VPLE

Vulnerable Pentesting Lab Environment

This is VPEL (Linux)

VPLE is an intentionally vulnerable Linux virtual machine. This VM can be used to conduct security training, test security tools, and practice common penetration testing Labs. In VPLE bunch of labs Available. (only run in VMWare Pls Don’t run in VirtualBox)

The default login and password is **administrator:password.**

Power on the VPLE:-

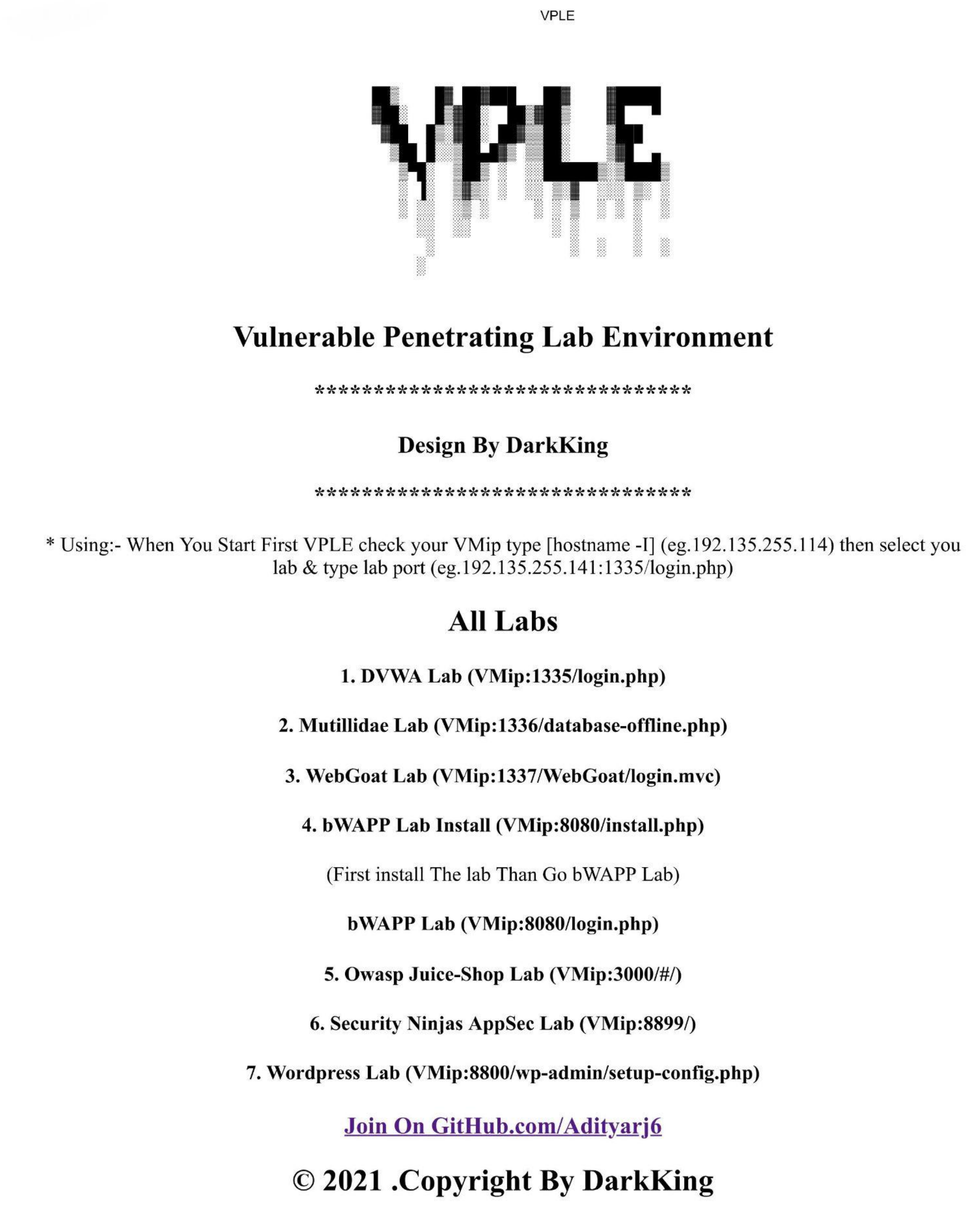
Once the VM is available on your desktop, open the device, and run it with VMWare Player. Alternatively, you can also use VMWare Workstation or VMWare Server.

Login in the VPLE:-

The login for VPLE is `administrator:password`.

Identifying IP Address of the VPLE:-

After you login to VPLE, you can identify the IP address that has been assigned to the virtual machine. Just enter `hostname -I` at the prompt to see the details for the virtual machine.



