Docker Composed
Deployment of a Compromised Web Server

Overview

This report outlines the process of deploying a compromised web server using Docker Compose, as captured in the command history. The task involved creating a directory for the project, navigating into it, and setting up a Docker Compose file to define the services required for the web server.

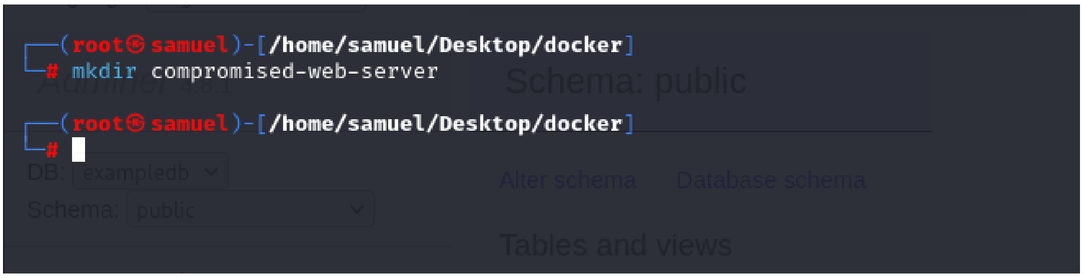
Objective:

Creating a Docker Compose file for educational purposes is okay, but it's important to clarify that setting up intentionally vulnerable or "compromised" systems should be done only within a secure and controlled environment, typically for learning or security training purposes. Ensure that you follow legal and ethical guidelines and that the setup is isolated from public networks to prevent security breaches.

Task Details

1. Directory Creation: The task began with the creation of a directory named `compromised-web-server` to house the project files. This was achieved using the `mkdir` command.

```
mkdir compromised-web-server
```



```
┌──(root㉿samuel)-[/home/samuel/Desktop/docker]
└─# mkdir compromised-web-server

┌──(root㉿samuel)-[/home/samuel/Desktop/docker]
└─# 
```

2. Navigation: The user then navigated into the newly created directory using the `cd` command.

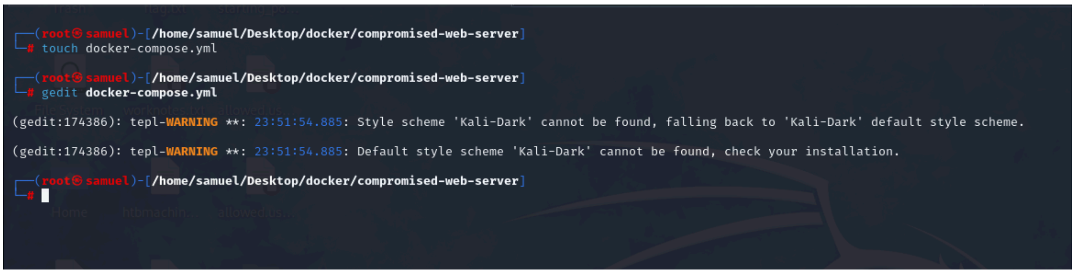
```
cd compromised-web-server
```

3. Docker Compose File Creation: A Docker Compose file named \`docker-compose.yml\` was created using the \`touch\` command. This file is crucial for defining the services that make up the web server.

