WEB APP RECON

Laboratory Exercise 6.2 – Web App Recon

1. Overview

For this lesson, students will use the latest Cyber Range: Cyber Basics Environment to discover subdomains, directories, and files. Students will also install a vulnerable web application in Docker.

2. Resources required

This exercise requires the latest Cyber Basics Environment running in the Cyber Range.

3. Initial Setup

For this exercise, you will log in to your Cyber Range account and select the latest Cyber Basics Environment, then click "start" to start your environment and "join" to get to your Linux desktop login. Log in using these credentials:

Username: **`student`**
Password: **`student`**

4. Tasks

Task 1: Finding Subdomains with Assetfinder

Assetfinder is a great tool that parses DNS information from other websites. This makes it fast but, not always accurate. Complete the following:

- In the root terminal, **`gedit ~/.bashrc`**
- Add the following to the bottom of the .bashrc file:

export GOPATH=/root/go
export PATH=$PATH:/root/go/bin
export GO111MODULE=on

```
172 export GOPATH=/root/go
173 export PATH=$PATH:/root/go/bin
174 export GO111MODULE=on
175
176
```

- Save and exit the .bashrc file.
- Create a directory on the student Desktop and name it WebApp.
- In a *new* root terminal, navigate to the WebApp directory.
- Type **`assetfinder virginiacyberrange.org`** and press enter.

```
Mahesh@/student$:assetfinder virginiacyberrange.org
virginiacyberrange.org
analytics.virginiacyberrange.org
artifacts.virginiacyberrange.org
ctf.virginiacyberrange.org
cyberfusion.virginiacyberrange.org
kb.virginiacyberrange.org
qa.kb.virginiacyberrange.org
openchallenges.virginiacyberrange.org
piwik.virginiacyberrange.org
ctf.poc.virginiacyberrange.org
api.qa.virginiacyberrange.org
ctf.qa.virginiacyberrange.org
www.virginiacyberrange.org
qa.www.virginiacyberrange.org
virginiacyberrange.net
login.virginiacyberrange.org
cyberfusion.ctf.virginiacyberrange.org
9bde87f5-0106-494e-88f8-e1f6855692fd.ctf.virginiacyberrange.org
aklab3.com
vacr.io
range.academiesctf.com
```

As you can see, assetfinder will list all the subdomains found. Don't worry if you did not get the same subdomains as this is normal. In many cases, this list will be very large. The best way to handle this is to output the information to a file.

