**Web App Sec 101**

I had chosen to do **Web App Sec 101** as my project of my Advanced Cyber Security Principles and Tools.

**Source** : Try hack me

A screenshot of a computer

Description automatically generated

**Task 1  Basic Description & Objectives**

* Initially I had started the Run Machine.
* To deploy the machine, I had connected to the VPN.
* I had used kali Linux to complete this project.

A computer screen capture

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with medium confidence

A computer screen capture

Description automatically generated with low confidence

A computer screen capture

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with medium confidence

A picture containing text, monitor, screenshot, computer

Description automatically generated

**Task 2 Walking through the application**

**A picture containing text, monitor, screenshot, computer

Description automatically generated**

* What version of Apache is being used?

2.4.7

* What language was used to create the website?

PHP

* What version of this language is used?

5.5.9

**Task 3  Establishing a methodology**

Now that we have walked through the application and know the functionality and technologies, how do we test it. There are 2 ways to test a web application for security vulnerabilities:

The first way is by going through every page and testing all the functionality. This would involve going through every page on the application, and depending on the functionality, testing for all the bugs/vulnerabilities that apply to the page. In this case, if we start off at the home page, we will try see what functionality we can exploit on the home page, and then move on to every page.

The second way is by breaking down the testing into different stages(including but not limited to):

1. **Authorization**

**Pros**

* Implementation is simple, as there is no encryption involved.
* Take relatively less time to respond as it has only one call.
* The lack of token creation and encryption method gives Client an advantage of using less code to call the API.
* The information is retrieved from the server with just one call, making it faster than other complex authentications.

**Cons**

* SSL takes time to run basic HTTP, so this will make the response time considerably slow.
* The lack of encryption makes the security risk high.

1. Authentication

Pros

* No need to worry about password thefts.
* It protects against brute force attacks.

Cons

* Users Are Hesitant About Trusting Password less Technology.

1. Injection
2. Client-Side Controls

Pros

* Execute quickly because they do not require a trip to the server.
* May improve the usability of Web sites for users whose browsers support scripts.

Cons

* More development time and effort might be required (if the scripts are not already available through other resources).

1. Application Logic

Pros

* Easy to use.

Cons

* Security issues.

**Task 4  Authentication**

Let us try guessing admin username and password. There’ re 4 combinations that I can think of:  
admin: admin  
admin: password  
root: root  
root: password.

A picture containing text, screenshot, monitor, screen

Description automatically generated

Admin, Admin worked

* What is the admin username?

Admin

* What is the admin password?

Admin

Graphical user interface, application

Description automatically generated

* What is the name of the cookie that can be manipulated?

Session

* What is the username of a logged-on user?

Bryce

A picture containing text, screenshot, monitor, indoor

Description automatically generated

* What is the corresponding password to the username?

Bryce

**Authentication**

* 5.1 Test for XSS on the search bar

Inject <script>alert('xss')</script> in the search. It will confirm that the search field is not sanitizing the user input and is vulnerable to non-persistent XSS attack.

* 5.2 Test for XSS on the guestbook page

The comment field of the guestbook is vulnerable to persistent XSS. If you enter <script>alert('xss')</script> in the comment field, it will save it to the database. Each time a user is going to the page will execute the script.ss

**Finally, I done my Web App Sec 101 try hack me room with 100%**

Graphical user interface

Description automatically generated

**My Try Hack Me Completed Card :**

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, website

Description automatically generated

**Screen shot of my tryhackme profile :**

**Graphical user interface, text, application

Description automatically generated**