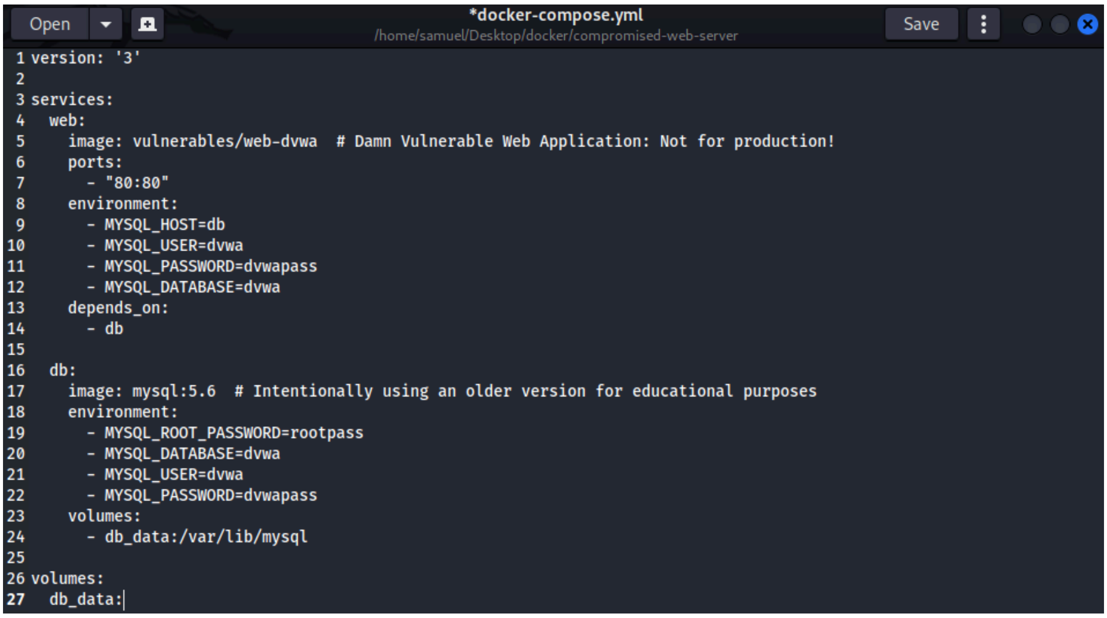
```
touch docker-compose.yml
```

4. Editing the Docker Compose File: The user opened the \`docker-compose.yml\` file in a text editor (\`gedit\`) to begin configuring the services.

```
gedit docker-compose.yml
```



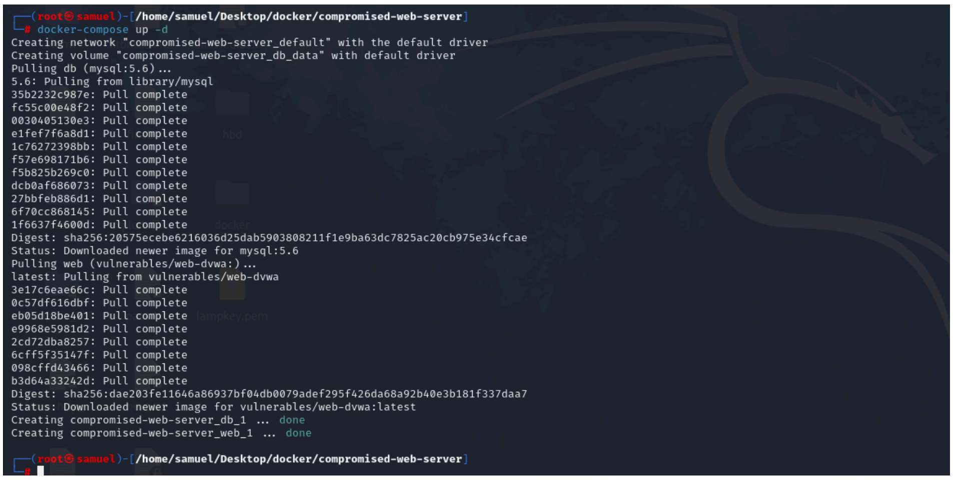
```
┌──(root㉿samuel)-[/home/samuel/Desktop/docker/compromised-web-server]
└─# touch docker-compose.yml

┌──(root㉿samuel)-[/home/samuel/Desktop/docker/compromised-web-server]
└─# gedit docker-compose.yml

(gedit:174386): tepl-WARNING **: 23:51:54.885: Style scheme 'Kali-Dark' cannot be found, falling back to 'Kali-Dark' default style scheme.

(gedit:174386): tepl-WARNING **: 23:51:54.885: Default style scheme 'Kali-Dark' cannot be found, check your installation.

┌──(root㉿samuel)-[/home/samuel/Desktop/docker/compromised-web-server]
└─# ▌
```



```
version: '3'

services:
  web:
    image: vulnerables/web-dvwa  # Damn Vulnerable Web Application: Not for production!
    ports:
      - "80:80"
    environment:
      - MYSQL_HOST=db
      - MYSQL_USER=dvwa
      - MYSQL_PASSWORD=dvwapass
      - MYSQL_DATABASE=dvwa
    depends_on:
      - db

  db:
    image: mysql:5.6  # Intentionally using an older version for educational purposes
    environment:
      - MYSQL_ROOT_PASSWORD=rootpass
      - MYSQL_DATABASE=dvwa
      - MYSQL_USER=dvwa
      - MYSQL_PASSWORD=dvwapass
    volumes:
      - db_data:/var/lib/mysql

volumes:
  db_data:
```