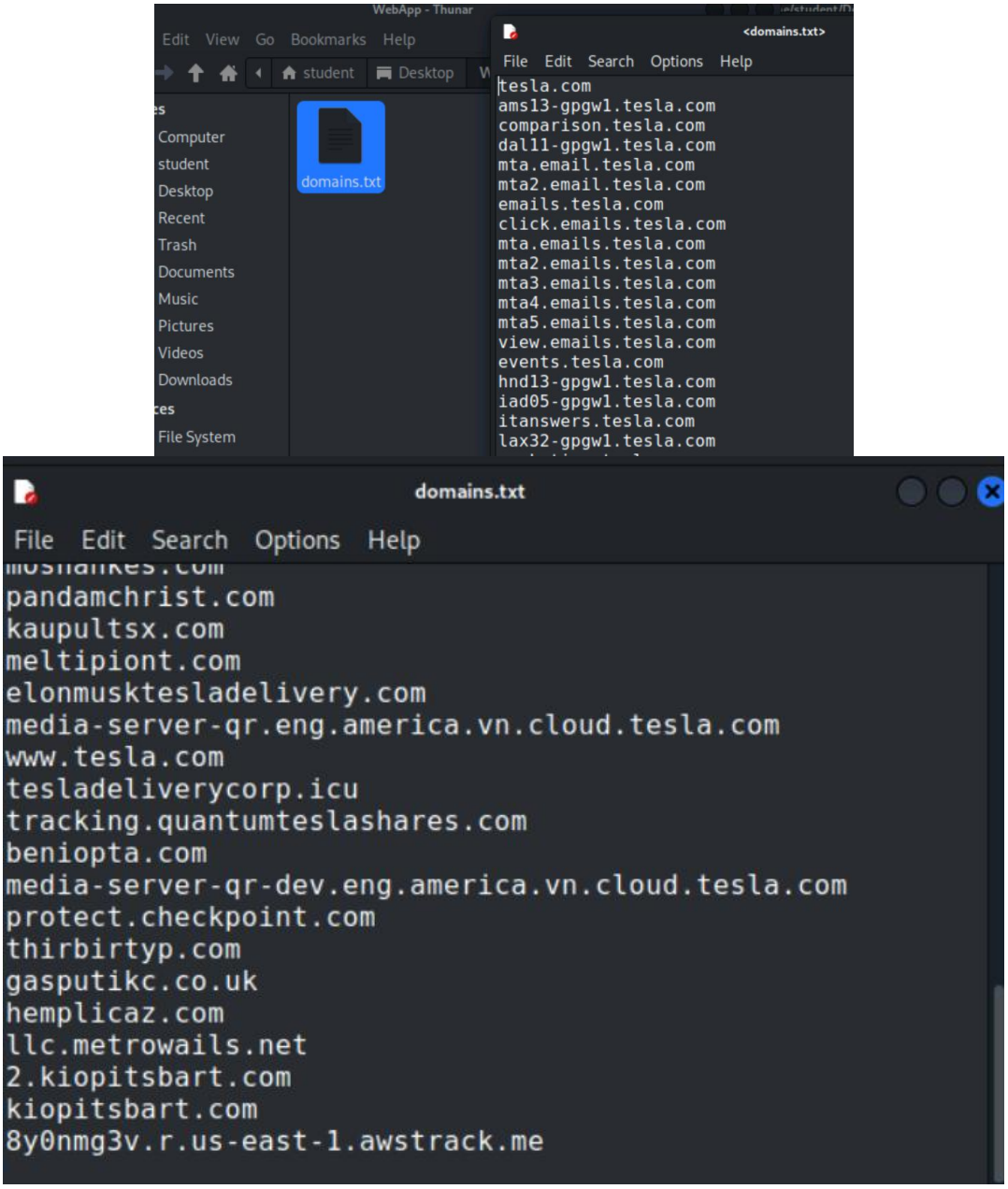
- In a root terminal, type:

  `assetfinder tesla.com > /home/student/Desktop/WebApp/domains.txt`
  and press enter.

```
Mahesh@/WebApp$:assetfinder tesla.com > /home/student/Desktop/WebApp/domains.txt
```

- Navigate to the WebApp directory and open the domains.txt file in a text editor. Here you will see the list of domains that assetfinder discovered.

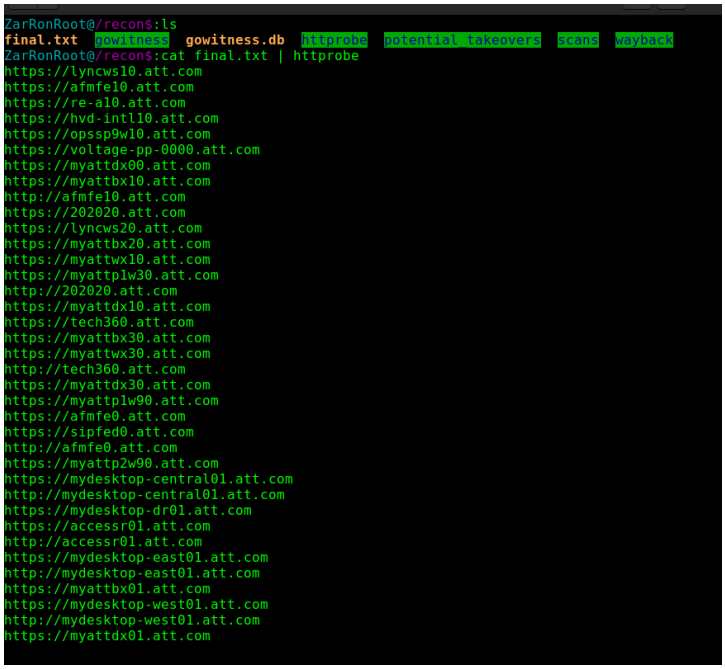
WEB APP RECON



The next step is to check to see what domains are "live." This means you can connect to them through an http request. This is most commonly done in the browser. A great tool for this is httprobe; however, it does not function in the Cyber Range. **I am providing the syntax so that you may try this outside of the Cyber Range.** There is also an example screenshot below.

