Request to http://127.0.0.1:8081

- Forward
- Drop
- Intercept is on

Raw | Params | Headers | Hex
---|---|---|---

POST /login HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Dalvik/2.1.0 (Linux: U: Android Host: 10.0.2.2:8080
Connection: Keep-Alive
Accept-Encoding: gzip
Content-Length: 31

username=jdoe&password=password

Figure 24: Burp proxy, Intercept client Requests (3)
In Burp click on the “Forward” button to allow this request pass through. Again push “Forward” button to pass another intercepted request as shown in the following figures.
In the emulator, click the Transfer button.
In Burp, click the Forward button to allow this request to pass through.
Another request is intercepted within a few seconds.
Click the Forward button again. The emulator shows a “Transfer complete” message.

In the transfer screen, insert the amount of 50, to transfer money from Debit to Credit account.
Click “Transfer”.

Then let the intercepted request pass through.

Figure 29: Burp proxy, Intercept client Requests (8)

Figure 30: Burp proxy, Intercept client Requests (9)
In Burp, click the Intercept sub-tab. You see the intercepted request. In the center of the Burp window, click the Params tab. You see the parameters of the request, including “amount”, “to_account”, and “from_account”, as shown in the following figure:
In the emulator, return to the Transfers screen. In Burp, forward the traffic, as required. The balances have changed - - Debit has gone down by $50 and Credit has gone up, as shown in the following figure:
Figure 35: Check the user balance
1.4.2 Stealing Money

In the emulator, take note of your Debit and credit cards, in the Transfer screen, enter an amount of 90 and click the Transfer button.

![Figure 36: Stealing Money (1)](image)

Burp intercepts the POST request.
In Burp, on the “to_account” line, double-click the account number of 987654321.

![Figure 37: Stealing Money (2)](image)
Change that number to “111111111” (someone else account number), and press Enter, as shown below:

![Figure 38: Stealing Money (3)](image)

In Burp, click the Forward button to allow this request to pass through. Another request is intercepted within a few seconds.

![Figure 39: Stealing Money (4)](image)
Click the Forward button again.
The emulator shows a “Transfer complete” message. In the emulator, return to the Transfers screen.

Figure 41: Stealing Money (6)

In Burp, forward the traffic, as required. Now check Credit balance, apparently the amount of credit balance was not increased!

You can also transfer money from somebody else’s account to your account just by changing “from_account to 111111111”.

**Note:** This account 111111111 should not be empty!
1.5 Countermeasure

The countermeasure here is the same as it would be in a regular web app, we have to perform some validation on the server, as shown below:

Add this chunk of code after line 114 in “app.py” file.

```python
//validate that amount is positive
if total_cents < 0:
    return error('E5')

//validate that accounts belong to user:
if to_account.user != session.user or from_account.user != session.user:
    return error('E6')
```

1.6 Source/Reference

[3] https://samsclass.info/
2 Activity 2: SSL Man In The Middle Attack - Stealing Credentials

What You Need for This Experiment
A Genymotion Virtual Machine with the Google Play Store running through the Burp proxy.

2.1 Background
This experiment demonstrates the poor security of an Android app using fake credentials in a controlled virtual environment. Some Android apps do not verify SSL certificates. Normal users will not notice any problem, but they are left open to MITM attacks. This experiment performs a very simple MITM attack using the Burp proxy and an untrusted certificate. In this activity, we perform a MITM attack performed by a proxy server exploiting app vulnerability.

2.1.1 Why People Use Proxy Server?
A proxy server is a computer that acts as an intermediary between the user’s computer and the Internet. It allows client computers to make indirect network connections to other network services. If use proxy server, client computers will first connect to the proxy server, requesting some resources like Web pages, Games, Videos, MP3, E-books, any other resources which are available from various servers over Internet.

![Proxy Server Diagram]

As soon as getting such request, the proxy server will seek for the resources from the cache in its local hard disk. If the resources have been cached before, the proxy server will return them to the client computers. If not cached, it will connect to the relevant servers and request the resources on behalf of the client computers. Then it “caches” resources from the remote...
servers, and returns subsequent requests for the same content directly. Nowadays, we use proxy server for various purpose like sharing Internet connections on a local area network, hide our IP address, implement Internet access control, access blocked websites and so on. Bellow are some benefits why people use proxy server:

- To share Internet connection on a LAN.
- To speed up Internet surfing.
- To hide the IP address of the client computer so that it can surf anonymous, this is mostly for security reasons.
- To bypass security restrictions and filters.

