

Web Application Enumeration with DirBuster and dirb (Using DVWA)

**The domain of this project encompasses
cybersecurity education and practical training in
web application security assessment using tools
like DirBuster.**

**CDAC, Noida
CYBER GYAN VIRTUAL INTERNSHIP
PROGRAM**

Submitted By:

Vivek Kumar

Project Trainee - Mr. Varun Mishra,

(21 Aug – 30 Aug) 2024

BONAFIDE CERTIFICATE

This is to certify that this project report entitled **Web Application Enumeration with DirBuster and dirb (Using DVWA)** submitted to CDAC Noida, is a Bonafede record of work done by **Vivek Kumar** under my supervision from 21 Aug 2024 to 30 Aug 2024.

(Signature)

HEAD OF THE DEPARTMENT

(Signature)

SUPERVISOR

Declaration by Author(s)

This is to declare that this report has been written by me/us. No part of the report is plagiarized from other sources. All information included from other sources have been duly acknowledged. I/We aver that if any part of the report is found to be plagiarized, I/we are shall take full responsibility for it.

Name of Author(S): Vivek Kumar

TABLE OF CONTENTS

1. PROBLEM STATEMENT	6
2. Learning Objective	6
3. APPROACH	7
3.1. Tools and Technologies Used	7
3.2. Infrastructure Created	8
3.3. Process Workflow	8
4. IMPLEMENTATION	9
4.1. Set Up the DVWA Server	9
4.2. Set Up the Attacker's Machine (Kali Linux)	11
4.3. Perform Enumeration with Dirb	11
4.4. Perform Enumeration with DirBuster	12
4.5. Documenting Indicators of Compromise (IoCs)	13
4.6. Final Documentation and Reporting	14
5. CONCLUSION & RECOMMENDATIONS	14
6. LIST OF REFERENCES	16

ACKNOWLEDGEMENT

Firstly, I am grateful to my mentor for providing me with valuable guidance and support throughout the internship period. Their insights and suggestions have been instrumental in shaping my research and analysis for this report.

I would like to express my appreciation to the entire team at the organization where I completed my internship. Their collaboration and willingness to share their knowledge and experiences have been invaluable to me.

Their constructive criticism and input have helped me refine my ideas and clarify my arguments. I am truly fortunate to have such a supportive academic community.

Vivek Kumar

Web Application Enumeration with DirBuster and dirb (Using DVWA)

PROBLEM STATEMENT:

In this project, I have used both Dirb and DirBuster to perform web application enumeration specifically on Damn Vulnerable Web Application (DVWA). I have accessed DVWA hosted on a local or virtual environment (<http://localhost/dvwa>) and configure both tools with the target URL for directory brute-forcing. Also, I have started by using Dirb to perform an initial enumeration. Dirb will systematically send HTTP requests to DVWA, analyzing responses to identify existing directories and files. Following this, I have used DirBuster to perform a more comprehensive enumeration, leveraging its advanced capabilities and graphical interface for deeper analysis.

Learning Objective

1. Understand Web Application Enumeration:

- Gain a solid understanding of the concept of web application enumeration and its importance in the cybersecurity domain.
- Learn how attackers use enumeration to identify hidden directories and files in web applications, which could potentially expose sensitive information.

2. Hands-On Experience with Enumeration Tools:

- Acquire practical skills in using Dirb and DirBuster, two essential tools for web application enumeration.
- Learn how to configure these tools for effective directory and file brute-forcing, understanding their strengths and limitations.

3. Assessing Web Application Security:

- Develop the ability to assess the security posture of a web application by identifying potential vulnerabilities through enumeration.
- Understand how directory and file enumeration can reveal weak points in web applications that need to be secured.

4. Documentation and Reporting:

- Enhance skills in documenting the enumeration process, including setup, configuration, and analysis of results.
- Learn to communicate findings effectively through written reports and presentation slides, making the results accessible to both technical and non-technical audiences.