VPLE

Vulnerable Penetrating Lab Environment

****

Design By DarkKing

****

* Using:- When You Start First VPLE check your VMip type [hostname -I] (eg.192.135.255.114) then select you lab & type lab port (eg.192.135.255.141:1335/login.php)

All Labs

1. DVWA Lab (VMip:1335/login.php)

2. Mutillidae Lab (VMip:1336/database-offline.php)

3. WebGoat Lab (VMip:1337/WebGoat/login.mvc)

4. bWAPP Lab Install (VMip:8080/install.php)

(First install The lab Than Go bWAPP Lab)

bWAPP Lab (VMip:8080/login.php)

5. Owasp Juice-Shop Lab (VMip:3000/#/)

6. Security Ninjas AppSec Lab (VMip:8899/)

7. Wordpress Lab (VMip:8800/wp-admin/setup-config.php)

Join On GitHub.com/Adityarj6

© 2021 .Copyright By DarkKing

How To Use

Getting Started

After the virtual machine boots, login to the console with username `administrator` and password `password`. From the shell, run the `hostname -I` command to identify the IP address.

Vulnerable Web Services

VPLE has deliberately vulnerable web applications pre-installed. The web server starts automatically when VPLE is booted. To access the web applications, open a web browser and enter the URL where `<IP>` is the IP address of VPLE.

In this example, VPLE is running at IP 192.168.255.143. Browsing to http://192.168.255.143/ shows the web application home page.

To access a particular web application, just type IP address then enter the port no of Particular web application lab

List Of All Labs:-

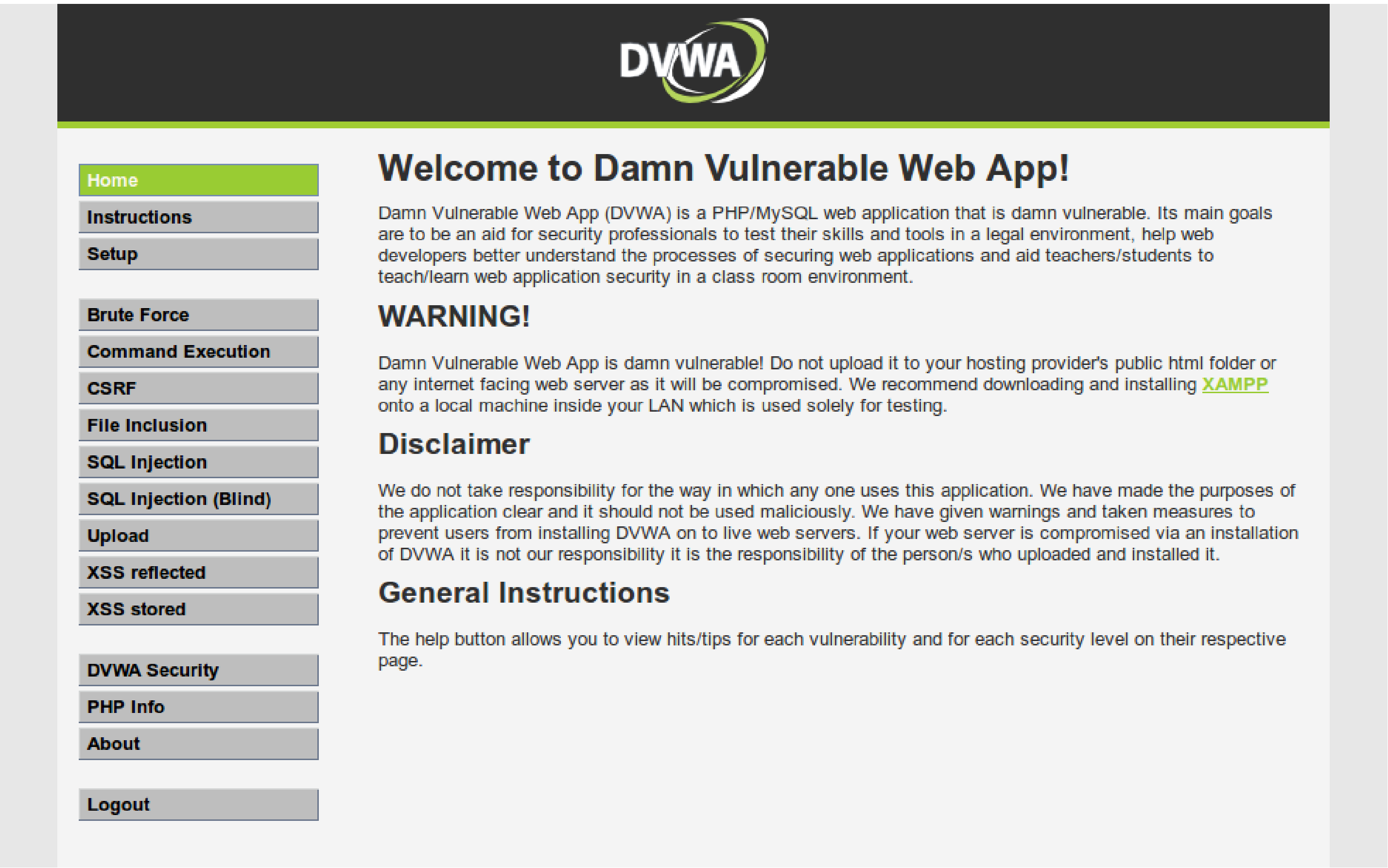
- Web-dvwa (eg.192.168.255.143:1335/)
- Mutillidae (eg.192.168.255.143:1336/)
- Webgoat (eg.192.168.255.143:1337/WebGoat/)

