CSE 345/545: Foundations to Computer Security

Homework Assignment II (100 Points)

Due: 2359hrs 14 October 2018

Plagiarism policies will be strictly enforced.

A large number of security vulnerabilities in a software system are the results of design and programming errors, some of which are covered in this lab assignment.

Part I [10 points]

IIIT-Delhi has a campus-wide 802.1X wireless network.

1. Explain the authentication technique used by IIIT-D, if any.
2. Is the IIIT-Ds wireless service susceptible to packet sniffing? Provide justification for your answer.
3. Describe what you would consider to be a typical Internet usage session. What could a potential attacker learn by sniffing this traffic?

Part II [25 points]

In this problem, you will use JavaScript to exploit Cross-Site Scripting (XSS) vulnerabilities. Open the part2.html using a web browser, and answer the following questions:

```
<HTML>
<style type="text/css">
div.board {border:1px solid #A4A4A4;
        padding:3px;background-color:#EEEEEE;width:'70%';text-align:left;}
</style>
<script type="text/javascript">
var newDiv = null;
function Write()
{   var userInput = document.getElementById('userInput').value;
    var _body = document.getElementsByTagName('body') [0];
    var a = "<Center><div class='board'>";
    var b = "</div></Center>";
    newDiv = document.createElement("div");
    newDiv.innerHTML = a+userInput+b;
    _body.appendChild(newDiv);
}
</script>
<body>
<P align=center><b><font color="#003399" size="5"><br>CSE345/545 Foundation to Computer Security</br></b> <i>Monsoon 2018</i></font></p>
<CENTER><IMG id="logo" src=" https://az616578.vo.msecnd.net/files/2016/06/17/636017786945631331-1203241112_information-age.jpg"> <BR><BR>
<a href=" https://www.iiitd.ac.in/" target=_blank> Go to IIIT-D homepage </a> <BR><BR>
<a href=" http://mail.iiitd.ac.in/" target=_blank> Check your email</a> <BR><BR><BR>
<font color="#003399" size="5"> <b> Discussion Board </b> </font><BR><BR>
<div class="board">
<input type='text' id='userInput' size='100' value='Enter your input here' />
<input type='button' onclick='Write()' value='Submit'/> <BR><BR>
</div>
<div class="board">This discussion board is vulnerable to Cross-Site Scripting (XSS) attack.</div>
<div class="board"> Hi <a href="javascript:void(document.getElementById('logo').src=' https://az616578.vo.msecnd.net/files/2016/06/17/636017786945631331-1203241112_information-age.jpg’);"> check out this cool link </a> </div>
<div class="board"> Hello, <a href="javascript:void(document.getElementsByTagName('a')[0].href='http://www.google.com');" > see this page </a> </div>
</BODY>
</HTML>
```

1. You can instruct web browsers to run JavaScript by typing JavaScript URIs in the address bar. For example, try to type the following URI into your browsers address bar.

    javascript:alert(’hello world’);

Describe what happens. Explain how the browser will interpret and process the JavaScript URI.

2. Execute the JavaScript snippet *javascript:void(document.getElementById('logo').src= ‘https://tctechcrunch2011.files.wordpress.com/2017/08/horror-movie.png?w=1279&h=727&crop=1’);* in the browser:

Describe what happens. Check whether the source code of this webpage has been changed. Explain how can an attacker change the contents of webpage without modifying its source code?

3. Web sites containing user-created content (UCC) are at great risk of having Cross-Site Scripting (XSS) vulnerabilities. The given web page contains a Discussion Board. Assume that the Discussion Board is dynamically updated by users. Click the link check out this cool link in the Discussion Board and describe what happens.

Modify the given source code to add a link in the Discussion Board that replaces the logo image with https://www.wpblog.com/wp-content/uploads/2017/08/wordpress-site-is-hacked.jpg

4. Using the Submit button, you can write anything on the Discussion Board. Add a link on the Discussion Board that replaces the logo image in your web browser with https://www.wpblog.com/wp-content/uploads/2017/08/wordpress-site-is-hacked.jpg.

What did you add on the Discussion Board?

5. Click the link Go to IIIT-D homepage to the IIIT-Ds homepage. Click the link Hello, see this page in the Discussion Board. Then, click the link Go to IIIT-D homepage again. Describe what happens. Check whether the source code of this webpage has been changed. How can an attacker trick other users' to visit a fake website without modifying the source code?

6. There is a link Check your email in the webpage obtained by using the above HTML code. Change the HTML code to add a link in the Discussion Board to make the link Check your email to lead to https://www.reddit.com/r/hacking/. How can an attacker steal other users email login information using this vulnerability?

