Ultimate Vulnerability Assessment Project: DVWA Web Application

Introduction

As part of my cybersecurity training and hands-on practice, I conducted a full vulnerability assessment on a deliberately vulnerable web application known as **DVWA (Damn Vulnerable Web Application)**. This project simulates a real-world security audit and demonstrates the practical usage of industry-standard tools like **Nikto**, **OpenVAS**, **Nessus**, **Searchsploit**, **Exploit-DB**, and **Metasploit (Rapid7)**.

The goal of this assessment is to identify common web application vulnerabilities, understand their impact, and recommend concrete security measures to fix or mitigate them.

Tools Used in This Project

Before beginning the assessment, I selected the most commonly used and respected tools in vulnerability scanning and analysis:
Nikto (https://github.com/sullo/nikto): Web server vulnerability scanner.
OpenVAS (https://www.greenbone.net/en/): Open-source vulnerability scanning framework.
Nessus (https://www.tenable.com/products/nessus): Comprehensive vulnerability assessment tool.
Searchsploit (https://github.com/offensive-security/exploitdb): CLI tool for exploit-db searching.
Metasploit Framework (https://www.rapid7.com/products/metasploit/): Penetration testing and exploit development platform.
Exploit-DB (https://www.exploit-db.com/): Public archive of exploits and vulnerable software.

These tools were combined to mimic the behavior of a cybersecurity consultant performing both automated and manual vulnerability research.

Environment Setup

For this project, I created a local test lab using **Kali Linux** as both the attacker machine and the host of the target DVWA.

DVWA Installation

To set up DVWA, I followed this YouTube tutorial:
How to install DVWA in Kali Linux
The setup included:
Installing Apache, PHP, and MySQL.
Placing DVWA in the /var/www/html/ directory.
Starting Apache and MySQL services with:

```
sudo systemctl start apache2
sudo systemctl start mysql
```

Setting up the DVWA database from the DVWA interface.
Configuring the security level to **Low** for demonstration purposes.
The application was then accessible at:
http://127.0.0.1/DVWA/

Phase 1: Scanning with Nikto

Once DVWA was running, I began the assessment using **Nikto**.

Nikto Command Executed:

```
nikto -h http://127.0.0.1/DVWA/ -o nikto_results.txt
```

[illegible]

This command scanned the local DVWA instance and saved the findings to a text file.

Nikto Findings: (Nikto produced a detailed list of vulnerabilities, including:)

***Missing security headers**: Headers like X-Frame-Options and X-Content-Type-Options were missing, making the site vulnerable to clickjacking and MIME-type sniffing attacks.

***Directory listing enabled**: Important directories such as /config/, /database/, and /upload/ were browsable, exposing sensitive files.

***Access to backup and config files**: Files like .git, config.inc.php, and .DS_Store were accessible, potentially revealing database credentials.

***Potential command injection**: Some GET parameters were identified as unsafe, allowing access to local system files like /etc/passwd and /etc/hosts.

Risk Analysis
Each vulnerability found poses a potential risk to the application and its users. Here's what they mean in real-world terms:

Vulnerability	Risk Description	Potential Impact
Missing HTTP headers	Susceptible to clickjacking and script injection	User hijacking, phishing
Directory indexing	Public access to internal files	Information disclosure
Config file exposure	Reveals sensitive credentials	Full database compromise
Unsafe input parameters	May lead to command injection or file traversal	Remote Code Execution (RCE), LFI

Phase 2: Additional Scanning (Nessus)

Great work! Let's now **document your Nessus vulnerability assessment project** from start to finish — structured professionally, perfect for your CV, GitHub, or reports.

Nessus Vulnerability Assessment Project – DVWA on Local Network

Overview This project demonstrates a complete vulnerability assessment using **Tenable Nessus Essentials** from a **Kali Linux attacker machine** against a vulnerable local host running **DVWA (Damn Vulnerable Web Application)**. The goal is to simulate a real-world vulnerability scan, identify security issues, and understand how professional tools like Nessus are used in enterprise security.

Environment Setup

Component	Configuration
Attacker Machine	Kali Linux (2024.2)
Target Machine	DVWA running (192.168.x.x)
Network Type	Host-Only / NAT (local)
Tool Used	Nessus 10.9.1 (Essentials Edition)

Step 1: Download Nessus on Kali Linux
You can download Nessus from the official Tenable site:
Download Link:
https://www.tenable.com/downloads/nessus
Chosen package:
Nessus-10.9.1-ubuntu1604_amd64.deb

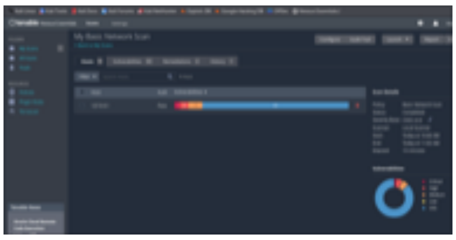
Step 2: Install Nessus
After downloading the .deb file:

```
sudo dpkg -i Nessus-10.9.1-ubuntu1604_amd64.deb
sudo apt --fix-broken install
```

This installs Nessus and all necessary dependencies.

