OWASP Security Misconfiguration Lab Exercise

The development of this document is/was funded by three grants from the DOD Grant Number H98230-19-1-0301

1 Overview

This Labtainer exercise explores Security misconfigurations. As the name suggests, expose vulnerabilities due to weak configurations of an IT asset. It doesn't affect web assets only. Any component which requires a configuration is subject to this vulnerability. This means that network devices, hardware, email services, etc. can suffer from this vulnerability.

2 Lab Environment

This lab runs in the Labtainer framework, available at http://my.nps.edu/web/c3o/labtainers. That site includes links to a pre-built virtual machine that has Labtainers installed, however Labtainers can be run on any Linux host that supports Docker containers.

From your labtainer-student directory start the lab using:

```
labtainer web-secmis
```

On most Linux systems, these are links that you can right click on and select "Open Link". **If you chose to edit the lab report on a different system, you are responsible for copying the completed report back to the displayed path on your Linux system before using "stoplab" to stop the lab for the last time.**

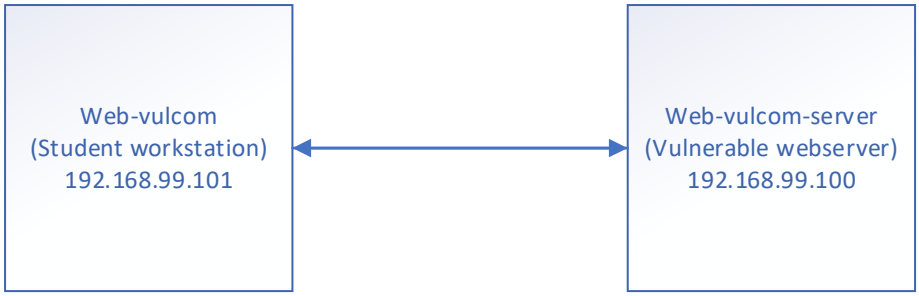
The resulting virtual terminal is connected to the student workstation; you will have OWASP ZAP and Firefox located on this workstation.

There are several accounts that are configured

Username	Password	Systems
admin@juice.org	admin123	Admin for web login
jim@juice.org	ncc-1701	web login
mc.safesearch@juice.org	Mr. N00dles	web login
Ubuntu	ubuntu	Student workstation
Ubuntu	ubuntu	vulnerable server
Ubuntu	ubuntu	Attacker workstation

3 Network Configuration

The student workstation (`web-secmis`) is configured to have IP address `192.168.99.101` while the vulnerable webserver (`web-secmis`-server) is `192.168.99.100;`



4 Lab Tasks

It is assumed that the student has received instruction or independent study on the basic operation of web operations.

4.1 Verify connectivity between student workstation and web server

A simple ping from the student workstation system will be sufficient.

```
ping 192.168.99.100
```

Note: to stop the ping use CTRL + C

4.2 Open Firefox and browse to the web server

At a terminal on the student workstation type:

```
firefox &
```

this will load Firefox, and type in the IP of the web server:

```
http://192.168.99.100
```

Record in Item #1 of your report why firefox might have been chosen to be the web browser used.

4.3 Open & Set up OWASP ZAP

At the terminal of the student workstation, type:

```
owasp-zap &
```

Note: if Firefox is running at the terminal and the "&" was not included then Firefox is not running in the background. Close Firefox and reopen using "Firefox &" at the terminal

OWASP ZAP Application should be open and it should be prompting the user for input.

- OWASP ZAP user input: select "yes, I want to persist this session with the name based on the current timestamp" then click start. This will open ZAP application.
- If you are prompted to "Manage Add-on" click close

4.4 Configure Firefox to use OWASP-ZAP as a Proxy

The objective of this task is to set up OWASP ZAP to be function and to allow the capture of traffic from the student workstation. Within the preference section of Firefox configure the following steps:

- In the student workstations Firefox, open "Preferences"
- In the find window type "proxy"
- In Network Proxy Setting, select "Settings"

- Select “Manual proxy configuration”
- In the HTTP Proxy section: use “127.0.0.1” and Port “8080”
- Also select “Use this proxy server for all protocols”
- Click “ok” to accept the settings

The above setting ensures that Firefox will use OWASP Zap as the proxy. Perform the following steps to ensure the Firefox is connecting and using ZAP as a proxy

- Refresh the webpage “192.168.99.100”
- A security warning stating “Your connection is not secure” will be displayed.
- This warning message must be accepted. To do that click on “Advanced”
- It will display the SSL certificate and should show a “SEC_ERROR_UNKOWN ISSUE” it is ok to use this cert, click “Add Exception”
- A confirmation window will pop up, confirm the exception by clicking the “Confirm Security Exception”

Record in Item #2 of your report why set up a proxy.

