Andrew Plum

Mr. Alves-Foss

Cyb 110

12/12/21

Ch. 13 Homework

1) Fuzzing tools are used to find unexpected problems through the process of fuzz testing. Fuzzing tools work to find the unexpected problems by bombarding applications with all types of data and inputs from a large variety of sources to see if the input can fail or perform some unexpected behavior. Fuzzing is commonly used among security researchers and those conducting security assessments of applications. Some fuzzing tools have a specific purpose and others have a more general focus.

3) It is important to remove extraneous files from a web server because you may hand the attackers exactly the information and resources they need to successfully compromise your system. Extraneous files are files not directly related to run a site or application. Archives of source code of your applications, backup copies of your files, text files containing notes or credentials, or any other related files are all examples of extraneous files which attackers can use to learn how to compromise your system. The best practice is to remove all extraneous files from the web server so the attackers can't use them when attacking your system.

4) The tool Burp Suite is a web application analysis tool which includes several more advanced features for conducting more in-depth attacks in addition to the standard sets of features found in any assessment tool. Burp Suite is on the lower end of the cost spectrum for web application analysis tools, but still provides a decent set of features. Burp Suite is also available as a free community version which allows you use of its standard and assessment tools, but it doesn't include the use of its more advanced features. You may want to use Burp Suite when you are on a low budget or if you aren't sure if you will be frequently using the web analysis application tool you purchase.

5) The two main categories of web security are the sides which deal with client-side attacks and server-side attacks. Client-side attacks exploit weak software loaded on the user's clients or use social engineering to fool the user. Some examples of client-side attacks include cross-site scripting, cross-site request forgery and clickjacking. Server-side attacks exploit vulnerabilities on the server side of web transactions. These vulnerabilities can vary widely depending on your operating system, web server software, and its versions, scripting languages, and many other factors. However, these vulnerabilities on the server side are usually caused by a few common factors; these are: lack of input validation, improper or inadequate permissions, and extraneous files.

7) Input validation is important because some of the most common server-side web attacks exploit the lack of input validation on the server-side. Software developers often neglect to properly validate user input, which makes the web server vulnerable. Lack of input can lead to successful directory traversal attacks where attackers can access files outside of the web server's structure. The important thing though is that these attacks can be stopped completely by implementing input validation and filtering out various special characters which can be used in such attacks.

10) You can prevent buffer overflows in all your applications by implementing a process known as bounds checking where you limit the amount of data each input can take in. Buffer overflows, also

known as buffer overruns, occur when the data input of your applications isn't accounted for properly. When more characters are entered than the amount that is allocated in storage the excess characters may overwrite other areas in memory used by the operating system or other applications. Buffer overflows can be used by attackers to alter other applications or cause the operating system to execute their own commands. The good thing is buffer overflows can be stopped with proper bounds checking and some computer science languages implement bounds checking automatically.