**Everything Highlighted in Yellow is the answer to the question**

**Directory/Path Traversal**

Developed by: Jebel Macias

**Directory/Path traversal attacks** is a type of HTTP attack that allows outside users to access directories that are not available from the web root directory. To perform a directory/path traversal attack, attackers will attempt to insert malicious string that use a sequence of the characters “../” to step up a level in the directory structure. Once access has been gained to restricted directories, attackers are now capable of viewing any sensitive information that was not intended to be viewed by outside users.

**What you will learn:** You will learn how to test for directory/path traversal vulnerabilities on a sample web server that has been created for you. You will learn how to use DotDotPwn, an intelligent fuzzer, to test for traversal vulnerabilities. You will learn to how to circumvent defenses against directory/path traversal attack by using cookie spoofing.

**Goal:** You are going to attempt to access a file (passwords.txt) outside of the web root directory(/var) by using a Directory/Path traversal attack.

Part 1 – Initial Web Server Setup

1. On the virtual machine, you will use the credentials below to login into the root account.

**username: root**

**password: root**

1. Find the terminal icon on the top-left of the screen and start a terminal.
2. Next, you will start up the web server using apache. To start an apache server, in the terminal run the following command:

**service apache2 start**

1. Next you will need your IP address. To obtain your IP address run the following command:

**ifconfig eth0**

1. What is your IP address? (Hint: It proceeds the inet label)
2. To check if the server is running, open a web browser  in kali linux, and enter the following url:

**http://<Answer to Question 5>/**

1. Once you have performed the previous steps, the web page should output the same web page as seen in figure 1.

Graphical user interface, text, application

Description automatically generated

Figure 1: Initial page on the local web server

Part 2 – Directory Traversal Fuzzing using DotDotPwn

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Fuzzer** is an automated testing tool that injects random data as input on systems to test for vulnerabilities and other bugs.

**DotDotPwn** is a fuzzer used to discover path/directory traversal vulnerabilities on HTTP/FTP/TFTP web servers. DotDotPwn produces a list of possible paths on a web server that are vulnerable to path/directory traversal attacks.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Open the terminal once again, and run the following command:

**./dotdotpwn.pl -m http -h <Answer to question 5> -f /password/passwords.txt -d 2 -t 100 -b**

1. Once you have run the command in question 8, the resulting output will be displayed as shown in figure 2. **Press enter** to begin testing for path/directory traversal vulnerabilities.

Text

Description automatically generated

Figure 2: DotDotPwn initial traversal report

1. What testing path was vulnerable according to dotdotpwn?
2. Next, enter the answer to question 10 into a web browser url and explain what you see? Were you able to gain access to the passwords.txt file?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Why do you think the path provided was not able to gain access to the passwords.txt file?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The dotdotpwn module provided a false positive result not allowing us to access the passwords.txt file. Furthermore, apache like many other web servers do not serve files outside of the web root directory.

Part 3 – Directory Traversal Attack by Cookie Spoofing

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

In the previous part, there were defenses against the path/directory traversal attacks setup by the apache server as apache servers do not serve files outside of the web root directory. Therefore, we will attempt to find other possible vulnerabilities in our web server script so that directory traversal attack may be performed.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. In the desktop on the virtual machine, there is a file named vulnerable.php. Open this file and view the contents. This is the initial page you see on the web server when you enter the URL in question 6.
2. What line in the script allows attackers to perform a path traversal attack? Why?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The second to last line is the issue, as it allows outside change the directory being served.

Now that we know what line in the php script is vulnerable, we will attempt to use cookie spoofing to perform the path traversal attack.

1. What is the name of the cookie we should attempt to spoof for our directory traversal attack?

filepath

Now that we have the cookie name, we need to determine the path to the password.txt file. Remember that the sequence of characters “../” are used to step a level up in the directory structure. The webroot directory is found at /var/, and the password file is found at /password/.

1. What is the path from the webroot directory to the password.txt file? (Hint: Your answer should look like ../../password/passwords.txt)

../password/passwords.txt

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

cURL is a library and command line tool that allows user to transfer data using different network protocols.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Finally, we can perform the attack now that we have both the path to the password.txt file and the cookie name. You will construct a curl command to spoof the cookie, and gain access to the contents of the password.txt file.

1. Open the terminal, and run the following command:

**curl -b “<Answer to question 15>=<Answer to question 16>” <Answer to question 5>**

1. If the path and cookie you provided were correct, the command in question 17 displays the contents of the password.txt file. If the contents of the passwords.txt file were not printed, redo questions 15-17.
2. Provide the correct passwords corresponding to each user in the passwords.txt file.

**Kyle: 12345**

**Leo: timHonks@12.asd**

**Tommy: lemonPotPie**

**Laura: Almonds123**

**Congratulations! You have completed the exercise**

**Homework Exercise**

In this exercise you will attempt to access the passwd file in the etc directory and determine how to mitigate the directory/path traversal vulnerability.

1. What is the path from the web root directory to the passwd file in the etc directory? (Hint: Your answer should look like the answer you provided in question 16)

../etc/passwd

1. Generate a curl command to gain access to the passwd file (Hint: Curl command should look like the command provided in question 17):

curl -b “filepath=../etc/passwd” 10.0.2.15

1. What changes should be made to vulnerable.php to prevent future directory/path traversal attacks?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_