Lab 12.1: SSH Tunneling with Putty
IN618 Security

May 21, 2015

Introduction

Earlier this week we saw how easy it is to intercept and read unencypted web traffic. In this lab we will configure Apache to use TLS/SSL to encrypt our web traffic.

1 Preliminaries

Obtain the ip address of your server from the lecturer. Verify that the web site is serving our example page by visiting the web site at `http://<your-ip>/secure-login`.

Use Putty to log in to your server and enter the commands below.

2 Procedure

1. `sudo a2enmod ssl`
2. `sudo openssl req -new -newkey rsa:2048 -nodes -keyout key.pem -out req.csr`
3. `sudo openssl x509 -req -days 365 -in req.csr -signkey key.pem -out server.crt`
4. `sudo mv server.crt /etc/ssl/certs/`
5. `sudo mv key.pem /etc/ssl/private/`
6. Edit the Apache vhost configuration file at `/etc/apache2/sites-available/default-ssl.conf`. If you're unfailiar with Linux, use the command `sudo nano /etc/apache2/sites-available/default-ssl.conf`. Edit the `SSLCertificateFile` and `SSLCertificateKeyFile` entries to use the files we set up above.
7. `sudo a2ensite default-ssl`
8. `sudo service apache2 restart`

3 Viewing your site with HTTPS

Check that the procedure works by visiting `http://<your-ip>/secure-login` with your browser. Because you are using a self signed certificate you will get a warning message requiring your to accept it.

Capture a login session with Wireshark to verify that the data is properly encrypted.

4 Getting a properly signed certificate

To make your web site ready for public use, you need to get your keys signed by a recognised certificate authority. An example authority is Thawte. Look at their web site and see the options for certificates they offer. Note that this isn't a recommendation for any particular CA. We are just using Thawte as an example.