Exercises Day 01 – Basics of Web Security

Experiential Learning Workshop

1 General Guidelines

1. Make a team of two unless stated otherwise.
2. For each exercise, use wireshark capture to verify contents
3. Ensure to use proper capture filter and don't capture irrelevant traffic
4. Where appropriate or applicable, use **wget** or **nc** to access the web server.
5. The default client for accessing web server is assumed to be browser, preferably **firefox**. You can use Chrome or any other browse as well.
6. The webserver in the example below is taken as 'myweb.com'. Please use your hostname or corresponding IP address instead in your exercise.
7. To kill any program in the **Linux** terminal, please press **Ctrl-C** and not **Ctrl-Z**. The latter will suspend the program and not stop it.

2 Hands-on 1: Installing SSL Certificate and Mixed Content

Note: It is assumed that Apache web server has been installed on one of the machine and let us call it web server. In case, it is not, please install it by invoking the command

```
sudo apt install apache2.
```

2.1 Invalid SSL certificate.

In the exercises below, we will work with SSL Certificate that are not compatible with browser expectation.

2.1.1 Accessing websites with invalid certificate.

1. Ping www.google.com and find its IP address (ping -c2 www.google.com). Let this IP address be `a.b.c.d`
2. Open firefox (or chrome) browser and access https://a.b.c.d.
3. The browser should display a web page with warning rather than displaying the google search page.
4. Analyze the warning. Click thru this warning and browser should display google search page.
5. Make an entry for this IP address a.b.c.d in your **/etc/hosts** file like below. Use editor of your choice to make an entry (e.g. nano, edit, gedit, vi etc.)
   a. a.b.c.d mygoogle.com
6. In the browser type the URL, https://mygoogle.com
7. Analyze the web page response

2.1.2 Running your own web site with HTTPS

1. Generate and deploy a web site SSL certificate e.g. for (`mywww.com`) on Apache web server. The steps are given Appendix.
2. Create an entry in the **/etc/hosts** file of the client machine (creating this entry on the web server doesn't really matter). The entry should be something like as below where 10.1.1.101 is server's IP Address
   a. `10.1.1.101 mywww.com`

3. In browser of client machine, enter https://mywww.com.
4. Analyze the web page response. It should be warning again, but this warning should have different indicator compared to when accessed mygoogle.com.
5. Analyze the warning web page.
6. Import the certificate into browser repository. Click thru and then browser should display the Apache welcome page. Import
7. Close the browser, reopen it again and access again the web page https://mywww.com. This time no warning should be displayed since browser knows the certificate authority.

2.1.3 Exercise Expectations

1. Understand SSL certificate errors in case of any mismatch of information.

2.2 Mixed Content.

In the exercises below, we will work with web pages that has embedded objects with both HTTP and HTTPS.

1. Create 3 web pages, namely pure.html, mixed.html and mixed-active.html as below. These 3 web pages are available at rprustagi.com/accs/pure.html, rprustagi.com/accs/mixed.html, and rprustagi.com/accs/mixed-active.html as well and download using wget.

a. pure.html

```
<html>
<head>
<title>Pure Content </title>
<body>
<h1>Pure Content Demonstration</h2>
<h2>Image 01 with inherited security access.</h2>
<h2>Image 02 with inherited security access.</h2>
<img src="//mywww.com/img/img-01.jpg" alt="Img 01">
<img src="//mywww.com/img/img-02.jpg" alt="Img 02">
</body>
</html>
```

b. mixed.html

```
<html>
<head>
<title>Mixed Content </title>
<body>
<h1>Mixed Content Demonstration</h2>
<h2>Image 01 with inherited security access.</h2>
<h2>Image 02 with insecure access.</h2>
<img src="//mywww.com/img/img-01.jpg" alt="Img 01">
<img src="http://mywww.com/img/img-02.jpg" alt="Img
02">
</body>
</html>
```

c. mixed.html

```
<html>
<head>
```

```
<title>Mixed Active Content </title>
</head>
<body>
<script src="http://rprustagi.com/js/mywww.js">
</script>
<h1>Mixed Content Demonstration</h2>
  <button type="button" onclick="hello()"> insecure
access
  </button>
<h2>Image 02 with insecure security access.</h2>
<img src="//rprustagi.com/img/img-02.jpg" alt="Img
01">

</body>
</html>
```