4.5 Accessing Restricted Areas

The objective of this section is to see what access is allowed and to determine even if you can access a restricted area if you will be allowed to do anything. According to OWASP WSTG-ATHZ-01, one of the first tasks when looking at a web site is to see what URLs and directories you have access to. Conducting a basic site survey using a web site crawler is the preferred method for data gathering on a web site.

Example 1 – Path Traversal

The following will allow OWASP Zap to crawl a website to determine if what paths are accessible.

- Open OWASP Zap and perform a site scan on the IP address:
  `192.168.99.100:3000`
- Under alerts you should see a section titled Path Traversal, see if you can see any paths that you would think are password protect or secured.
- Do you see any links to administrative pages?
- Please save your scan results from OWASP Zap, save it to the desktop and title it “traversal.html”

Example 2 – feedback modification

- In Firefox navigate to the following URL
  `http://192.168.99.100:3000/#/administatration`
- Once you are at the administrative page are you able to remove feedback? Remove all 1-star feedback

Record in Item #3 of your report how should an administrative page be protected?

4.6 Provoke Error Handling

The objective of this task is to example how the server handles unexpected errors and how it handles them. According to OWASP WSTG-ERRH-01, often, during a penetration test on web applications, we come up against many error codes generated from applications or web servers. It’s possible to cause these errors to be displayed by using a particular requests, either specially crafted with tools or created manually. These codes are very useful to penetration testers during their activities, because they reveal a

lot of information about databases, bugs, and other technological components directly linked with web applications. A common error that we can see during testing is the HTTP 404 Not Found.
The following steps should provoke errors from the web server.

- From your student workstation navigate to the following URLs
  https://192.168.99.100:3000/ftp
  https://192.168.99.100:3000/ftps
  https://192.168.99.100:3000/rest
  https://192.168.99.100:3000/users
- Which of the above URLs caused an error? Were all of the errors the same?

Record in Item #4 of your report why is error handling and error reporting a critical step when it comes to security of a web server?

4.7 Explore Remaining Deprecated Interfaces

The objective of this task is to see if you can access an area of the site that has or uses a deprecated or obsolete function. According to OWASP, the use of deprecated or obsolete functions may indicate neglected code. As programming languages evolve, functions occasionally become obsolete due to functions that are removed are usually replaced by newer counterparts that perform the same task in some different and hopefully improved way.

The following steps should allow for deprecating an interface that wasn't locked down.

- From your student workstation log in as Jim.
- Click on the Complaint form, in the Contact Us dropdown to go to the File Complaint form
- Clicking the file upload button for Invoice and browsing some directories you might notice that .pdf and .zip files are filtered by default.
- Trying to upload another other file will probably give you an error message on the UI stating exactly that: Forbidden file type. Only PDF, ZIP allowed.
- Open the main.js in your `DevTools` and find the declaration of the file upload (e.g. by searching for zip).
- In the `allowedMimeType` array you will notice "`application/xml`" and "`text/xml`" along with the expected PDF and ZIP types.
- Click on the Choose File button.
- In the File Name field enter *.xml and select any XML file. Press open. Note: You may need to create a file with the .xml extension.
- Enter some Message text and press Submit
- On the JavaScript Console of your browser you will see a suspicious 410 (Gone) HTTP Error. In the corresponding entry in the Network section of your browser's DevTools, you should see an error message, telling you that B2B customer complaints via file upload have been deprecated for security reasons!

Record in Item #5 of your report why s removing old remnants of a website crucial to protecting a new site?

4.8 Gain Access to Paid Feature without Paying

The objective of this task is to see if you can access an area of the site that requires elevated permissions. According to OWASP WSTG-CONF-05, elevated interfaces may be present in the application or on the application server to allow certain users to undertake privileged activities on the site. Tests should be undertaken to reveal if and how this privileged functionality can be accessed by an unauthorized or standard user.

To test the elevated interfaces, follow the steps below.

- Log in as Jim and navigate to
  http://192.168.99.100:3000/#/score-board
- Inspecting the HTML source of the corresponding row (Premium Paywell) in the Score Board table reveals a HTML comment that is obviously encrypted:
  <!--
  IvLuRfBJYlmStf9XfL6ckJFngyd9LfV1JaaN/KRTPQPidTuJ7FR+D/nkWJUF+0xUF07C
  eCeqYfxq+OJVVa0gNb
  qgYkUNvn//UbE7e95C+6e+7GtdpqJ8mqm4WcPvUGIUxmGLTTAC2+G9UuFCD1DU
  jg==--> .
- This is a cipher text that came out of an AES-encryption using AES256 in CBC mode.
- To get the key and the IV, you should run a Forced Directory Browsing attack against the application. You can use OWASP ZAP for this purpose.
- The search will uncover https://192.168.99.100:3000/encryptionkeys as a browsable directory
- Open https://192.168.99.100:3000/encryptionkeys/premium.key to retrieve the AES encryption key EA99A61D92D2955B1E9285B55BF2AD42 and the IV 1337
- In order to decrypt the cipher text, it is best to use openssl.
- On your student workstation, in a terminal, type:

```
echo
"IvLuRfBJYlmStf9XfL6ckJFngyd9LfV1JaaN/KRTPQPidTuJ7FR+D/nkWJUF+0xUF07CeCeqYfxq+O
JVVa0gNbqgYkUNvn//UbE7e95C+6e+7GtdpqJ8mqm4WcPvUGIUxmGLTTAC2+G9UuFCD1DUj
g==" | openssl enc -d -aes-256-cbc -K EA99A61D92D2955B1E9285B55BF2AD42 -iv
1337133713371337 -a -A
```

