SAINT LOUIS UNIVERSITY

CENTER FOR WORKFORCE AND ORGANIZATIONAL DEVELOPMENT

SEC500

APPLICATION SECURITY

SAINT LOUIS UNIVERSITY WORKFORCE CENTER | WOOL CENTER - 2ND FL | 3545 LINDELL BLVD, ST. LOUIS, MO 63103
PHONE: 314.977.3226 | EMAIL: INFO@WORKFORCECENTER.SLU.EDU | WEB: WORKFORCECENTER.SLU.EDU

SEC500 Exercises/Labs

This page intentionally left blank

SEC500 Application Security Night 1

Exercise 1

1. What are the concerns about web security mention on page 1.
2. Page 3 mentions why web applications contain vulnerabilities. Expand upon the development of applications.
3. In addition to public Internet, web applications have been widely adopted inside _____ to support key business functions.
4. List some benefits of Web Applications.
5. What is SSL? (Page 7)
6. T/F SSL is not needed because it has nothing to do with the 6 key attacks.
7. Expand upon Cross Site Scripting.
8. Expand upon SQL Injection.
9. Expand upon Information Leakage.
10. T/F Cross-site scripting vulnerabilities are the largest percentage of incidents on web applications.
11. What is the core security problem?
12. Expand upon "The New Security Perimeter". (Pages 12 and 13)

13. See page 13 and complete this sentence: A malicious attacker can leverage a benign but vulnerable web application to attack ___ ____ ____ ____ ____**.**

SEC500 Application Security Night 1

Hands-On Exercise 1

1. Use the Internet and identify a web application vulnerability scanner.

2. Use the Internet and identify a web application vulnerability assessment tool.

3. Use the Windows 10 Virtual Machine and identify some common security tools.

   Username: student

   Password: student2013

4. Use the Linux Virtual Machine and start up the web server.

   Verify the web server software by typing in the following at a command prompt:

   ```
   [student@localhost ~]$ su –
   ```

   Enter the password at the prompt

   ```
   [root@localhost ~]# rpm -qi httpd
   ```

   Write down some specific information about the web server software below:

5. Ensure the web server is started:

   ```
   [root@localhost ~]# service httpd start
   ```

   Or

   ```
   [root@localhost ~]# service httpd status
   ```

Exercise 2

1. Complete this sentence, “The fundamental security problem with web applications – that all user input is _____.
2. List the 4 defense mechanism employed by web applications. (Page 17)
3. List some techniques that can be used to improve online authentication.
4. What is session management?
5. T/F Session management uses an Internet cookie.
6. T/F Session tokens that are sequential invite cross-site scripting.
7. Explain why one level of access control can be problematic.
8. Complete this sentence, “All user input is _____”. (Page 21).
9. Expand upon the term “Reject Known Bad”. (Page 23)
10. Expand upon the term “Accept Known Good”.
11. Expand upon Sanitization.
12. See page 28 and Expand upon multistep validation and Canonicalization?
13. Why are logs important?
14. Explain managing the application on page 35.

Hands On Exercise 2

1. Login to the Linux Virtual machine and switch users to the root user using the following commands (type what is underlined):

   ```
   [student@localhost ~]$ su -
   Password: <enter password instructor provides>
   ```

2. Login to the Linux Virtual Machine and change to the appropriate directory.

   ```
   [root@localhost ~]# cd /var/www/html
   ```

3. Create the following web page by using a standard editor, name the web page index.html. Uitilze either "vi", "nano" or "gedit" to modify files in the following portions of the exercise. All examples will be using "gedit" for simplicity.
   Please edit: /var/www/html/index.html (or index.html in /var/www/html)
   Example: `[root@localhost html]# gedit index.html`

   Please the following within the file:

   ```
   <H1> Application Security (SEC500) </H1>
   <HR>
   ```

   Exit and save the file (which is located in /var/www/html).

4. Determine the IP Address of the Linux Virtual Machine

   ```
   [root@localhost html]# ifconfig
   Or
   [root@localhost html]# ip addr show
   ```

5. User the WINDOWS 10 Virtual Machine to test the web site.

Hands On Exercise 2 (continued)

6. Create the following form by using

   ```
   [root@localhost html]# gedit myform.html
   ```

   Within the editor, please enter the following content:

   ```
   <form  name=”myform” method=”GET” Action=”mytest.php”>
        <input type=hidden name=”login”>
        Last Name <input type=text name=”Last”><br>
        First Name <input type=text name=”First”><br>
        Password <input type=password name=”Password”><br>
        <input type=”submit”> <input type=”reset”>
   </form>
   ```

7. Test the form using the Windows 10 Virtual Machine.

8. List the problems with the above form.

9. After further investigation you determine that the session numbers are predictable, is this a problem?

10. After further investigation you determine that there is only one level of security for all users, either accepted or failed, is this a problem?