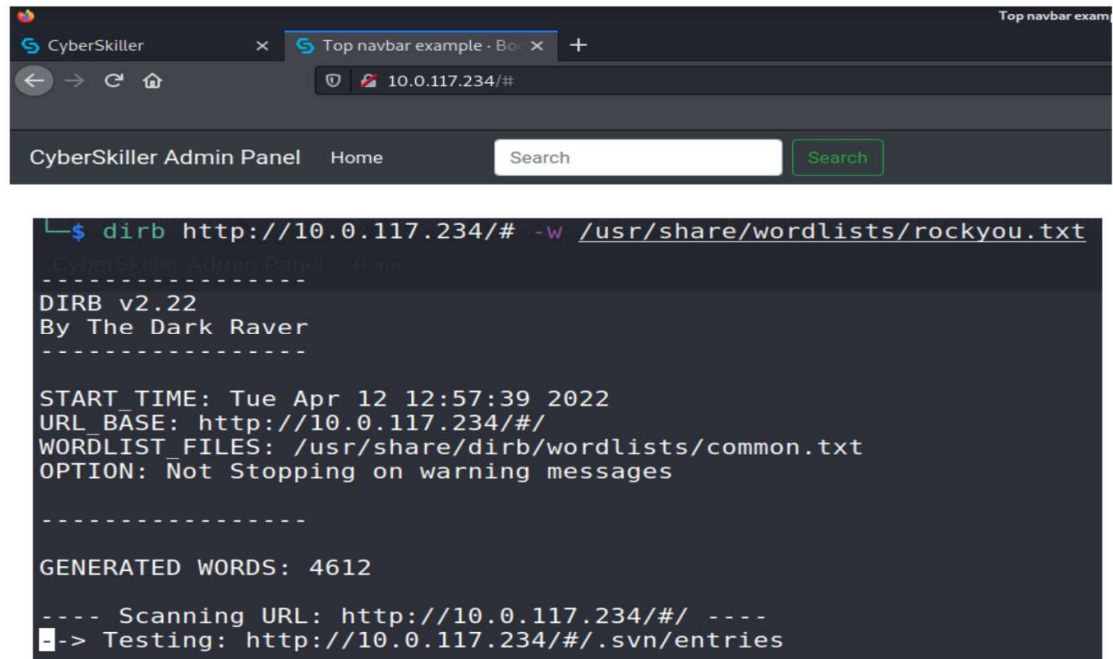
5. Familiarity with DVWA:

- Gain experience with the Damn Vulnerable Web Application (DVWA), a widely used tool for practicing web application security testing in a controlled environment.

APPROACH:

1. Tools and Technologies Used:

- **Dirb:** A command-line tool used for brute-forcing directories and files on a web server by sending HTTP requests and analyzing the responses.



The screenshot shows a web browser window with the address bar set to `10.0.117.234/#`. Below the browser, a terminal window displays the output of the `dirb` command. The command used is `dirb http://10.0.117.234/# -w /usr/share/wordlists/rockyou.txt`. The terminal output shows the Dirb version (v2.22), the URL base, the wordlist file, and the generated words (4612). It also shows the scanning process for the URL `http://10.0.117.234/#` and the testing of `http://10.0.117.234/#.svn/entries`.

```
$ dirb http://10.0.117.234/# -w /usr/share/wordlists/rockyou.txt
-----
DIRB v2.22
By The Dark Raver
-----

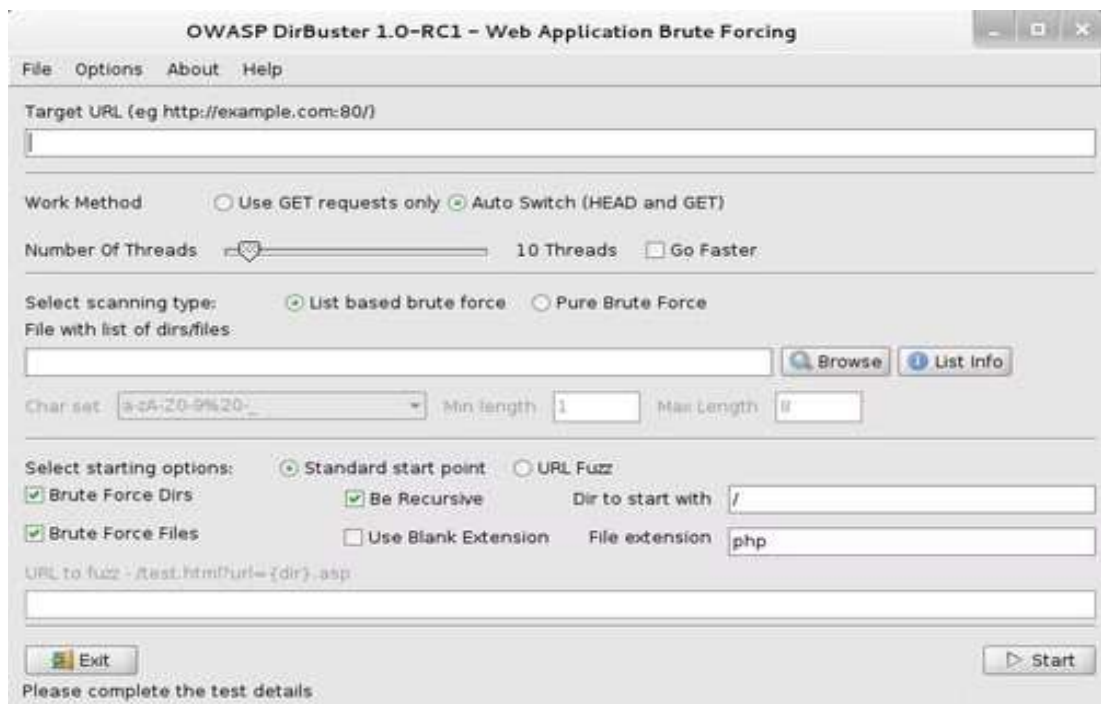
START TIME: Tue Apr 12 12:57:39 2022
URL_BASE: http://10.0.117.234/#/
WORDLIST FILES: /usr/share/dirb/wordlists/common.txt
OPTION: Not Stopping on warning messages

-----

GENERATED WORDS: 4612

---- Scanning URL: http://10.0.117.234/#/ ----
-> Testing: http://10.0.117.234/#.svn/entries
```