- ➢ Bwapp (eg.192.168.255.143:8080/install.php & 192.168.255.143:8080/install.php)
- ➢ Juice-shop (eg.192.168.255.143:3000/)
- ➢ Security-ninjas (eg.192.168.255.143:8899/)
- ➢ Wordpress (eg.192.168.255.143:8800/)

All Labs Details:-

DVWA

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a classroom environment.



Welcome to Damn Vulnerable Web App!

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

WARNING!

Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing XAMPP onto a local machine inside your LAN which is used solely for testing.

Disclaimer

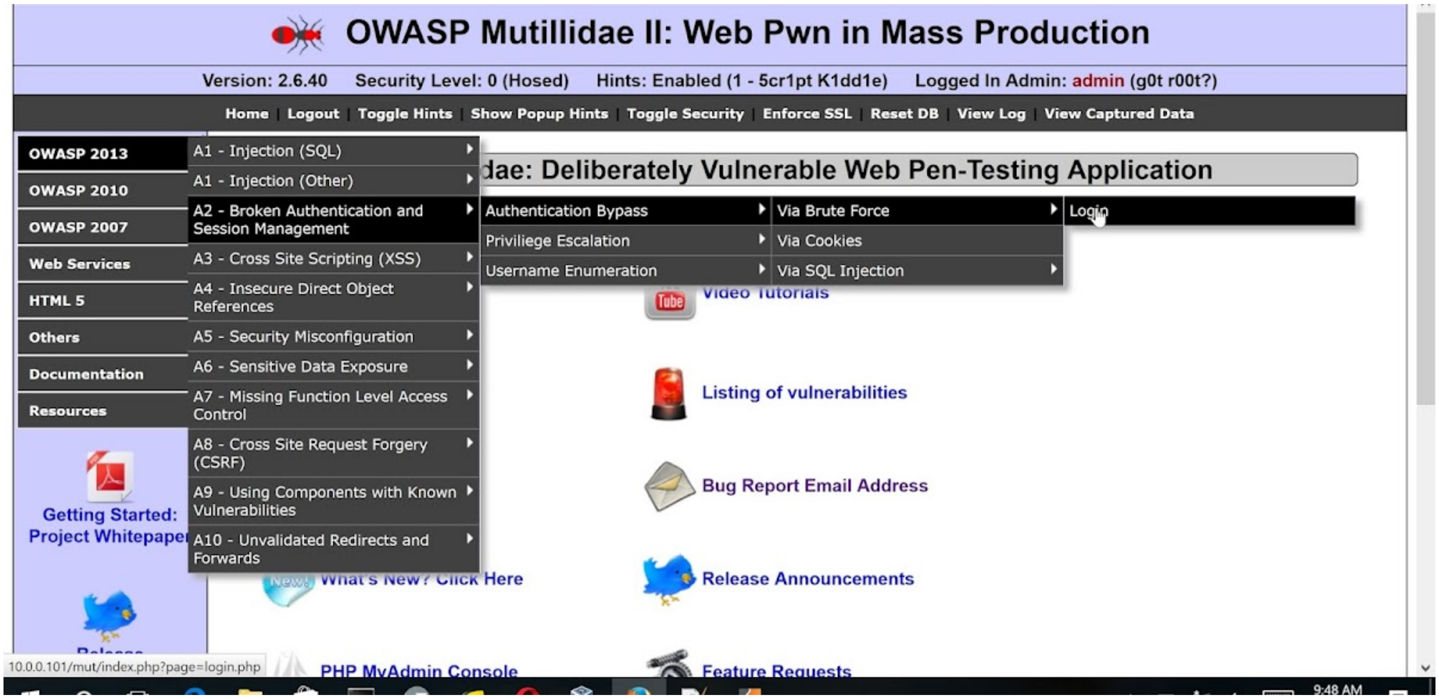
We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

General Instructions

The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

Mutillidae

The Mutillidae web application contains all of the vulnerabilities from the OWASP Top Ten plus a number of other vulnerabilities such as HTML-5 web storage, forms caching, and click-jacking. Inspired by DVWA, Mutillidae allows the user to change the "Security Level" from 0 (completely insecure) to 5 (secure). Additionally three levels of hints are provided ranging from "Level 0 - I try harder" (no hints) to "Level 2 - noob" (Maximum hints). If the application is damaged by user injections and hacks, clicking the "Reset DB" button resets the application to its original state.