Extra Credit

1. Create the following server-side application, which corresponds to the above script.

```
[root@localhost ~]# cd /var/www/html

[root@localhost html]# gedit mytest.php

<html>
<body>
<?php print "Hello World<br> ";?>
</body>
</html>
```

2. Test the form and server-side application by using your browser on the Windows 10 client and go to the IP Address of the Linux Virtual Machine. Type in some data and submit the form. You should get a response back of Hello World.
3. Copy from myform.html or create the following web page, to create another page to submit data.

   Note: Be sure to adjust the mytest.php to mytest2.php

```
[root@localhost ~]# cd /var/www/html

[root@localhost html]# gedit myform2.html


<form  name=”myform” method=”GET” Action=”mytest2.php”>
<input type=hidden name=”login” >
Last Name <input type=text name=”Last”> <br>
First Name <input type=text name=”First”> <br>
Password <input type=password name=”Password”> <br>
<input type=”submit”> <input type=”reset”>
</form>
```

4. Finally create the server side application

5. [root@localhost ~]# cd /var/www/html

6. [root@localhost html]# gedit mytest2.php

```
<html>
<body>
<?php
$Last= $_GET["Last"];
$First= $_GET["First"];
$Password= $_GET["Password"];
print "You entered $Last as your last name<br>";
print "You entered $Password as your Password<br> ";
?>
</body>
</html>
```

7. Use your browser on the Windows 10 host computer to test out the code. After you submit your form with some input data you should get a response back of your last name, first name, and password.

8. Explain how you could improve security by using multifactor authentication and SSL.

SEC500 Application Security Night 2

Exercise 3

1. What does the acronym HTTP stand for? (p. 39)
2. T/F The latest version of HTTP is 1.1? (p. 40)
3. Identify all sections of the HTTP Request below: (p. 40)

```
GET  /index.html   HTTP/1.1
```

4. What is the referer header? (p. 41)
5. What is the User-agent header? (p. 41)
6. What is the host header? (p. 41)
7. What is the cookie Header? (p. 41)
8. What is the difference between a GET and a POST? (p. 42 - 43)
9. List some other methods? (p. 42)
10. T/F All webservers support all methods. (p. 44)
11. T/F the PUT method is harmless. (p. 43)
12. T/F Cookies can be used for session tokens. (p. 47)
13. Identify Status codes. (Page 44) (p. 48)
14. List the HTTP Authentication types. (p. 50)
15. Evaluate the Java Platform and the ASP.NET. (p. 53 & 54)
16. Evaluate PHP and Perl. (p. 54 - 55)
17. What is Encoding? (p. 66)

SEC050 Application Security Night 2

Hands On Exercise 3

1. Use the Linux VM and telnet to itself using Port 80, and explain what you see.

```
[student@localhost ~]$ telnet 127.0.0.1 80 <press enter>
GET / HTTP/1.1 <press enter>
Host: 127.0.0.1 <press enter>
<press enter again>
```

2. Repeat problem 1 with http://workforcecenter.slu.edu/

```
[student@localhost ~]$ telnet workforcecenter.slu.edu 80 <press enter>
GET / HTTP/1.1 <press enter>
Host: workforcecenter.slu.edu <press enter>
<press enter again>
```

3. Repeat problem 1 with a website of your choice.

4. Use the Linux VM and telnet to itself using Port 80 and explain what you see.

```
[student@localhost ~]$ telnet 127.0.0.1 80 <press enter>
OPTION / HTTP/1.1 <press enter>
Host: 127.0.0.1 <press enter>
<press enter again>
```

SEC500 Application Security Night 2

Exercise 4

1. Complete this sentence, “The first step in the process of attacking an application is to _____ and _____ examine some key information ...”. (p. 73)
2. What is web spidering? (p. 74)
3. List some software that can be used for web spidering? (p. 74)
4. What is the file “robots.txt” used for? (p. 74)
5. T/F Web spidering is illegal. (p. 77)
6. What is User-Directed Spidering? (p. 77)
7. T/F Most web application tools run as a proxy on port 8080.
8. T/F Most web application tools require user browser to be redirected to the proxy.
9. Expand upon the term “Discovering Hidden content”. (p. 80)
10. What is a Web archives? (p. 89 - 90)
11. What is banner grabbing? (p. 101)
12. What is HTTP Fingerprinting? (p. 102)
13. List some common Directory name. (p. 105)
14. List some common Session Tokens. (p. 105)