- **DirBuster:** A GUI-based tool for directory and file brute-forcing with advanced capabilities, including wordlist selection, threading, and recursive scanning.



- **Damn Vulnerable Web Application (DVWA):** A deliberately vulnerable web application used for practicing web security testing.



- **Operating System:**
 - **Kali Linux:** Used as the attacker's machine for running Dirb and DirBuster.
 - **Ubuntu (or any other Linux-based distribution):** Used to host the DVWA.
- **Virtualization Software:**
 - **VirtualBox or VMware:** Used to create virtual machines for both the attacker's machine and the DVWA server.
- **Networking:**
 - **Localhost/Private Network:** The DVWA is hosted locally, and the enumeration tools are run on the same network.

2. Infrastructure Created:

The project involves setting up a simple but realistic environment where DVWA is hosted on a virtual machine (VM), and the enumeration is performed from another VM using Kali Linux. The entire setup can be done on a single physical machine using virtualization software.

- **DVWA Server:**
 - **OS:** Ubuntu (or any Linux-based distribution)
 - **IP Address:** 192.168.142.205 (example IP within the private network)
 - **Web Server:** Apache
 - **DVWA URL:** [http:// 192.168.142.205/dvwa](http://192.168.142.205/dvwa)
 - **Firewall Settings:** Minimal firewall configuration to allow HTTP traffic to DVWA.
- **Attacker's Machine:**
 - **OS:** Kali Linux
 - **IP Address:** 192.168.142.205 (example IP within the private network)
 - **Tools Installed:** Dirb, DirBuster

3. Process Workflow:

- **Setup the DVWA Server:**
 - Install Ubuntu (or another Linux distribution) on a virtual machine.