OWASP Mutillidae II: Web Pwn in Mass Production

Version: 2.6.40 Security Level: 0 (Hosed) Hints: Enabled (1 - 5cr1pt K1dd1e) Logged In Admin: admin (g0t r00t?)

Home | Logout | Toggle Hints | Show Popup Hints | Toggle Security | Enforce SSL | Reset DB | View Log | View Captured Data
OWASP 2013
OWASP 2010
OWASP 2007
Web Services
HTML 5
Others
Documentation
Resources
A1 - Injection (SQL)
A1 - Injection (Other)
A2 - Broken Authentication and Session Management
A3 - Cross Site Scripting (XSS)
A4 - Insecure Direct Object References
A5 - Security Misconfiguration
A6 - Sensitive Data Exposure
A7 - Missing Function Level Access Control
A8 - Cross Site Request Forgery (CSRF)
A9 - Using Components with Known Vulnerabilities
A10 - Unvalidated Redirects and Forwards
Authentication Bypass
Priviliege Escalation
Username Enumeration
Via Brute Force
Via Cookies
Via SQL Injection
Login
dae: Deliberately Vulnerable Web Pen-Testing Application
Video Tutorials
Listing of vulnerabilities
Bug Report Email Address
Release Announcements
What's New? Click Here
Getting Started: Project Whitepaper
PHP MyAdmin Console
Feature Requests
10.0.0.101/mut/index.php?page=login.php
9:48 AM

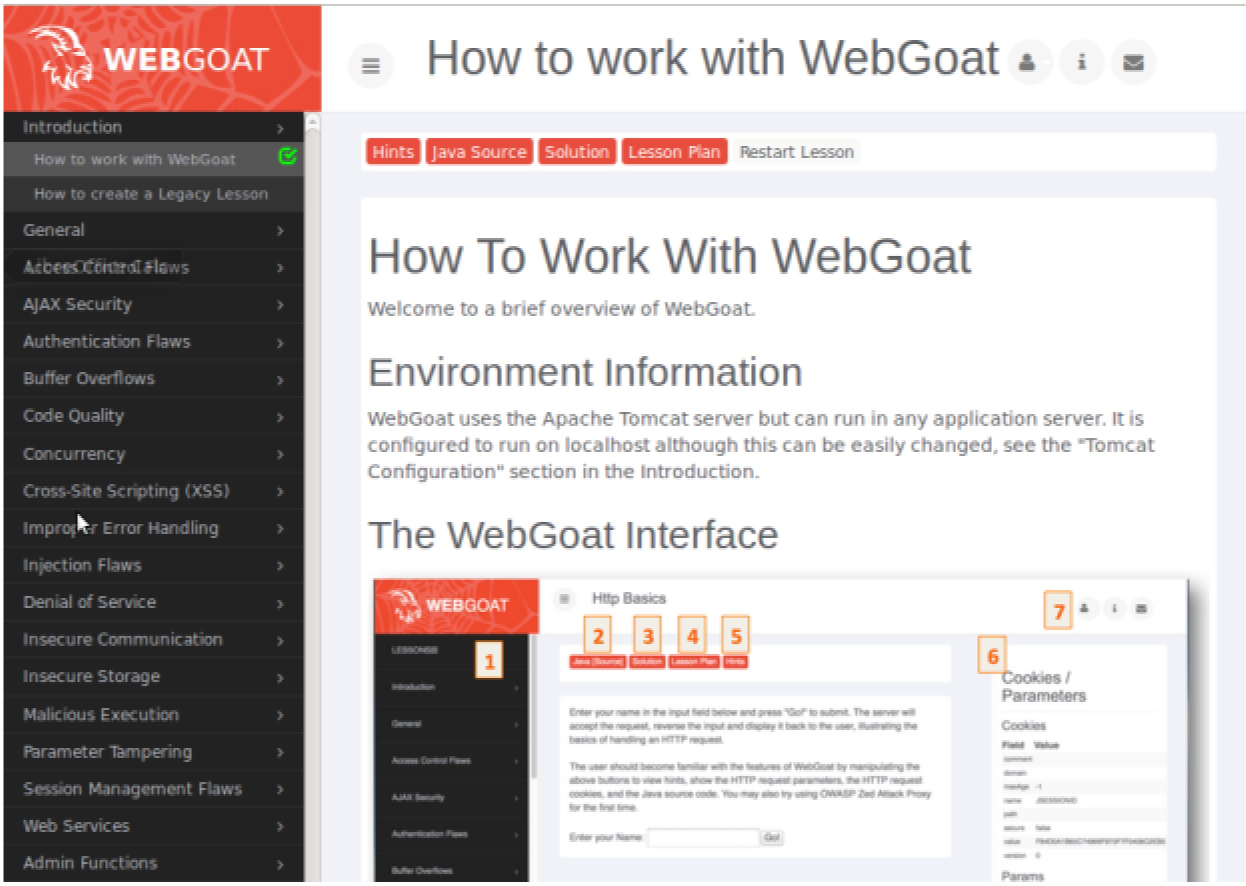
OWASP WebGoat

Web application security is difficult to learn and practice. Not many people have full blown web applications like online book stores or online banks that can be used to scan for vulnerabilities. In addition, security professionals frequently need to test tools against a platform known to be vulnerable to ensure that they perform as advertised. All of this needs to happen in a safe and legal environment.