- In a root terminal, type **`cat <location of file> | httprobe`**
- In our case, this would look like the following:

  `cat /home/student/Desktop/WebApp/domains.txt | httprobe`



Once the live subdomains have been discovered, we can check each individually. Instead, we will copy and paste the domains and save them as domains2.txt. Gowitness takes a screenshot of each domain and stores it in a file. We can navigate to the file and then scroll through the images to determine if any look interesting enough to test further. This feature does not function in the Cyber Range. **I am providing the syntax so that you may try this on your own outside of the Cyber Range.**

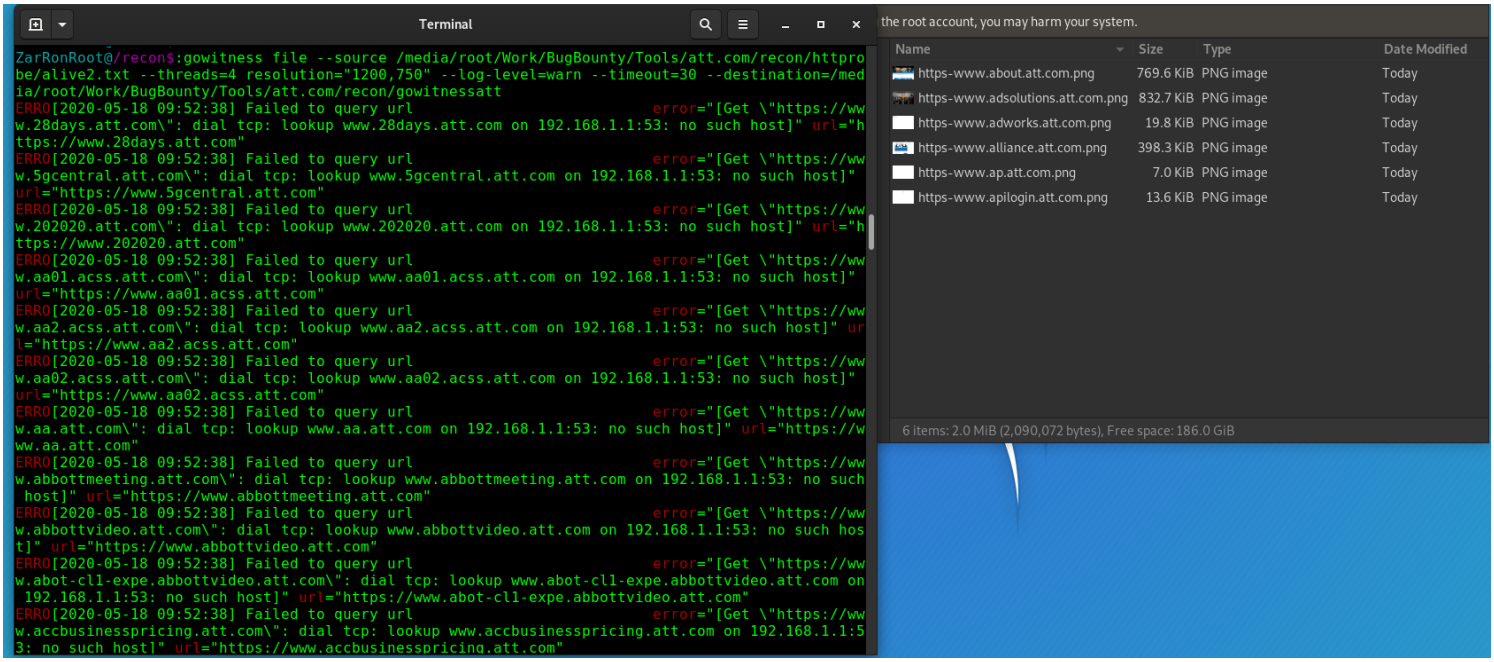
- In a root terminal type:

  gowitness file --source=<urls.txt> --threads=4 --resolution="1200,750" --log-level=warn --timeout=30 --destination=<save location>

