OWASP Sensitive Data Exposure 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 the disclosure of data, which is not meant to be publicly accessible, this is known as sensitive data exposure (SDE). On the one hand, this data can be at rest, like a databases or files. On the other hand, it can be in transit. Apart from exposing customers' data which is a scandal.

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-sde
```

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.**

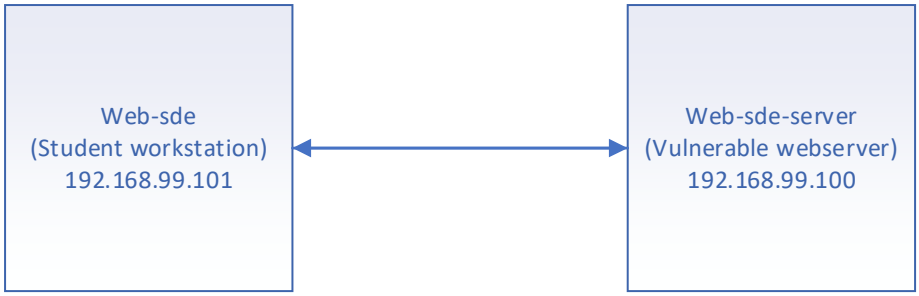
The resulting virtual terminal is connected to the student workstation; you will have OWASP ZAP and Firefox located on this workstation.

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
Ubuntu	ubuntu	Student workstation
Ubuntu	ubuntu	vulnerable server

3 Network Configuration

The student workstation (`web-sde`) is configured to have IP address `192.168.99.101` while the vulnerable webserver (`web-sde` -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 web-xss 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 – Web Survey

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”

Record in Item #3 of your report How would one typically restrict access to an administrative portion of a web site?

Record in Item #4 of your report Why are path traversal an issue? Can OS common be injected if the directory path can reach the root of a storage drive?

4.6 Accessing Confidential information

The objective of this task is to explore recent found directories to see if they contain sensitive data. According to OWASP WSTG-INFO-05, it is very common, and even recommended, for programmers to include detailed comments and metadata on their source code. However, comments and metadata included into the HTML code might reveal internal information that should not be available to potential attackers. The object of this is to first review webpage comments and metadata to better understand the application and to find any information leakage. Second to identify and gather JavaScript files, review JavaScript code in an application to better understand the application and to find any information leakage.

Example 1 – Gaining Access to confidential document

- Upon successful complete of a site survey, there were a few files that were identified that might warrant a closure inspection. Follow the link to titled "Check out terms of use" the URL is

  `https://192.168.99.100:3000/ftp/legal.md?md_debug=true`

  - The link is also on the About Us page.
- You can modify the URL to access the directory directly, the URL is

  `https://192.168.99.100:3000/ftp`

- Review all of the files and see which one may have sensitive data in it.
- You should be able to open the following URL

  `https://192.168.99.100:3000/ftp/acquisitions.md`
- Save the acquisitions.md file to the desktop of your workstation.

Example 2 – Finding hashes

- Upon further review of the ftp directory of the server, you may notice a file that seems to contain passwords.
- Access this file at:

  `https://192.168.99.100:3000/ftp/password_list#.txt%2500.md`

  - Note: You will need to replace the # with the actual number for the file you find. This technique will be explained later in the lab.
- Copy the contents of the file using nano into a file called hashes.txt in your /home/ubuntu folder for later decryption.

Record in Item #5 of your report Why are saving passwords on in a publicly accessible directory a bad thing?

Record in Item #6 of your report What if the passwords are hashed, does that make a difference?

4.7 Gaining Access to Server Data

The objective of this task is to see if the learner can access server logs or other metric data that the server generates. According to OWASP WSTG-CRYP-03, sensitive data must be protected when it is transmitted or even at rest. If data is transmitted over HTTPS or encrypted in another way the protection mechanism must not have limitations or vulnerabilities. If data is at rest, there needs to be a control in place to secure sensitive data. As a rule of thumb if data must be protected when it is stored, this data must also be protected during transmission. sensitive data is typically defined as passwords or authentication-based content, but it can also include server logs and other information used to compromise a system or service.

The first example below will have the learner look through some left-over information from an IT server team member to aid in identifying were log files may be. The second example will let the learner explore the site survey map and see if they can find metric location.