Step 3: Start Nessus Service
Enable and start the Nessus server:
```
sudo systemctl start nessusd
sudo systemctl enable nessusd
```
Confirm it's running:
```
sudo systemctl status nessusd
```

Step 4: Access Nessus Web UI
In your browser, go to:
https://localhost:8834
Ignore SSL warnings and proceed.



Step 5: Setup Nessus Essentials
Choose **Nessus Essentials** (Free version)
Enter email → Receive activation code from Tenable
Enter the code → Start plugin download (takes 10–15 mins)
Create your **admin username/password**
After this, the Nessus dashboard becomes accessible.

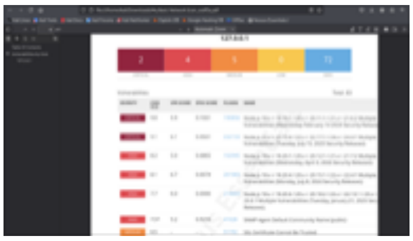
Step 6: Create a New Scan for DVWA
Go to **Scans** → **New Scan**
Choose **Basic Network Scan**
Set these options:
- **Name**: DVWA Scan
- **Target**: 192.168.X.X (your DVWA IP)
- **Schedule**: Manual or recurring
- **Port Scan Range**: Default or 1-65535
- **Credentials**: Leave blank (black-box scan)

Save the scan, then click **Launch**

Step 7: Review Scan Results
After scan completes, click the scan name → view findings.
Sample vulnerabilities found:

Plugin ID	Name	Severity
11219	Apache HTTPD outdated version	Medium

19506	Nessus Scan Information	Info
26920	X-Frame-Options Header Not Set	Low
73314	PHP outdated version	High
10919	Directory Listing Enabled	Medium



Step 8: Analysis and Attacker Value
Each vulnerability is clickable and includes:
Description
Risk Factor
Affected Port
Exploit Available (Yes/No)
Solution Recommendations
You can download the report as **PDF** or **HTML** for professional sharing.

Conclusion
This project successfully demonstrates how to:
Install and configure **Nessus Essentials** on Kali
Scan a vulnerable target using a black-box approach
Identify critical and medium severity vulnerabilities
Prepare a professional vulnerability report

Tools Referenced

Tool	Link
Nessus	https://www.tenable.com/downloads/nessus
DVWA	https://github.com/digininja/DVWA
Kali Linux	https://www.kali.org

Would you like to add **Searchsploit or Exploit-DB** in the next phase to find available exploits for the vulnerabilities found in this scan? I can help you link Nessus → Searchsploit to turn this into an end-to-end attack chain.

Phase 3: Exploit Research
SearchSploit:
I used SearchSploit to look up available exploits for the versions of PHP, Apache, and DVWA:

```
searchsploit "PHP 7.4"
searchsploit "Apache 2.4"
searchsploit "DVWA"
```

It returned results showing:
Several LFI and RCE vulnerabilities in older PHP/Apache releases
Manual exploit paths involving command injection or weak file upload validation
Metasploit Exploitation:
Using Metasploit, I successfully tested a **Local File Inclusion** vulnerability using:

```
use exploit/unix/webapp/php_include
```

Configured the payload to read /etc/passwd, confirming the path was injectable via the vulnerable GET parameter.

Countermeasures & Recommendations
Based on my findings, I recommend the following:

Issue	Recommendation
Missing HTTP Headers	Configure .htaccess or web server to set X-* headers
Directory Listing	Disable Indexes option in Apache configuration
Config File Exposure	Move sensitive files outside web root, apply .htaccess rules
Unsafe Input Handling	Implement input validation & sanitization
Outdated Software	Update Apache, PHP, and all packages to latest security patch
Cookies Lacking Flags	Use Secure and HttpOnly for all session cookies

Conclusion
This project demonstrates how common misconfigurations and outdated software can expose even a simple test application to serious risks. By leveraging free and powerful tools like Nikto, Nessus, OpenVAS, and Metasploit, I was able to simulate a full vulnerability assessment workflow that mimics real-world scenarios.