SEC050 Application Security Night 2

Hands On Exercise 4

1. Login to the Windows 10 Virtual Machine, install Burp located on the Desktop in the SEC500 folder and within the Burpsuite folder. The file should be named something similar to:

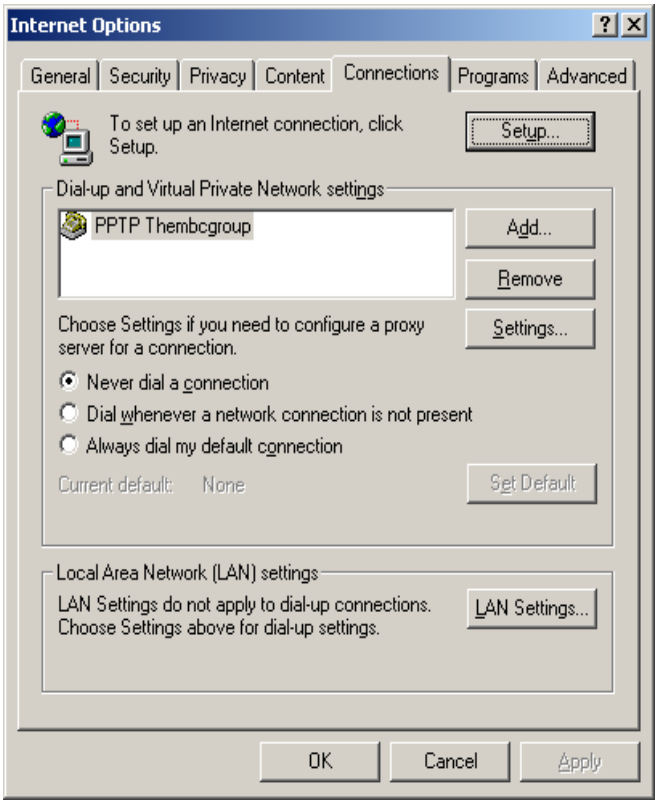
   burpsuite_community_windows-x64_v2020_2_1.exe

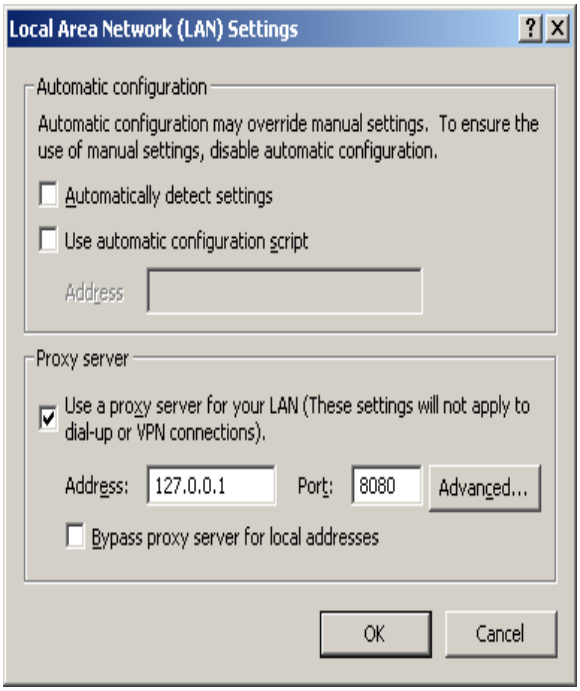
2. Start up Burp by clicking on the desktop icon.

3. On the Windows 10 Virtual Machine, configure Internet Explorer to use the proxy.

   Address: 127.0.0.1

   Port: 8080





4. On the Windows 10 Virtual Machine, do not configure Firefox to use the proxy.

5. On the Windows 10 Virtual Machine, go to www.ucla.edu. (Notice that the Burp program will start intercepting your data.

6. On Burp, Click on the Proxy tab, then the Intercept tab.

7. Notice you are seeing the data as it is being sent to your browser.

8. Next click on forward until all of the datagrams have been received.

9. Finally Click on the action button and select “send to spider”

10. Accept the addition of spider.

11. Next click on the target tab.

12. Download and install httprint and use it for http fingerprinting (a copy is in the SEC500 folder under “Tools”). Document your findings below.

