Lab 5: Basic Web Security

Due: 10pm Friday, June 6, 2014

Challenge Problems

Problem 1: HTTP Cookies

See http://192.168.14.50/lab05/prob01/ on the seclab virtual network.

Problem 2: More HTTP Cookies

See http://192.168.14.50/lab05/prob02/ on the seclab virtual network.

Problem 3: HTTP Basic Auth

See http://192.168.14.50/lab05/prob03/ on the seclab virtual network.

Login with username "bob". You know his password already.

Problem 4: Thousands of Cookies

See http://192.168.14.50/lab05/prob04/ on the seclab virtual network.

Problem 5: Bypassing Simple Javascript Protections

See http://192.168.14.50/lab05/prob05/ on the seclab virtual network.

Problem 6: Javascript Page Modifications

See rewriteme.html (rewriteme.html) for an HTML file that embeds Javascript from an external file (called rewritepage.js). Download rewriteme.html and create a rewritepage.js such that, when you view rewriteme.html in a browser, both headings and both paragraphs contain entirely different content, and the page's title is "I have been rewritten!".

Turn in your rewritepage.js in the Lab5 dropbox in D2L (https://d2l.pdx.edu/). Your script will be tested using Firefox on Linux, with the same versions as in your seclab VM.

Problem 7: Reflected XSS

See http://192.168.14.50/lab05/prob07/ on the seclab testbed.

The page is vulnerable to a reflected XSS injection. Craft a malicious URL that injects Javascript into the page and pops up an "alert" message in the browser. Your alert message should say "I successfully completed Problem 7".

Turn in your malicious URL in a plain text file called "problem7.txt" in the Lab5 dropbox in D2L (https://d2l.pdx.edu/).

Problem 8: SQL Injection

See http://192.168.14.50/lab05/prob08/ on the seclab testbed.

The page is vulnerable to SQL injection. Your task is to retrieve the password for the user "flag".

Problem 9: A Simple Web Server

Write a simple web server and listen for HTTP connections on port 80. You can either write your own server from scratch using the basic socket API's, or you can build on Python's BaseHTTPServer (https://docs.python.org/2/library/basehttpserver.html) or SimpleHTTPServer (https://docs.python.org/2/library/simplehttpserver.html) classes.

A client will connect to your VM and request the resource /stuff.txt. If you give him the contents of this file (stuff.txt) in a valid HTTP response, he will then make a second HTTP request containing the flag.