- In our case, this would look like the following :

  gowitness file --source=/home/student/Desktop/WebApp/domains2.txt --threads=4 --resolution="1200,750" --log-level=warn --timeout=30 --destination=/home/student/Desktop/WebApp/

This screenshot is an example showing the terminal and the folder that the images will populate



Task 2: Installing a Docker Image

Docker is a container software that allows us to host a simple web application in the VM. We will create a container for a web application called Juice Shop. Juice Shop is a purposefully vulnerable web application, designed for educational purposes.

To install Docker complete the following:

- Navigate to the opt folder. Then in a root terminal, install the PGP key by typing:

  `curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -` and press enter.

- Install the apt to help with installation to Kali by typing:

  `echo 'deb [arch=amd64] https://download.docker.com/linux/debian buster stable' | sudo tee /etc/apt/sources.list.d/docker.list` and press enter.

- Update your Kali VM. (**NOTE**: Remember, we created an alias for this in the last module.)

- Install Docker by typing:

  `apt install docker-ce` and press enter. (**NOTE**: Type "y" at the appropriate time.)

- In most situations you will need to run Docker by typing the following:

dockerd and press enter. This terminal will be left running and a new root terminal will be opened. In our case, Docker will run after installation in the background. However, if you leave the VM or the VM times out, the above command and process will get Docker up and running again.

The Docker software is now running. *We will complete the last command each time we log in to run the docker.* Although Docker is capable of running on startup, we do not want the system to become unstable.