- The response should be:
  /this/page/is/hidden/behind/an/incredibly/high/paywall/that/could/only/be/unlocke
  d/by/sending/1btc/to/us
- Navigate to the page.
  https://192.168.99.100:3000/this/page/is/hidden/behind/an/incredibly/high/paywall/that/could/onl
  y/be/unlocke d/by/sending/1btc/to/us

Record in Item #6 of your report how would web server administrator protect features that are paid for?

Record in Item #7 of your report what is the loss if paid features are able to be exploited for free?

4.9 Account Impersonation

The objective of this section is to see if you can post feedback while log in to one account to a different account. According to OWASP WSTG-ATHZ-02, a common testing method is to ensure that each session is only accessible to the owner of that session and that tasks performed by the session owner is

only directly effective that user. This section reviews if it is possible to access functions and resources that should be accessible to a user that holds a different role or privilege.

Example 1 – Forged feedback

- In Firefox, navigate to Customer feedback, if necessary, sign in as jim to view this page.
- In Firefox under comments, leave a comment. Take note of the Author. Leave a Ratting and enter the CAPTCHA result, don't click submit yet submit.
- In OWASP ZAP ensure that you have breakpoints set up and is turned on.
- In Firefox click submit
- In OWASP ZAP you will see that the submit was intercepted. In the body of them message you should see a line that looks like the following:

```
{"UserID:1,"captchaId":1,"captcha":"30","comment":
"comment entered (***.org)","rating":4}
```

  NOTE: the captcha might be different.
  - How to read the above line:
    - UserID:1 – this is the user ID who is posting the comment
    - captchaId:1 – this is the captcha question
    - captcha:30 – this is the captcha answer
    - comment: "comment entered (***.org)" – this is the comment being left
    - "rating":4 – this is the rating that was left
- Modify the User ID to number 1, and allow the message to go through.
- Check feedback and see at the following link and see if the user was the one you logged in as or if it was Jim or Admin who posted the feedback. URL is:

  http://192.168.99.100:3000/#/administatration

Record in Item #8 of your report what is account impersonation?

4.10 User login without a known password

The objective of this task is to see if there are ways past requiring a user to login in the traditional sense. According to OWASP WSTG-ATHN-04, a common method of testing is known as Black-Box Testing using either an Injection or direct page request method for bypassing the authentication schema.

Example 1 – bypass user authentication using double dash

- By simply appending a "'—" at the end of any user, this allows for a login with any password.
- Try logging in with jim@juice.org'-- and use anything for the password.
- What other users are susceptible to this this type of authentication bypass? User the user list below and see which users will allow this exploit to function.

Example 2 – bypass user email by using a name

- After completing example 1, and seeing the damage having two dashes can cause. Let's see if we can log in with a user name instead of an email address.
- At the log in page type:

```
('%jim%');--
```

- Select any password, did the login work?

Record in Item #9 of your report why does the dash dash method allow for users to log in without having a proper password?

4.11 Generate Report

In OWASP ZAP once the tester has found the different vulnerabilities, in the ZAP application under Report on the top and save a HTML report. Save the report to the home directory of the web-inject workstation, call the file "`report_zap`"

Record in Item #10 of your report any interesting finding of the report. Find three areas of interest and explain why they were important and how they may have an impact on the security of this website.

4.12 Review Journal output

This task allows the user to see output that the server would be generating when certain attacks or exploits have run against them. The learner will review journal output on the server.

- On the shell of the web server type the following command:
  - Sudo journalctl -u juice-shop
- Review the output and space bar until the end, record anything of note and if there are any possible exploits that had ran. Support your claim with output from the journal. Record this at the end of your report under question 11.

5 Stop the Labtainer

When the lab is completed, or you'd like to stop working for a while, run

```
stoplab web-secmis
```

from the host Labtainer working directory. You can always restart the Labtainer to continue your work. When the Labtainer is stopped, a zip file is created and copied to a location displayed by the stoplab command. When the lab is completed, send that zip file to the instructor.

References