Even if your intentions are good, we believe you should never attempt to find vulnerabilities without permission. The primary goal of the WebGoat project is simple: create a de-facto interactive teaching environment for web application security. In the future, the project team hopes to extend WebGoat into becoming a security benchmarking platform and a Java-based Web site Honeypot.

WARNING 1: *While running this program your machine will be extremely vulnerable to attack. You should disconnect from the Internet while using this program.* WebGoat's default configuration binds to localhost to minimize the exposure.

WARNING 2: *This program is for educational purposes only. If you attempt these techniques without authorization, you are very likely to get caught. If you are caught engaging in unauthorized hacking, most companies will fire you. Claiming that you were doing security research will not work as that is the first thing that all hackers claim.*



How To Work With WebGoat

Welcome to a brief overview of WebGoat.

Environment Information

WebGoat uses the Apache Tomcat server but can run in any application server. It is configured to run on localhost although this can be easily changed, see the "Tomcat Configuration" section in the Introduction.

The WebGoat Interface

Enter your name in the input field below and press "Go!" to submit. The server will accept the request, reverse the input and display it back to the user, illustrating the basics of handling an HTTP request.

The user should become familiar with the features of WebGoat by manipulating the above buttons to view hints, show the HTTP request parameters, the HTTP request cookies, and the Java source code. You may also try using OWASP Zed Attack Proxy for the first time.

Enter your Name:

bWAPP

bWAPP, or a *buggy web application*, is a free and open source deliberately insecure web application.

It helps security enthusiasts, developers and students to discover and to prevent web vulnerabilities.

bWAPP prepares one to conduct successful penetration testing and ethical hacking projects.

What makes bWAPP so unique? Well, it has over **100 web vulnerabilities**!

It covers all major known web bugs, including all risks from the OWASP Top 10 project.

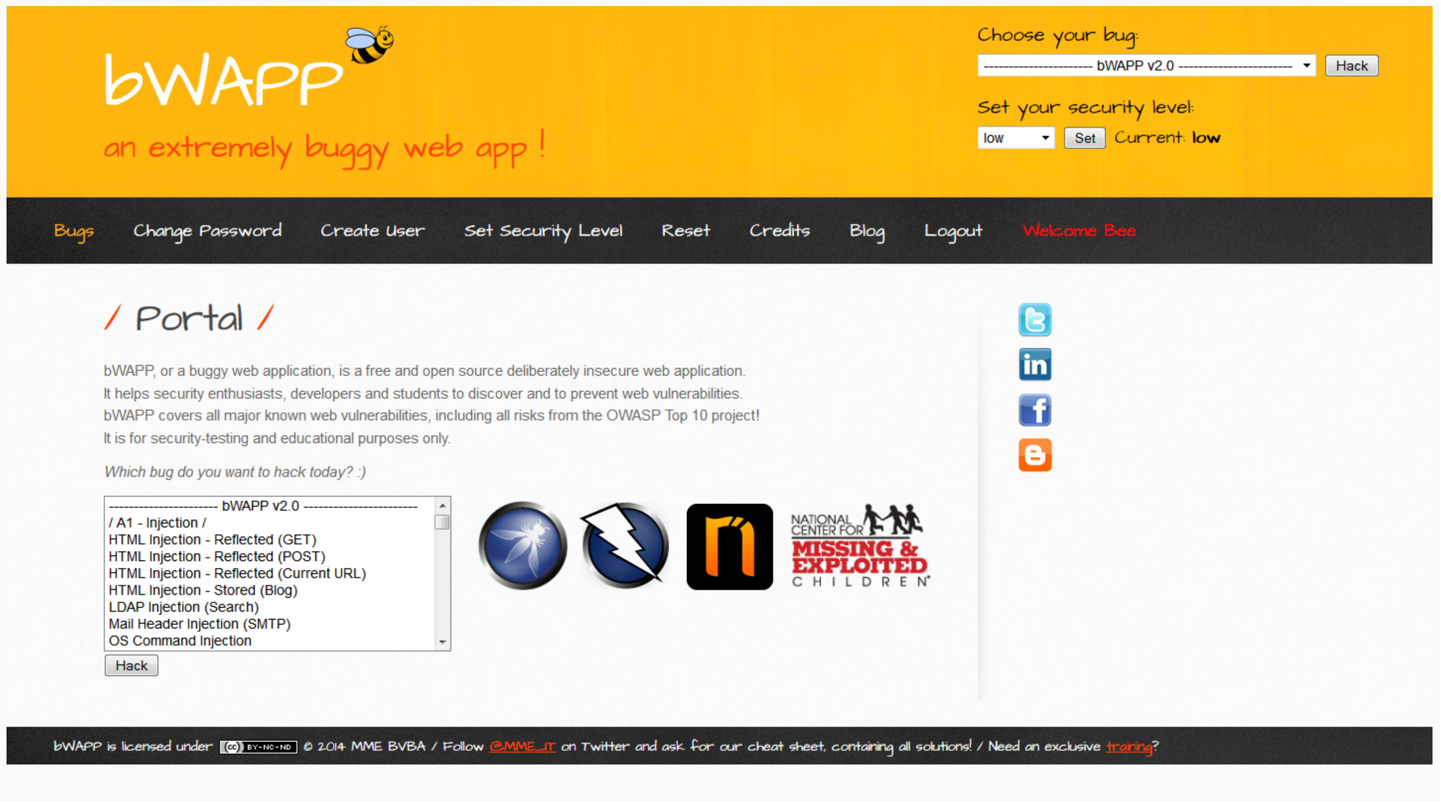
bWAPP is a PHP application that uses a MySQL database. It can be hosted on Linux/Windows with Apache/IIS and MySQL. It can also be installed with WAMP or XAMPP.