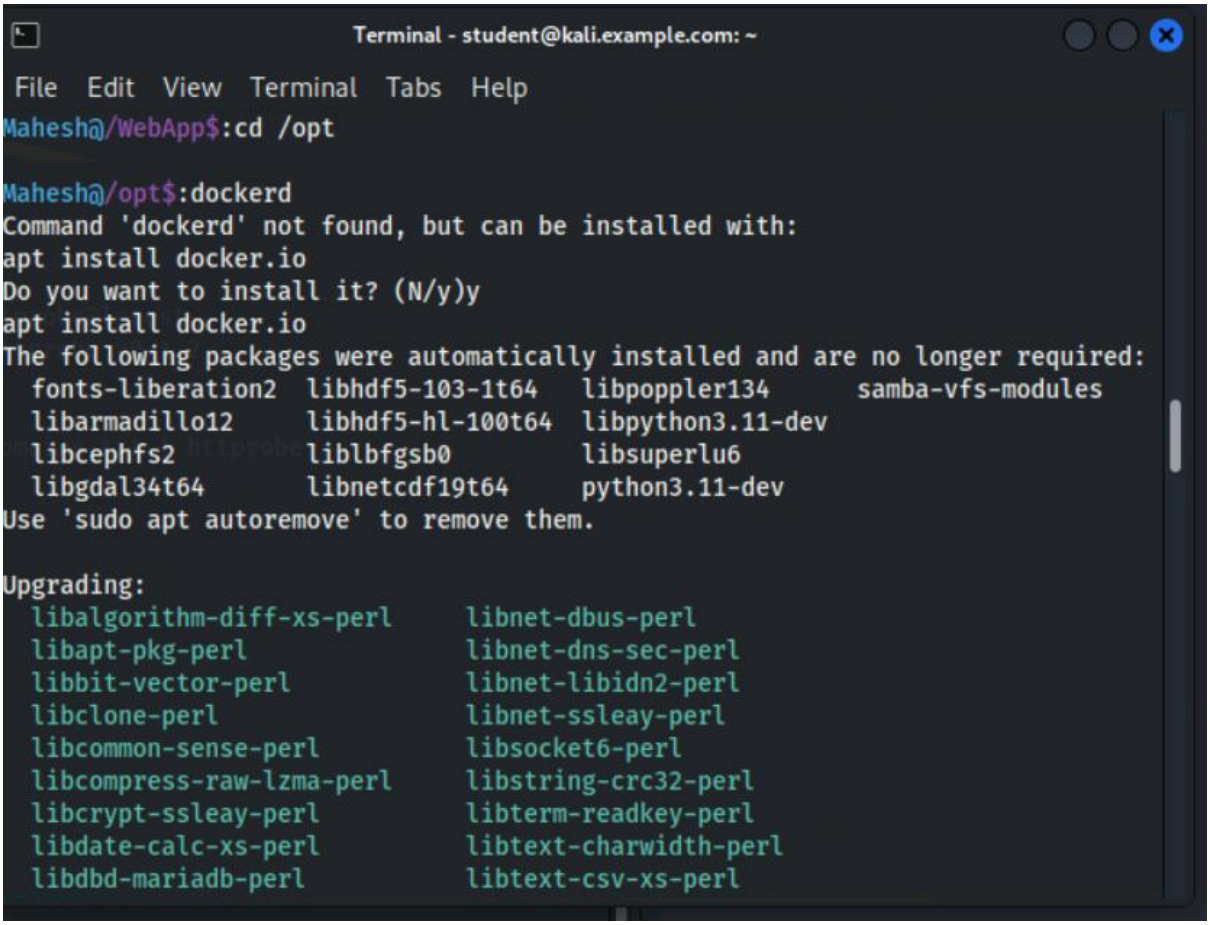
Task 3: Installing Juice Shop

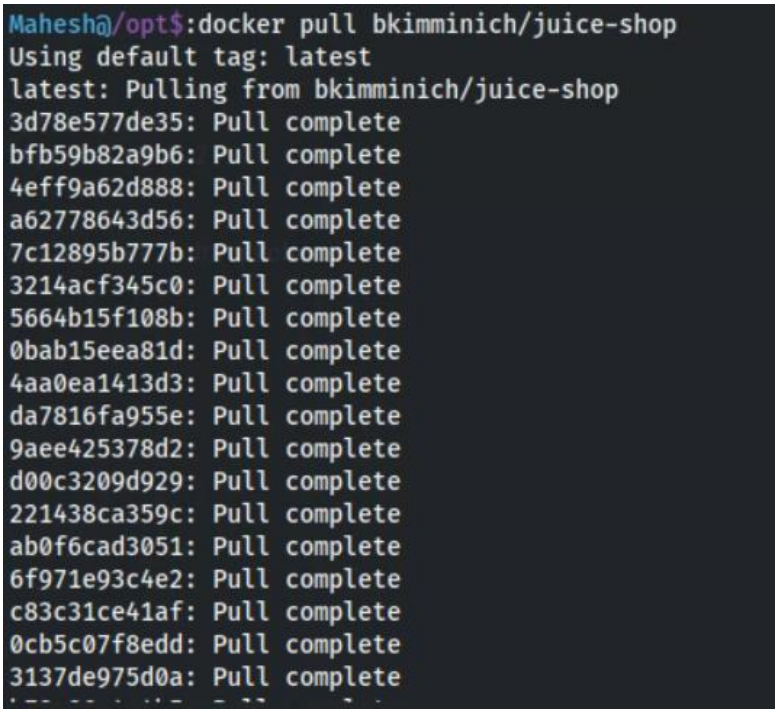
We will pull the docker from a GitHub repository to install Juice Shop.

Complete the following:

- In the opt folder and a root terminal, type

  cddocker pull bkimminich/juice-shop and press enter. This will download the files and place them in the right location.