7. Besides changing the content of a webpage, an attacker can also inject malicious code and execute it in your web browser using XSS vulnerabilities. This is very dangerous because the malicious code will have the same privilege as your web browser. In this part, you will simulate Session Hijacking attack which is one of the most common attack patterns using XSS vulnerabilities.

Set your Session ID as 12345 by typing the following JavaScript URI into your browsers address bar.

javascript:void(document.cookie='Session ID=12345');

Check whether the Session ID is set correctly by typing the following JavaScript URI into your browsers address bar.

javascript:alert(document.cookie);

Add the link given in db link.htm to the Discussion Board using the Submit button.

Click the new link and describe what happens. How can an attacker steal your Session ID using this vulnerability?

8. How can you prevent the XSS vulnerabilities? Modify the code to prevent it.

9. Create a Self signed Certificate using OpenSSL and host the code using any webserver of your choice (SSL). You can get creative and try without using a webserver.

Think Shakespeare !

10. Would Using https do any good for the security of the site in the context discussed so far.

Part III [15 points]

Use any web security application testing tool that serves as a proxy for intercepting browser web requests and web server replies.

1. With the tool turned on, go to your favorite Web sites and interact with them, such as logging

and downloading files. Explain the kinds of information that can be captured.

2. List three vulnerabilities/possible attacks that can occur by exploiting the data obtained.

3. If you were a developer, what could you do to prevent misuse of this data? Provide justification on why your approach would prevent the misuse of the data.

Tool you can use is OwaspZap /WebScarab available at:
http://sourceforge.net/projects/owasp/files/WebScarab/

Some tutorial links are here: https://www.owasp.org/index.php/WebScarab Getting Started

http://yehg.net/lab/pr0js/training/webscarab.php

Part IV [50 points]

1. [25 points] Aalok, an IIITD Student needs to implement a simple OTP generation program. He cannot use any OTP generation library directly but is allowed to use hash library (hashlib) in python for generating hash values for any input. He thinks of an idea to generate and verify the OTP involving below steps: -

- He plans to retrieve the current system time and date to be taken as input for generating hash as the value would be unique each time.
- He then plans to generate hash (say **H**) of the string of date-time using SHA-256 function (using hashlib) and plans on storing the Hash **H** thus generated in a text file.
- Now he thinks of implementing a function (say **F**) which takes a hash value **H** as input to process specific characters/numbers from **H** to generate an OTP (say **O**) such that OTP is of length 4 and is completely numeric.
- He then plans to send the OTP **O** to the user and keep the value **H** with him (stored in the text file).
- For verification, he plans to build a simple program which takes **O** as input and **H** as input. It then performs function **F** on hash value **H** to give output **O'**. Now he'll check if **O** matches **O'.** If it matches, OTP verification will be "Successful", otherwise it will be "Unsuccessful". Once an H matches, he'll remove that **H** from the text file. For simplicity, it is assumed that only one hash value H is generated and stored at a time

a) You need to implement both of the OTP generation and OTP verification programs for Aalok and also describe the details of how did you implement function F. Also Reason why you chose a specific function F.

b) Could you think of any vulnerabilities in this OTP generation approach and in your implementation of F ? If yes, mention them and explain how could they be exploited. If not, reason how is this approach and your F implementation secure.

Note - You are not allowed to directly use pyotp or any other direct OTP generation libraries. If you're using Java / C then you can use a SHA-256 hash generation library but no direct OTP generation libraries allowed.

2. [10 points] Use the **OpenSSL C** library to create a naive chat application, which sends and receives encrypted data. Be creative in design so no instructions here. (Only C language allowed for this question).

   Verify by using Wireshark. (Provide screenshots of credentials sniffed using wireshark)

3. [10 points] Download the ToR source code (https://www.torproject.org/download/download.html.en). **Don't download the ToR Browser Bundle, build only ToR from Source Code**. Run the binary on the command line, and configure your browser to use it. The configuration should be such that your computer should accept connections from any computer on IIITD-LAN. Bonus Points for using bridges and even more bonus points for using any transport method like obs4 etc.

4. [5 points] Create your own key file, public/private keys as deemed appropriate. Encrypt the large file.txt using OpenSSL tool:

- With a stream Cipher RC4

- With a symmetric Cipher AES-256

- With RSA

Time all the encryption algorithms. Compare the encryption times for the three classes of encryption algorithms. Can you reason sanely why some are blisteringly fast and others terribly slow?

Submission Guidelines (points will be deducted if not followed):

Submit a zip file containing

- A pdf file with brief steps and screen shots completely documenting the process, as you went. Note that while taking screen-shots your username is visible in the screen shot.

- The Source code of programs you write.

Please post it onto Backpack by the deadline. Do not send it by email! No email submissions will be entertained.