Another possibility is to download the *bee-box*, a custom Linux VM pre-installed with bWAPP.

Download our **What is bWAPP?** Introduction tutorial, including free exercises...

bWAPP is for web application security-testing and educational purposes only.

Have fun with this free and open source project!



/ Portal /

bWAPP, or a buggy web application, is a free and open source deliberately insecure web application.
It helps security enthusiasts, developers and students to discover and to prevent web vulnerabilities.
bWAPP covers all major known web vulnerabilities, including all risks from the OWASP Top 10 project!
It is for security-testing and educational purposes only.

Which bug do you want to hack today? :)

```
---------------------- bWAPP v2.0 ----------------------
/ A1 - Injection /
HTML Injection - Reflected (GET)
HTML Injection - Reflected (POST)
HTML Injection - Reflected (Current URL)
HTML Injection - Stored (Blog)
LDAP Injection (Search)
Mail Header Injection (SMTP)
OS Command Injection
```

Hack

bWAPP is licensed under (cc) BY-NC-ND © 2014 MME BVBA / Follow @MME_IT on Twitter and ask for our cheat sheet, containing all solutions! / Need an exclusive training?

OWASP Juice Shop

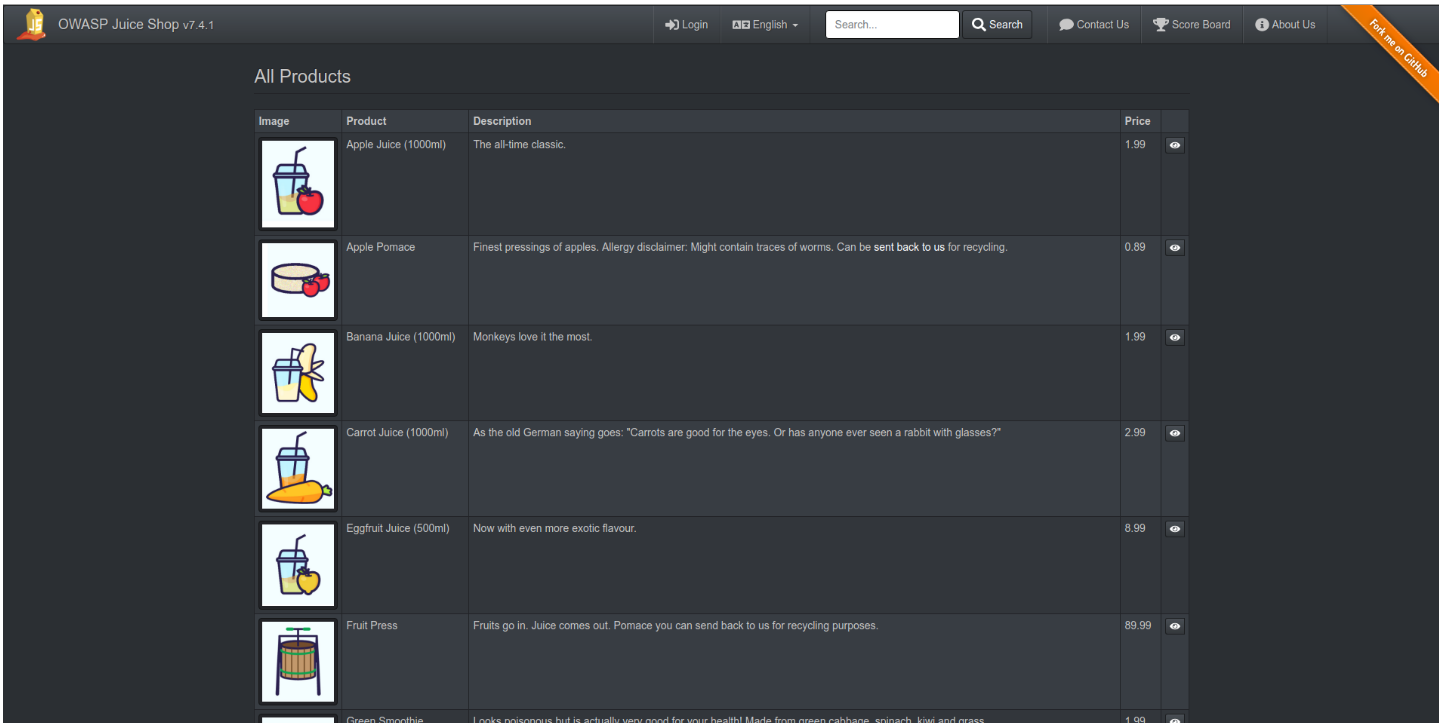
OWASP Juice Shop is probably the most modern and sophisticated insecure web application! It can be used in security training, awareness demos, CTFs and as a guinea pig for security tools! Juice Shop encompasses vulnerabilities from the entire OWASP Top Ten along with many other security flaws found in real-world applications!