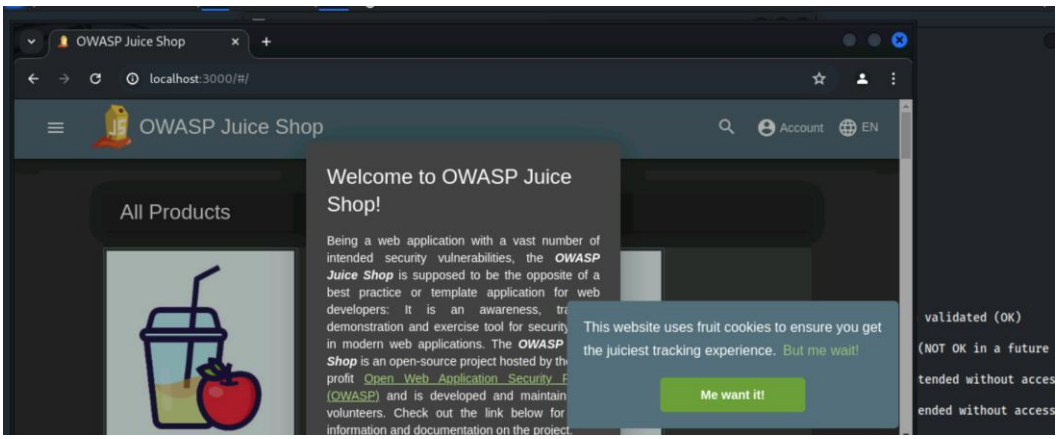


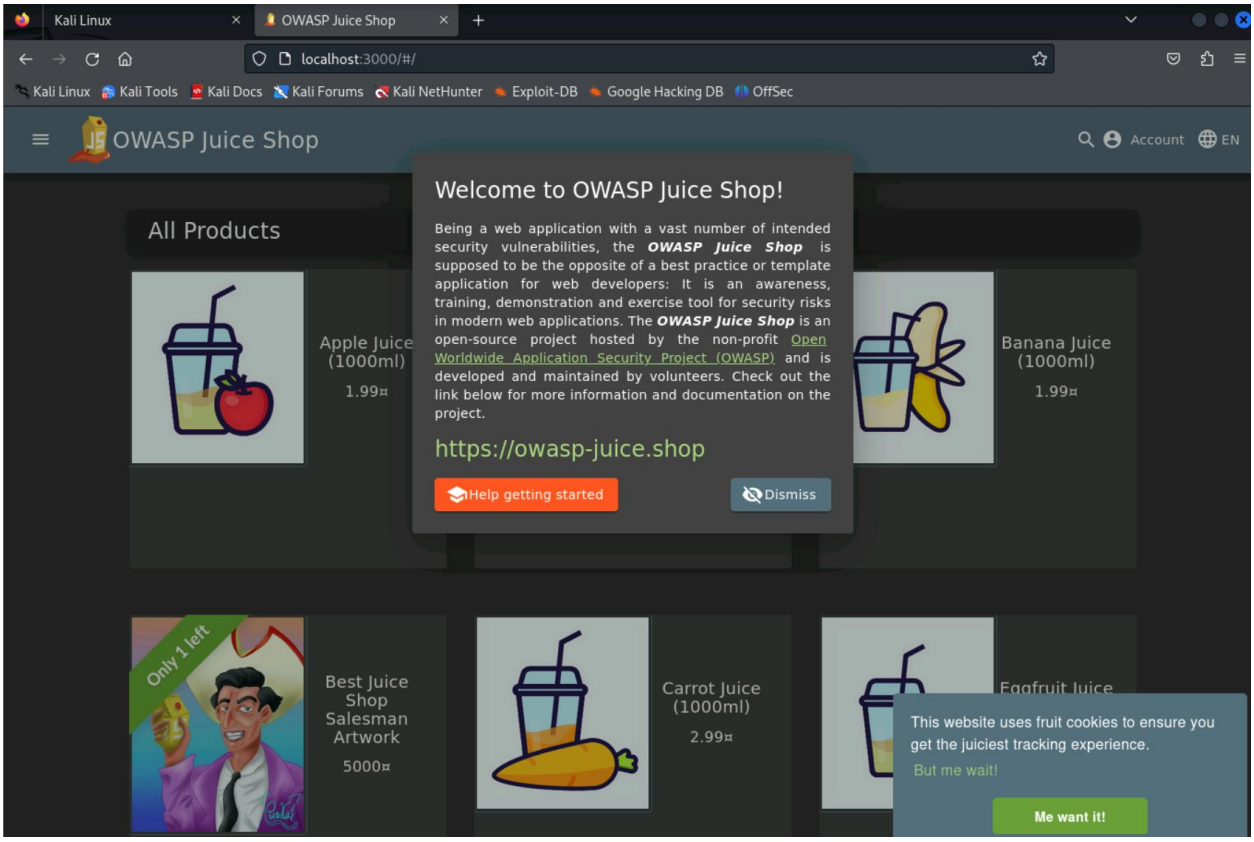
In the terminal, type

`docker run --rm -p 3000:3000 bkimminich/juice-shop` and press enter.

```
Mahesh@/opt$:docker run --rm -p 3000:3000 bkimminich/juice-shop
info: Detected Node.js version v20.19.0 (OK)
info: Detected OS linux (OK)
info: Detected CPU x64 (OK)
info: Configuration default validated (OK)
info: Entity models 19 of 19 are initialized (OK)
info: Required file server.js is present (OK)
info: Required file index.html is present (OK)
info: Required file styles.css is present (OK)
info: Required file main.js is present (OK)
info: Required file tutorial.js is present (OK)
info: Required file runtime.js is present (OK)
