ETHICAL HACKING TRAINING

PROBLEM STATEMENT

We are glad that you have completed the training and cleared the final test. Now, it’s time to test your skills in a practical manner and for that, we have setup a real life-like web application in the form of an online e-commerce portal.

Your task is to test this e-commerce platform and find all possible vulnerabilities and loopholes in it, collect relevant PoCs and then prepare a Detailed Developer Level Report.

For reporting each vulnerability, you must follow the sample report given to you in Module 8 and make sure the following things are mentioned:

- Title of Vulnerability.
- A Short Description.
- Exact URL which has the vulnerability.
- The parameters which are vulnerable (with parameter type like GET, POST, Cookie, Header, etc.).
- Payload that you used to trigger the vulnerability.
- Observation slides containing step by step information to replicate the exploit with PoCs.
- Business Impact of the vulnerability, explaining in detail what can be done by a hacker.
- Recommendations on how to fix the vulnerability.
- Reputed References for the vulnerabilities.

Remember, each and every kind of vulnerability you learnt about, might be somewhere in this application. All you have to do is open the application and start exploring its features. Once you have understood each feature the website has, you can start playing around with it and fuzzing into various places.

A big part of the VA has been already done for you as you have the exact IP and the application which you have to test, but there could be hidden pages and components too, so keep that in mind.

To give you a benchmark and a target to achieve, here is a list of all the vulnerabilities which we have intentionally kept and which are supposed to be found and reported by you:

SQL Injection
Reflected and Stored Cross Site Scripting
Insecure Direct Object Reference
Rate Limiting Issues
Insecure File Uploads
Client Side Filter Bypass
Server Misconfigurations
Components with Known Vulnerabilities
Weak Passwords
Default Files and Pages
File Inclusion Vulnerabilities
PII Leakage
Open Redirection
Bruteforce Exploitation
Command Execution Vulnerability
Forced Browsing Flaws
Cross-Site Request Forgery

So, there are a total of 28 vulnerabilities (some vulnerabilities have more occurences than 1) intentionally kept but these do not include combinational vulnerabilities like Bruteforce Exploitation and Rate Limiting. If you are able to guess the password, you can either count it in Bruteforcing or count it in rate limiting but yes, while writing recommendations, write recommendations for both. Similarly, if you find a public software that allows PHP file upload, you can either count it in file upload or in Components with known vulnerabilities.

If you do find other general vulnerabilities apart from these you can report them too but do not count them in the 28.

Happy bug hunting!

Steps to access the Project:

1. Login to trainings.internshala.com
2. Go to Ethical Hacking Training
3. Go to Progress Tracker
4. Click on the 'GO TO PROJECT WEB APPLICATION' button.