Juice Shop is written in Node.js, Express and Angular. It was the first application written entirely in JavaScript listed in the OWASP VWA Directory.

The application contains a vast number of hacking challenges of varying difficulty where the user is supposed to exploit the underlying vulnerabilities. The hacking progress is tracked on a scoreboard. Finding this scoreboard is actually one of the (easy) challenges!

Apart from the hacker and awareness training use case, pentesting proxies or security scanners can use Juice Shop as a “guinea pig”-application to check how well their tools cope with JavaScript-heavy application front ends and REST APIs.

Translating “dump” or “useless outfit” into German yields “Saftladen” which can be reverse-translated word by word into “juice shop”. Hence the project name. That the initials “JS” match with those of “JavaScript” was purely coincidental!



All Products

Image	Product	Description	Price	
	Apple Juice (1000ml)	The all-time classic.	1.99	👁
	Apple Pomace	Finest pressings of apples. Allergy disclaimer: Might contain traces of worms. Can be sent back to us for recycling.	0.89	👁
	Banana Juice (1000ml)	Monkeys love it the most.	1.99	👁
	Carrot Juice (1000ml)	As the old German saying goes: "Carrots are good for the eyes. Or has anyone ever seen a rabbit with glasses?"	2.99	👁
	Eggfruit Juice (500ml)	Now with even more exotic flavour.	8.99	👁
	Fruit Press	Fruits go in. Juice comes out. Pomace you can send back to us for recycling purposes.	89.99	👁
	Green Smoothie	Looks poisonous but is actually very good for your health! Made from green cabbage, spinach, kiwi and grass.	1.99	👁

Opendns Security Ninjas

Security Ninjas is an Application Security Training Program that I created for our software developers here at OpenDNS. It has really helped our developers write more secure code and hence reduced the burden on our security team, so we thought of open sourcing it for the benefit of the community.

The training program slide deck covers the OWASP Top 10 (2013) vulnerabilities and some general security best practices. The hands-on training lab consists of 10 fun real world like hacking exercises, corresponding to each of the OWASP Top 10 vulnerabilities. Hints and solutions are provided along the way. Although the backend for this is written in PHP, vulnerabilities would remain the same across all web based languages, so the training would still be relevant even if you don't actively code in PHP.

Why Is Application Security Training Important?

1. It's hard to scale the Security Engineering team with the Software Development teams. It's practically impossible for the security team to review each line of code before it goes in production.
2. It's best to train developers so that they are not only able to catch security bugs during peer reviews, but also avoid writing vulnerable code in the first place. This approach scales well with fast dev cycles.
3. This sort of proactive approach also reduces the amount of work that needs to be put in reactively—both by the security team during reviews and by the dev teams while fixing bugs. Meaning? Less friction and faster code deployment!
4. This knowledge also helps developers understand security issues, risks and consequences faster especially when security bugs are reported.
5. Last but not the least it makes developers unconsciously care/informed about security.