info: Required file vendor.js is present (OK)
```

- Open Google Chrome and navigate to http://localhost:3000 and you will find the web application.





Task 4: Finding Web Application Directories

In the last lesson, we installed Dirsearch. This tool uses a word list to brute force web directories. The wordlist can be modified or changed; however, for this task, we will use the default directory. Dirsearch can find files if extensions are specified. Another nice feature is that it can brute force recursively. This means that if a directory is found, the program will attempt to brute force that directory.

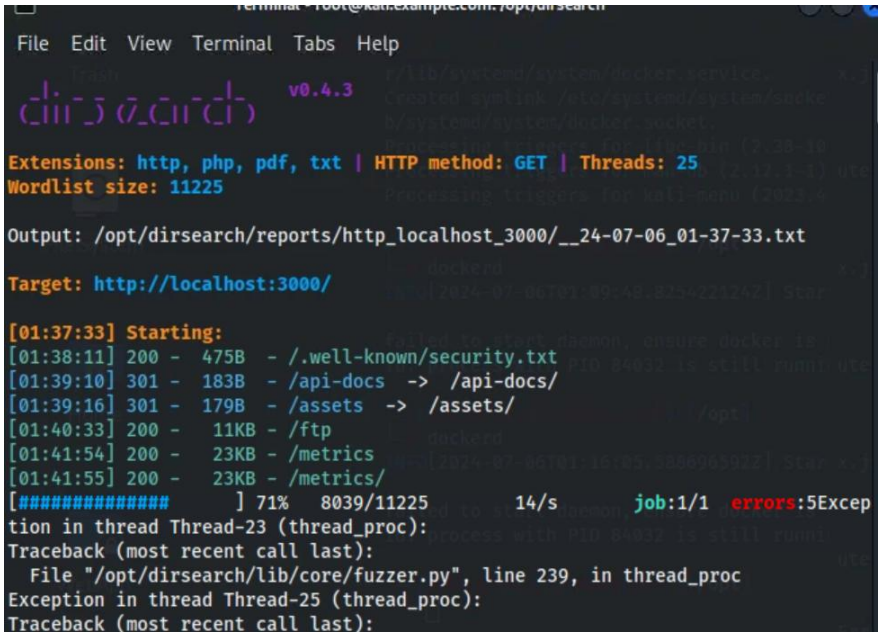
Complete the following:

- In a root terminal type

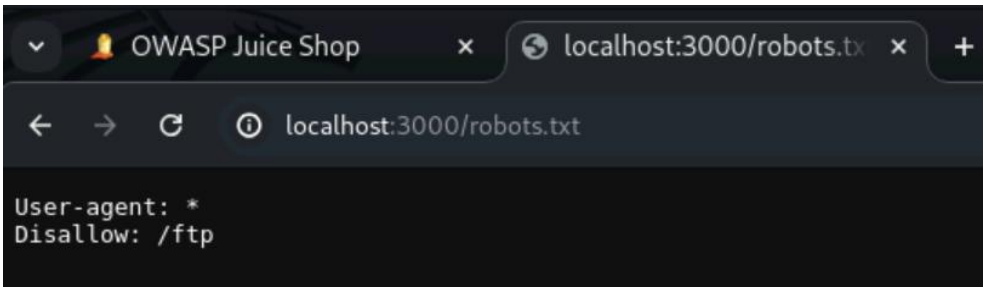
  ```
  dirsearch http://localhost:3000 -e http,php,pdf,txt
  ```

  and press enter

Dirsearch will list all the discovered directories and files in the vulnerable web application running at http://localhost:3000/ (Juice Shop).



We can look at these directories and files by appending them to the site in the browser. You can see what happens when we append /robots.txt or /profile to the URL of this web app in the screenshots below. Now we know a little more about this vulnerable web app.



Dirb is a program built into Kali that will also brute force directories. Sometimes it is good to use more than one tool or wordlist to discover more data.

WEB APP RECON

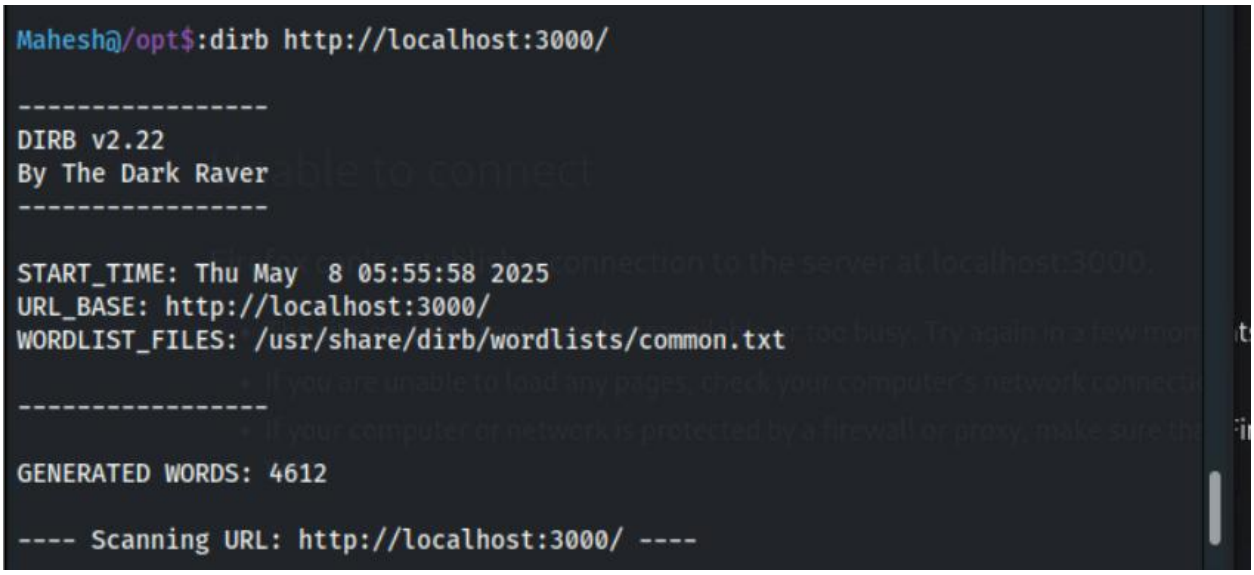
Complete the following

- In a root terminal, type:

```
dirb http://localhost:3000/
```

As you can see, Dirsearch found more data; however, Dirb found /Video.

Note* *If you get a "could not connect" error restart the Juice Shop on port 3000.*



Challenge (Optional but recommended):

Complete the techniques in task 4 on the DVWA web application located at http://dvwa.example.com. Save your findings in the WebApp Folder. Remember, you can always use standard output to save what otherwise is printed to the terminal.

In this lesson, you learned how to install a web application in a Docker container, and how to find subdomains and directories in that web application. In the next lesson, we will learn how to manually perform reconnaissance using BurpSuite.