13. Use web.archive.org and research an old website. Document your findings below.

SEC500 Application Security Night 3

Exercise 5

1. Complete this sentence " SQL injection is the elder statesman of code injection _attacks_, being still one of the more _prevelant_ _vulnerabilities_ in the wild, and frequently one of the most _devestating_.? (Page 237)
2. T/F Most programming environments already have base functions to handle SQL injection. (Class)
3. T/F There are other types of code injections other than SQL.
4. List other types of code injections.
5. T/F Interpreted languages are more vulnerable to code injection than compiled. (p. 288)
6. Finish this sentence " In the most serious cases, SQL _____ can enable an _____ _____ to read and modify ____ _____ stored with the database". (p. 291)
7. What does ending an SQL statement with "OR 1=1--" do? (p. 290)
8. T/F Storing user names and passwords in a database is more secure than a file.
9. T/F Storing user names and passwords in a database is the most secure.
10. T/F You cannot retrieve user name and passwords from Active Directory or an LDAP server.
11. Explain the example on page 288 "Bypassing a Login" (p. 288)
12. See page 294 and discuss the "Insert Statements". (p. 294)
13. See page 296 and discuss the "Update Statements". (p. 296)

14. See page 297 and discuss the “Delete Statements”. (p. 297)

15. See page 304 and discuss the “Union Operator”. (p. 304)

16. What does it mean to fingerprint a Database? (p.303)

17. Why would someone want to fingerprint a Database?

18. Explain pages 257-260 titled “An Oracle Hack”.

19. Explain pages 260-262 titled “An MS-SQL Hack”.

20. Explain the concept of “Bypassing Filters”. (p. 311)

21. Explain “Avoiding Blocked Characters”. (p. 311)

22. Explain “Circumventing Simple Validation”. (p. 312)

23. Explain the term “Out-of-Band Channel”. (p. 316)

24. T/F Out of Band Channel is the best security for databases and is unbeatable.

25. Explain the term “Leveraging the Operating System”. (p. 319)

26. Explain the term “Inference” as it applies to injection of code. (p. 319)

27. What is Absinthe? (Page 322)

28. See page 326 and explain xp_cmdshell. (p. 326)

29. See page 338 and discuss methods of preventing SQL Injections. (p. 338)

30. List other types of code injections.

31. See page 381-383 and discuss “Remote file Inclusion”. (p. 381 - 383)

32. See page 397-402 and discuss “Injecting into SMTP”. (p. 397 - 402)

33. See pages 326-30 and discuss “LDAP Injection”. (p.349 - 354)

SEC050 Application Security Night 3

Hands On Exercise 5

1. Ensure you web server on the Linux box is working by surfing to it from your Windows VM.

2. Startup BurpSuite on your Windows VM and configure your web browser to use Burp to contact other sites.

3. Using Burp's Proxy, connect to your Linux box using the URL:

   http://<Linux Host IP>/DVWA-1.9

   Logon to the site with username: "admin" and password: "password"

4. Once logged into DVWA, ensure the security level is set to “impossible” vice "low”. This can be done by clicking on "DVWA Security" on the lower, left side of the menu.

5. Change the Security Level to "Impossible", then click "Submit".
6. Now click on "SQL Injection". We won't be using this right now, but it will provide a sampling for different session variables.

7. Using Burp Suite, identify if there are any Cookies, hidden fields or other mechanisms within the application that may be of interest. Don't forget to look at the request and responses within Burp.

8. Record what you find below:

Hands On Exercise 6 (SQL Injection)

9. Ensure DVWA is still functioning. Change the security level to "low", the same way that you turned it to "impossible" in the past exercise. Ensure Burp is running on this so you can inspect or attack via burp.

10. Click on the "SQL Injection" vulnerability on the left side.

11. Near the top and center of the screen there is a "User ID:" prompt with a submit button. This is what you are attempting to use SQL Injection to attack.

12. Start with the basic SQL injection techniques and try to guess how this function is working. On objective is to understand how the statement is working before trying to SQL inject it.

    How does this page function?

    What is it being used for?

    How might you perform SQL injection against it?

13. Attempt to get all the User IDs to display by using the text entry box to create a SQL injection attack or via the Burp Intercept proxy.