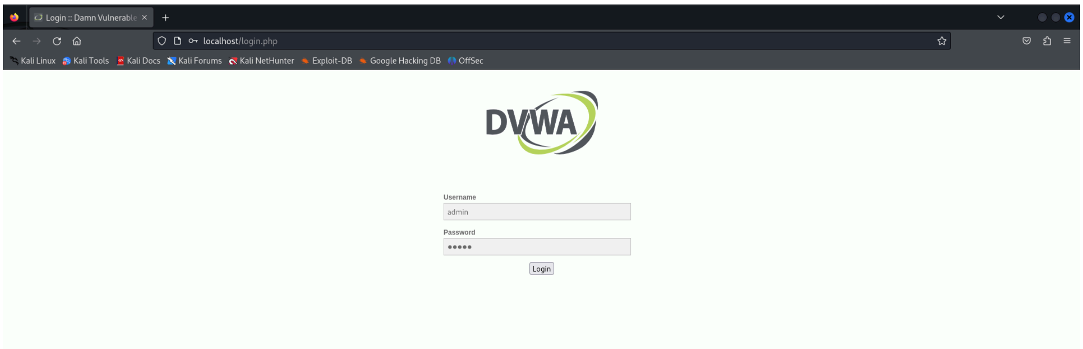
5. Deployment: Finally, the Docker Compose environment was deployed using the \`docker-compose up -d\` command. This command starts the services defined in the \`docker-compose.yml\` file in detached mode, allowing the user to continue using the terminal.

```
docker-compose up -d
```

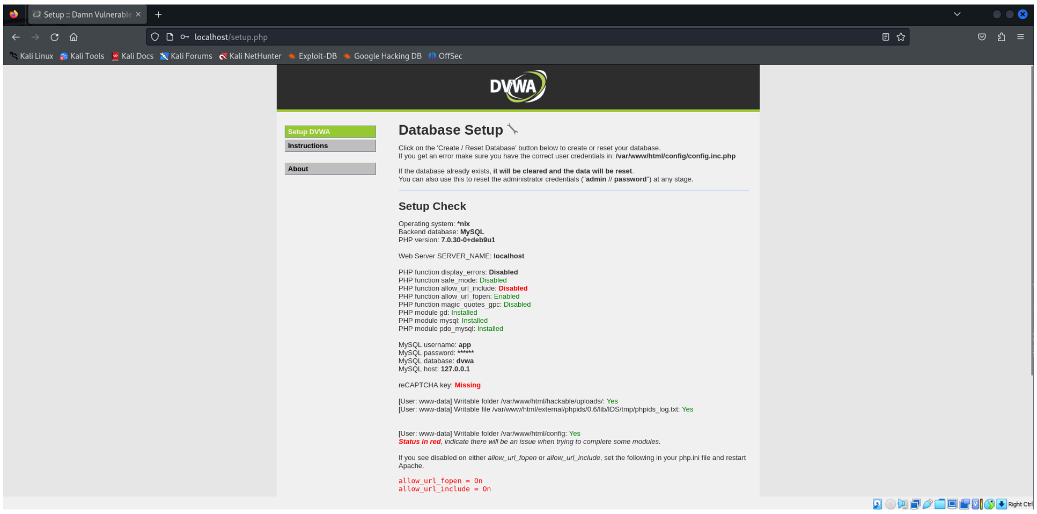


```
┌──(root㉿samuel)-[/home/samuel/Desktop/docker/compromised-web-server]
└─# docker-compose up -d
Creating network "compromised-web-server_default" with the default driver
Creating volume "compromised-web-server_db_data" with default driver
Pulling db (mysql:5.6)...
5.6: Pulling from library/mysql
35b2232c987e: Pull complete
fc55c00e48f2: Pull complete
0030405130e3: Pull complete
e1fef7f6a8d1: Pull complete
1c76272398bb: Pull complete
f57e698171b6: Pull complete
f5b825b269c0: Pull complete
dcb0af686073: Pull complete
27bbfeb886d1: Pull complete
6f70cc868145: Pull complete
1f6637f4600d: Pull complete
Digest: sha256:20575ecebe6216036d25dab5903808211f1e9ba63dc7825ac20cb975e34cfcae
Status: Downloaded newer image for mysql:5.6
Pulling web (vulnerables/web-dvwa:)...
latest: Pulling from vulnerables/web-dvwa
3e17c6eae66c: Pull complete
0c57df616dbf: Pull complete
eb05d18be401: Pull complete
e9968e5981d2: Pull complete
2cd72dba8257: Pull complete
6cff5f35147f: Pull complete
098cffd43466: Pull complete
b3d64a33242d: Pull complete
Digest: sha256:dae203fe11646a86937bf04db0079adef295f426da68a92b40e3b181f337daa7
Status: Downloaded newer image for vulnerables/web-dvwa:latest
Creating compromised-web-server_db_1  ... done
Creating compromised-web-server_web_1 ... done

┌──(root㉿samuel)-[/home/samuel/Desktop/docker/compromised-web-server]
└─# 
```

6. Accessing Your Web Server: Open your preferred web browser and go to http://localhost:8080. You should see your HTML page served by the Nginx web server.