Example 1 – Gaining Access to Server Logs

- While completing the tasks above, and reviewing content in the ftp directory, a file titled "`incident-support.kdbx`" should be reviewed.
- Upon inspection of the file a directory should stand out, and that is the location the support team saves log files.
- You can also review task 4.5 site survey and determine if there is a support directory listed. View the contents of the following URL

  `https://192.168.99.100:3000/support/logs`

- Can you access the server logs? Save the access log to your computer.

Example 2 – Gaining Access to Server Metrics

- In task 4.5, a web site survey was performed, one directory of note that was found was the metric directory
- Analyze the directory and determine if this directory houses and serves usage data.

Record in Item #7 of your report Why is gaining access to support logs an issue?

Record in Item #8 of your report What is the purpose of a metric directory? Why is usage data of interest for someone trying to break a web server?

4.8 Discover of Backup files

The objective of this task is to identify a method for accessing one if not two possible backup files left over from the development or sales teams. According to OWASP WSTG-CRYP-03 and OWASP ASVS - Verification V9, there is a base level of security that should be applied on items that may contain sensitive data, such as backup files. In the example below we are going to try to identify and access one if not two backup files.

- In task 4.5, a web site survey was performed, one directory of note that was found was the ftp directory.
- Browse to https://192.168.99.100:3000/ftp
- Try to open https://192.168.99.100:3000/ftp/package.json.bak directly, does it work?
- You should note that directly access the MD file will result in an error.
- Using a *Poison Null Byte* (%00) the filter can be tricked, but only with a twist:
  - NOTE: accessing https://192.168.99.100:3000/ftp/package.json.bak%00.md will surprisingly **not** succeed...
  - ...because the % character needs to be URL-encoded (into %25) as well in order
- Navigate to https://192.168.99.100:3000/ftp/package.json.bak%2500.md
- Try to identify any other backup files that are present in the FTP folder.

Record in Item #9 of your report If a backup file is accessible, can someone take it and start a recovery from the backup files?

4.9 Accessing misplaced files

The objective of this task is to identify a method for accessing possible suspicious or other misplaced data files. According to OWASP ASVS - Verification V9, there is a base level of security that should be applied on items that may contain sensitive data, things such as signature files need to be secured and

stored properly. In the example below the learner will be exploring a suspicious file and seeing what data it may contain.

- In task 4.5, a web site survey was performed, one directory of note that was found was the ftp directory.
- Browse to https://192.168.99.100:3000/ftp
- Try to open https://192.168.99.100:3000/ftp/suspicious_errors.yml directly, does it work?
- You should note that directly access the MD file will result in an error.
- Navigate to the following URL:

  `https://192.168.99.100:3000/ftp/suspicious_errors.yml%2500.md`
- What is the nature of this file?

Record in Item #10 of your report Why does opening the error.yml file above directly not work? Why is the %2500 needed?

4.10 Retrieve Designs for a product

The objective of this task is to identify a method for accessing product blueprints that a user could retrieve. According to OWASP WSTG-CRYP-03 and OWASP ASVS - Verification V9, there is a base level of security that should be applied on items that may contain sensitive data. However, sometimes non-sensitive files can also be vulnerable. In this task the learner will assess product images that may also allow for 3D printing.

- In task 4.5 when the web site was scanned, look at projects that are "jpg" files and that also have exif data tags. See if you can find more than one.
- Navigate to the following URL:

  `https://192.168.99.100:3000/public/images/products/3d_keychain.jpg`

- View the Exif metadata from this file
- Researching the camera model entry OpenSCAD, what information do you find?
- What are the file types that OpenSCAD uses?
- Download the following file:

  `http://192.168.99.100:3000/public/images/products/JuiceShop.stl`
- Files that end with ".stl" are extensions that work in most 3D-print thus this allows the learner to take the file and make their own models!
- Can you review the web site scan specially the directory "`public/images/products`" for any other ".stl" files?

Record in Item #11 of your report What is the purpose of Exif metatags?

Record in Item #12 of your report Why is being able to access a .stl file an issue?

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 #13 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 14.

5 Stop the Labtainer

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

```
stoplab web-sde
```

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