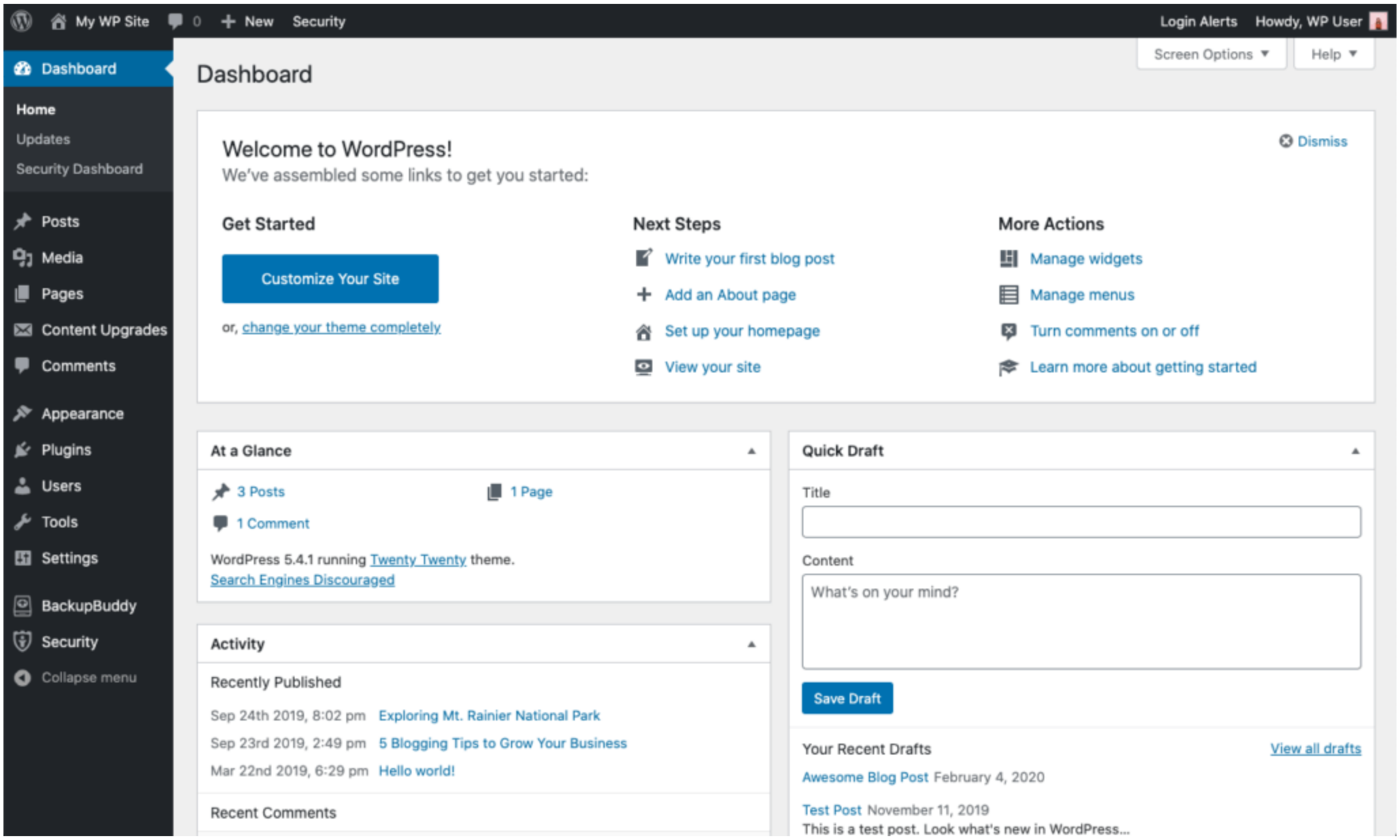
© OpenDNS. All rights reserved. | OpenDNS Security

Wordpress

WordPress (WP, WordPress.org) is a free and open-source content management system (CMS) written in PHP and paired with a MySQL or MariaDB database. Features include a plugin architecture and a template system, referred to within WordPress as Themes. WordPress was originally created as a blog-publishing system but has evolved to support other web content types including more traditional mailing lists and forums, media galleries, membership sites, learning management systems (LMS) and online stores. WordPress is used by 41.4% of the top 10 million websites as of May 2021, WordPress is one of the most popular content management system solutions in use. WordPress has also been used for other application domains, such as pervasive display systems (PDS).

WordPress was released on May 27, 2003, by its founders, American developer Matt Mullenweg and English developer Mike Little, as a fork of *b2/cafelog*. The software is released under the GPLv2 (or later) license.

To function, WordPress has to be installed on a web server, either part of an Internet hosting service like WordPress.com or a computer running the software package WordPress.org in order to serve as a network host in its own right. A local computer may be used for single-user testing and learning purposes.



Dashboard

Welcome to WordPress!

We've assembled some links to get you started:

Get Started

or, change your theme completely

Next Steps

- Write your first blog post
- Add an About page
- Set up your homepage
- View your site

More Actions

- Manage widgets
- Manage menus
- Turn comments on or off
- Learn more about getting started

At a Glance

3 Posts 1 Page

1 Comment

WordPress 5.4.1 running Twenty Twenty theme.

Search Engines Discouraged

Activity

Recently Published

Sep 24th 2019, 8:02 pm Exploring Mt. Rainier National Park

Sep 23rd 2019, 2:49 pm 5 Blogging Tips to Grow Your Business

Mar 22nd 2019, 6:29 pm Hello world!

Recent Comments

Quick Draft

Title

Content

What's on your mind?

Your Recent Drafts View all drafts

Awesome Blog Post February 4, 2020

Test Post November 11, 2019

This is a test post. Look what's new in WordPress...

Thank You!!

Join On my **GitHub**

Contact:- *DarkKing6@protonmail.com*