Username
admin
Password
•••••
Login



Database Setup

Click on the 'Create / Reset Database' button below to create or reset your database.
If you get an error make sure you have the correct user credentials in: **/var/www/html/config/config.inc.php**

If the database already exists, **it will be cleared and the data will be reset**.
You can also use this to reset the administrator credentials ("**admin // password**") at any stage.

Setup Check

Operating system: ***nix**
Backend database: **MySQL**
PHP version: **7.0.30-0+deb9u1**

Web Server SERVER_NAME: **localhost**

PHP function display_errors: **Disabled**
PHP function safe_mode: Disabled
PHP function allow_url_include: **Disabled**
PHP function allow_url_fopen: Enabled
PHP function magic_quotes_gpc: Disabled
PHP module gd: Installed
PHP module mysql: Installed
PHP module pdo_mysql: Installed

MySQL username: **app**
MySQL password: **********
MySQL database: **dvwa**
MySQL host: **127.0.0.1**

reCAPTCHA key: **Missing**

[User: www-data] Writable folder /var/www/html/hackable/uploads/: Yes
[User: www-data] Writable file /var/www/html/external/phpids/0.6/lib/IDS/tmp/phpids_log.txt: Yes

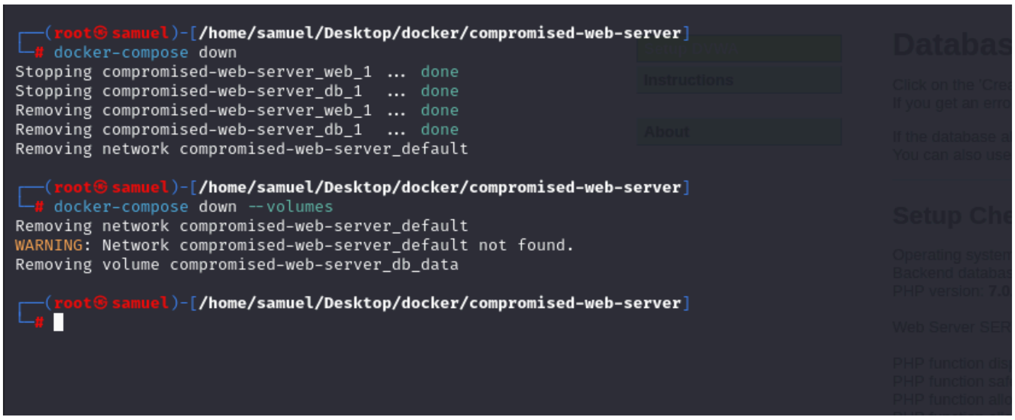
[User: www-data] Writable folder /var/www/html/config: Yes
Status in red, *indicate there will be an issue when trying to complete some modules.*

If you see disabled on either *allow_url_fopen* or *allow_url_include*, set the following in your php.ini file and restart Apache.

```
allow_url_fopen = On
allow_url_include = On
```

7. Stopping the Service: When you're finished, stop your Docker Compose service by running:

```
docker-compose down
```



```
┌──(root㉿samuel)-[/home/samuel/Desktop/docker/compromised-web-server]
└─# docker-compose down
Stopping compromised-web-server_web_1  ... done
Stopping compromised-web-server_db_1   ... done
Removing compromised-web-server_web_1  ... done
Removing compromised-web-server_db_1   ... done
Removing network compromised-web-server_default

┌──(root㉿samuel)-[/home/samuel/Desktop/docker/compromised-web-server]
└─# docker-compose down --volumes
Removing network compromised-web-server_default
WARNING: Network compromised-web-server_default not found.
Removing volume compromised-web-server_db_data

┌──(root㉿samuel)-[/home/samuel/Desktop/docker/compromised-web-server]
└─# 
```

Conclusion

The task successfully demonstrated the process of setting up a compromised web server using Docker Compose. This includes creating the necessary directory structure, defining the services in a Docker Compose file, and deploying the environment. The deployment was executed in detached mode, allowing for the server to run in the background. This setup is a common practice in development and testing environments, where the focus is on the functionality and integration of the services rather than their security.