2.2 Set up
Open up an Ubuntu terminal and navigate to home directory of Genymotion Virtual Machine and type in:
```
cd /home/hossein/Downloads/androidLAB/genymotion
./genymotion
```
Click on a virtual device and Click “Start”.

2.3 Run and Configure Burp Proxy
Open up another Ubuntu terminal and navigate to home directory of Burp Proxy and type in:
```
cd /home/hossein/Downloads/androidLAB
java -jar burpsuite_free_v1.6.01.jar
```
In Burp Proxy tab, “options” sub-tab define the interface that Burp is listening to it, as shown below:

![Configure Burp Proxy](image.png)

Figure 43: Configure Burp Proxy (1)
Add a new listener and insert the following data:
- Bind to port: 8080
- Specific address: Your local IP address

**Note:** to obtain local IP address, open up an Ubuntu terminal and type: `ifconfig`
Take note of your IP address.
Click on Settings, hit Wi-Fi, push WiredSSID for a few seconds and then modify network. To find IP address open up an Ubuntu terminal and type in “ifconfig”. Select Manual Proxy and set IP address of your host computer there then Save it as shown in the following figure:
In your Genymotion device, open a Web browser and go to https://google.com. This page should load without any error messages. If you see errors saying the certificate is untrusted, install the “PortSwigger” certificate as explained in “LAB 1”. In Burp, on the Proxy tab, on the “HTTP history” sub-tab, you should see: https://google.com.
This is normal HTTPS interception by a trusted proxy, and does not demonstrate any security problem.

2.4 Uninstalling the PortSwigger Certificate

In Genymotion, click Home, Circle, Settings, Security, “Trusted credentials”, USER. The PortSwigger certificate appears, as shown in the following figure:

Figure 52: Uninstalling the PortSwigger Certificate (1)
Click PortSwigger. A “Security certificate’ page opens. Scroll to the bottom and click the Remove button, as shown in the following figure. Click OK.

![Figure 53: Uninstalling the PortSwigger Certificate (2)](image)

Click Home. Open the Web browser. In the URL bar, retype `https://google.com` and press Enter. A “Security warning” box appears, as shown in the following figure:

![Figure 54: Security Warning](image)
This indicates that the browser is properly implementing SSL security and detecting the MITM attack that Burp is performing.

2.5 MITM Attack

In a MITM attack, the attacker is in a position to intercept messages sent between communication partners. In a passive MITM, the attacker can only eavesdrop on the communication, and in an active MITMA, the attacker can also tamper with the communication.

MITMA attack against mobile devices are somewhat easier to execute than against traditional desktop computers, since the use of mobile devices frequently occurs in changing and untrusted environments.

The Secure Sockets Layer (SSL) and its successor, Transport Layer Security (TLS), are cryptographic protocols that were introduced to protect network communication from eavesdropping and tampering. SSL is fundamentally capable of preventing both passive and active Man In The Middle attacks.

2.5.1 Installing a Vulnerable App

There are some Android applications failing validate SSL certificate properly. In the following you can see some of these apps. Android apps that fail to validate SSL:

https://docs.google.com/spreadsheets/d/1t5GXwjw82SyunALVJb2w0zi3FoLRIkGPc7AMjRF0r4/edit#gid=123856677).

In your Genymotion Android device, open Google Play.

Install “Instarepost” as shown in the following figure:

![Figure 55: Installing Instarepost app (1)](image)
In the Genymotion device, start the “Instarepost” app and Log in with the below fake credential. Enter these fake credentials:
Email: my_name@gmail.com
Password: my-sercet-password
Figure 57: Log in Instarepost app

Figure 58: Insert Login Credential (1)
The credentials are rejected, which is OK – what matters for us is how they were transmitted to the server. In Burp, examine the HTTP History. You should find your username and password. In Burp, examine the HTTP History. You should find a POST request to a `https://instagram.com` server containing your username and password, as shown below:
Note: this is VERY WRONG. Burp is intercepting and opening HTTPS traffic with an untrusted certificate, and the app is ignoring it. SSL certificate is no longer validated!!!
2.6 Source/Reference

[1] https://samsclass.info/