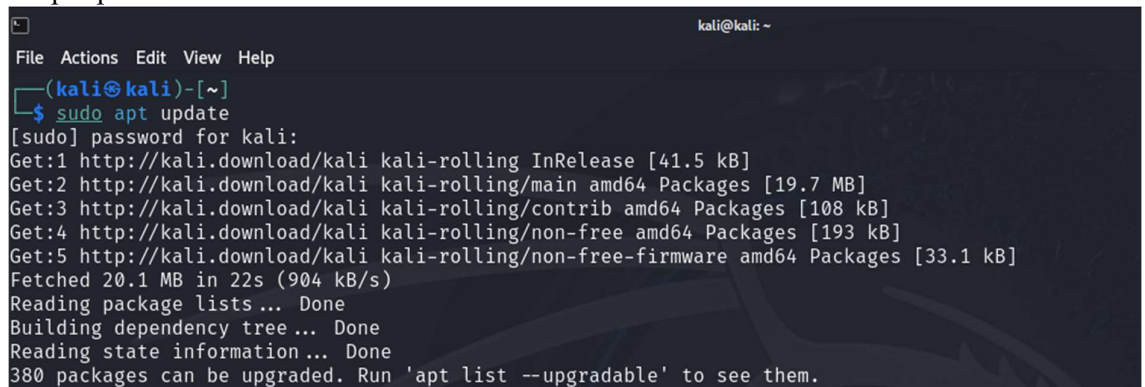
- Install and configure Apache, MySQL, and PHP.
 - Download and configure DVWA.
 - Set the network interface to use a private network and assign the IP 192.168.142.205.
- **Configure the Attacker's Machine:**
 - Install Kali Linux on a separate virtual machine.
 - Ensure the machine is on the same private network as the DVWA server.
 - Install Dirb and DirBuster.
 - Verify connectivity to the DVWA server by accessing `http://192.168.142.205/dvwa` from a web browser on Kali Linux.
- **Perform Enumeration with Dirb:**
 - Run Dirb from the Kali Linux machine against the DVWA server.
 - Record the discovered directories and files.
- **Perform Enumeration with DirBuster:**
 - Launch DirBuster from the Kali Linux machine.
 - Target the DVWA server and perform a more in-depth enumeration.
 - Analyze and compare the results with those from Dirb.
- **Document the Findings:**
 - Compile the findings into a report, detailing the setup, tools used, and the results of the enumeration.

IMPLEMENTATION:

The implementation section details the step-by-step process followed to perform web application enumeration using Dirb and DirBuster on Damn Vulnerable Web Application (DVWA).

Step 1: Set Up the DVWA Server

1. **Install Ubuntu on a Virtual Machine:**
 - Download the Ubuntu ISO and install it on a virtual machine using VirtualBox or VMware.
 - Configure the network interface to use a private network (e.g., Host-Only Adapter).
2. **Install Apache, MySQL, and PHP (LAMP Stack):**
 - Update the package lists:
`sudo apt update`



```

kali@kali: ~
File Actions Edit View Help
(kali@kali)-[~]
$ sudo apt update
[sudo] password for kali:
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [19.7 MB]
Get:3 http://kali.download/kali kali-rolling/contrib amd64 Packages [108 kB]
Get:4 http://kali.download/kali kali-rolling/non-free amd64 Packages [193 kB]
Get:5 http://kali.download/kali kali-rolling/non-free-firmware amd64 Packages [33.1 kB]
Fetched 20.1 MB in 22s (904 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
380 packages can be upgraded. Run 'apt list --upgradable' to see them.

```

- Install Apache:
sudo apt install apache2

```
kali@kali: ~
File Actions Edit View Help

(kali@kali)-[~]
$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.62-1).
0 upgraded, 0 newly installed, 0 to remove and 380 not upgraded.
```

- Install MySQL:
sudo apt install mysql-server

- Install PHP:
sudo apt install php libapache2-mod-php php-mysql

```
kali@kali: ~
File Actions Edit View Help

(kali@kali)-[~]
$ sudo apt install php libapache2-mod-php php-mysql
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
php is already the newest version (2:8.2+93+nmu1).
libapache2-mod-php is already the newest version (2:8.2+93+nmu1).
php-mysql is already the newest version (2:8.2+93+nmu1).
0 upgraded, 0 newly installed, 0 to remove and 380 not upgraded.
```

3. Download and Configure DVWA:

- Clone the DVWA repository:
git clone <https://github.com/digininja/DVWA.git>

```
kali@kali: ~
File Actions Edit View Help

(kali@kali)-[~]
$ git clone https://github.com/digininja/DVWA.git
Cloning into 'DVWA'...
remote: Enumerating objects: 4710, done.
remote: Counting objects: 100% (260/260), done.
remote: Compressing objects: 100% (163/163), done.
remote: Total 4710 (delta 133), reused 198 (delta 92), pack-reused 4450 (from 1)
Receiving objects: 100% (4710/4710), 2.37 MiB | 41.00 KiB/s, done.
Resolving deltas: 100% (2228/2228), done.
```

- Move DVWA to the web server's root directory:
sudo mv DVWA /var/www/html/dvwa

```
kali@kali: ~
File Actions Edit View Help

(kali@kali)-[~]
$ sudo mv DVWA /var/www/html/dvwa

(kali@kali)-[~]
$
```