14. If you would like a hint, you can click o the "View Source" button on the lower right hand of the screen - this will show you the code that is being executed at the current security level and allow you to have a better view of the SQL statements.

15. When you are able to inject code, record what and how you did it.

SEC500 Application Security Night 4

Exercise 7

1. Complete this sentence “The attacks described in this chapter are in a different category, because the primary target of the attacker is the _____ of other users.” (p. 431)
2. Complete this sentence “Cross-site scripting (or XSS) is the _____ of attacks against other _____.” (p. 432)
3. T/F Many professions don’t consider XSS real threats.
4. Define XSS.
5. Explain the term “Reflected XSS Vulnerabilities”. (p. 434)
6. See page 436 and explain the steps in “Exploiting the Vulnerability”. (p. 436)
7. Explain the term “Stored XSS Vulnerabilities”. (p. 438)
8. Explain the term “Storing XSS in Uploaded Files”. (Page 484-487)

9. Explain the term “Chaining XSS and Other Attacks”? (p. 450)

10. What is Virtual Defacement? (p. 443)

11. Give examples of escalating the Client-side Attacks.

12. Explain Beating Sanitization. (p. 468)

13. Expand on Preventing Reflected and Stored XSS. (p. 492)

Lab 7 - XSS (Cross Site Scripting)

XSS (Reflected) Lab

1. You may choose to use Burp or not for this exercise. If you choose to use it, please ensure you (a) start up Burp Suite (b) configure your browser to use Burp Suite and (c) enable or disable the intercept proxy as you see fit so pages load.

2. Use Firefox or IE to open the DVWA web site (http://<your Linux IP here>/DVWA-1.9/).

NOTE: Chrome has enough XSS protections built in, it is hard to demonstrate the issue with Google's browser (this is a good and bad thing).

3. Ensure the security level of DVWA is set "low" in the beginning of this lab.

4. Click on the "XSS (Reflect)" option on the left side of the screen.

5. Within the "What's your name?" text box, you are permitted to put in any text and that text will be re-displayed to the window. As an example, put your name in the box and click "submit" and it will be redisplayed on the page.

6. Now, attempt to place some JavaScript that will generate a pop-up window. You may want to look at the content via View Source, if it is not working properly or via Burp if you are using it.

7. Experiment to get a pop-up window to display the cookie information from the website.

8. Record what you entered that successfully displayed the cookie data:

9. (optional) Next try to increase the security level to "medium" an attempt to get the pop-up to occur. Additionally, you can use the "View Source" button to examine the source code to identify what is going on behind the scenes of the application, if that will help you.

XSS (Stored) Lab

1. Now attempt the same experiment with the "XSS (Stored)" sample page (left side).

2. Be sure to adjust the security of the application back to "Low", if you adjusted it in the past lab.

3. In this example, you have the ability to sign the guest book and store data within the database so each subsequent page view will show all guest book entries. This is an ideal method to collect numerous sets of credentials (or in our case make pop-ups occur after each page view).

4. How does this lab differ from the XSS (Reflected) Lab? What challenges were posed in this lab?

5. What additional steps did you have to do to circumvent some of the simple protections?

6. Increase the security of the site and see if you can be successful on this page. Where you and what did you have to change?

7. If you need to clear successful attempts that are stored, you can click on the "Setup / Reset DB" tab and then click on the "Create / Reset Database" - this will reset the database to it's original configuration, to prevent the stored XSS from re-appearing on future page views.

SEC050 Application Security Night 4

Final

1. What is the core security problem?
2. Expand upon Cross Site Scripting.

3. Expand upon SQL Injection.

4. Expand upon Information Leakage.

5. Expand upon “The New Security Perimeter”. (Pages 12 and 13)

6. See page 13 and complete this sentence: A malicious attacker can leverage a benign but vulnerable web application to attack _____ ____ ____ ____ ____.

7. Use the Internet and identify a web application vulnerability assessment tool.

8. List some web servers?

9. Expand upon the term “Reject Known Bad”. (Page 23)

10. Expand upon the term “Accept Known Good”.

11. Expand upon Sanitization.

12. See page 28 and expand upon multistep validation and Canonicalization?

13. Why are logs important?

10. What does it mean to fingerprint a Database? (Page 303)

11. Explain “Avoiding Blocked Characters”.

12. Explain “Circumventing Simple Validation”.

14. Explain the term “Out-of-Band Channel”. (Page 316)

15. Define XSS.