2. Deploy these web page in your Apache web server. Copy these contents to directory `/var/www/html` on Apache2 web server. Create the directory `/var/www/html/img` and copy some images with the name `img-01.jpg` and `img-02.jpg` in the images directory.
3. Access the web page pure.html using http and https://mywww.com/pure.html/. (Alternately you can use rprsutagi.com/accs instead of myweb.com) Both the web pages should be displayed properly
4. In Firefox browser, access the web page https://mywww.com/mixed.html (alternately https://rprustagi.com/accs/mixed.html).
5. You should see a grayed lock icon with warning sign. Analyze and correct the error and see if the lock icon becomes green.
6. In the Firefox browser, access the web page https://mywww.com/mixed-active.html (alternately https://rprustagi.com/accs/mixed-active.html).
7. Browser should show gray lock icon with a cross.

2.3 Exercise Expectations

1. Understand lock icons display when web page has insecure references i.e. mixed contents.

3 Hands-on 2: ARP Spoofing and MITM

This exercise requires setup of 3 machines.

3.1 Silent snooping

1. Identify the attacker machine, say X. Identify the IP address of A and B.
2. Install **dsniff** on attacker machine (`sudo apt install dsniff`)
3. On A, ping IP_B and on B, ping IP_A and note down their MAC address in ARP Table (`arp -an`). Verify that ARP table has correct MAC addresses.
4. Run the ARP Spoof attack, i.e. issue following commands on attacker machine X. The first command will send gratuitous APR message and send will enable routing on X.

a. `sudo arpspoof -I enp1s0 -t` IP_A `-r` IP_B
b. `sudo sysctl -w net.ipv4.ip_forward=1`

5. Look at the ARP table of A and B. On A, for IP_B, it should show MAC address of X. Similarly, on B, for IP_A, it should show MAC address of B.
6. Run wireshark capture on X with capture filter as host IP_A and host IP_B.
7. Do the chat using nc between A and B. Identify chat contents of A and B on X in wireshark

3.2 Dealing with ARP spoofing

1. In the above setup, issue make the static ARP entry for B on A and vice versa e.g.
   a. (On A) `sudo arp -s` IP_B MAC_B
   b. (On B) `sudo afp -s` IP_A MAC_A
2. Explore if silent snooping by attacker can still work.

3.3 Exercise Expectations

1. Able to understand working of ARP spoofing and simplicity with which it can be done.
2. Able to understand working APR and how to avoid silent ARP spoofing and snooping attacks.

4 Appendix: Generating and installing SSL certificate

1. Follow the steps below to create a self -igned certificate with a private key for the website mywww.com (or choose your own website name) with validity period of 10 year (3650 days).

2. `sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/self.key -out /etc/ssl/certs/self.crt`

It will prompt to provide following information items:

i. Country Name
ii. State or Province Name
iii. Locality Name
iv. Organization Name
v. Organizational Unit Name
vi. Common Name
vii. Email Address.

Use the value `mywww.com` for Common name. This is the website name that will be used in the browser to access the website using HTTPS. The above command creates the required files in the directory **/etc/ssl** but can be kept anywhere as per your preference.

3. Edit the apache web server configuration file `/etc/apache2/site-enabled/default-ssl.conf` and make/update the following entries.

```
ServerName      mywww.com
SSLEngine on
SSLCertificateFile  /etc/ssl/certs/self.crt
SSLCertificateKeyFile   /etc/ssl/private/self.key
```

4. Enable the SSL changes in Apache web server.
   ```
   sudo a2enmod ssl
   sudo a2ensite default-ssl
   ```

5. Restart Apache web server.
   ```
   sudo systemctl restart apache2
   ```

← end of exercise: Basics of Web Security →