- Set the necessary permissions:
sudo chown -R www-data:www-data /var/www/html/dvwa

```
(kali㉿kali)-[~]
$ sudo chown -R www-data:www-data /var/www/html/dvwa

(kali㉿kali)-[~]
$
```

- Configure DVWA by editing the config/config.inc.php file:

sudo nano /var/www/html/dvwa/config/config.inc.php

```
File Actions Edit View Help
GNU nano 8.1 /var/www/html/dvwa/config/config.inc.php *
<?php
// DBMS connection settings.
$_DVWA = array();
$_DVWA['db_server'] = 'localhost'; // Database server. (Usually 'localhost')
$_DVWA['db_database'] = 'dvwa'; // Database name.
$_DVWA['db_user'] = 'root'; // MySQL username.
$_DVWA['db_password'] = 'password'; // MySQL password.

// Only used with PHP FPM. In most cases you can ignore this setting.
//$_DVWA['db_port'] = '3306';

// ReCAPTCHA settings (if you want to use it).
$_DVWA['recaptcha_public_key'] = ''; // Public key.
$_DVWA['recaptcha_private_key'] = ''; // Private key.

// Default security level (1 = low, 5 = high).
$_DVWA['default_security_level'] = '1'; // Can be changed via the web interface.

// Allow URL include (for Remote File Inclusion exercises).
$_DVWA['allow_url_include'] = 'off'; // Default: 'off'

?>
```

- Set the MySQL username, password, and database.

4. Access DVWA:

- Open a browser and go to <http://192.168.142.205/dvwa>.
- Complete the DVWA setup by creating the database.

Step 2: Set Up the Attacker's Machine (Kali Linux)

1. Install Kali Linux on a Virtual Machine:

- Download the Kali Linux ISO and install it on another virtual machine.
- Ensure the network interface is on the same private network as the DVWA server.

2. Verify Connectivity:

- Ping the DVWA server from Kali Linux to confirm connectivity:
ping 192.168.142.205
- Access DVWA from a browser on Kali Linux:
[http:// 192.168.142.205/dvwa](http://192.168.142.205/dvwa)

Step 3: Perform Enumeration with Dirb

1. Run Dirb:

- Open a terminal in Kali Linux.

- Execute the Dirb command to target the DVWA server:
dirb <http://192.168.56.101/dvwa>
- Dirb will start brute-forcing directories and files.

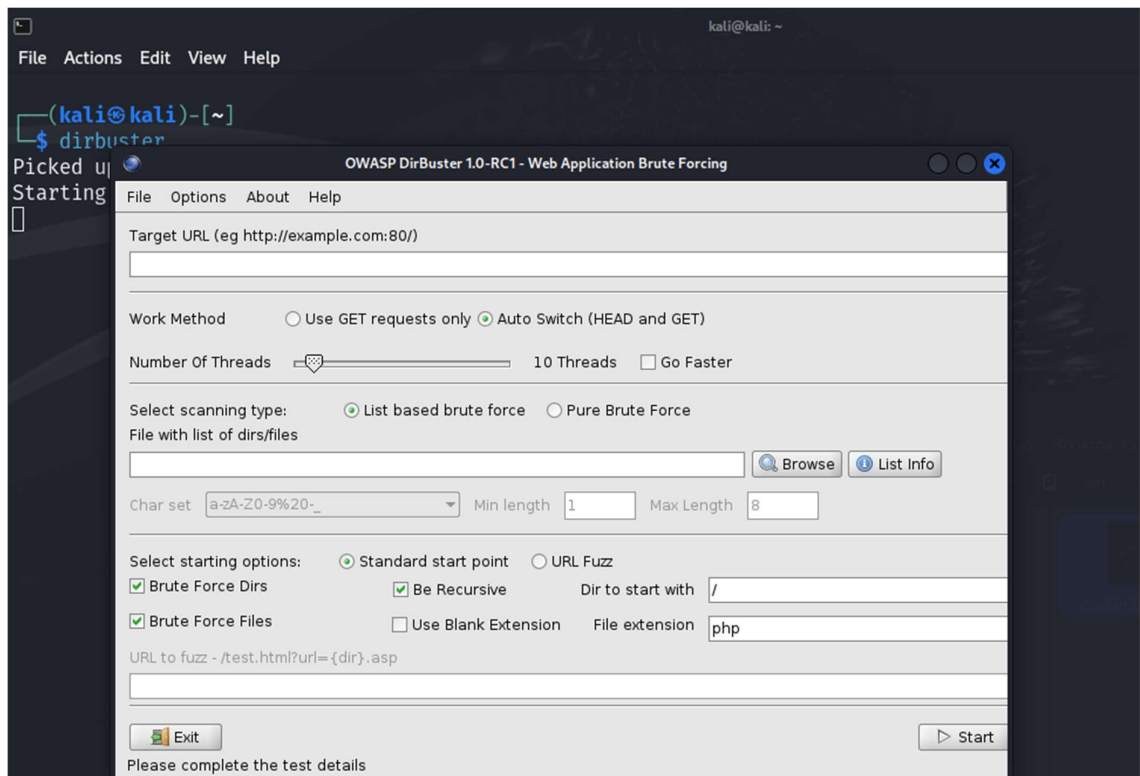
2. Analyze the Results:

- Note any significant directories or files discovered (e.g., /admin/, /uploads/).

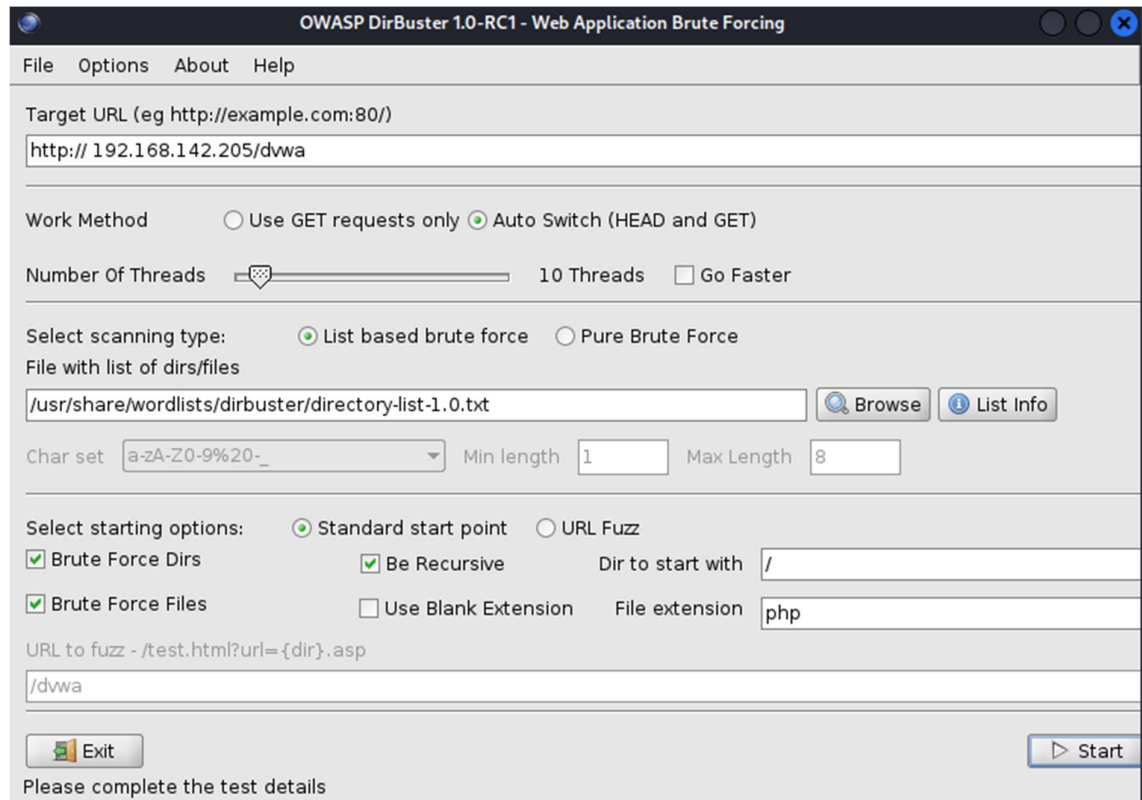
Step 4: Perform Enumeration with DirBuster

1. Launch DirBuster:

- Open DirBuster in Kali Linux from the terminal:
Dirbuster



- Enter the target URL ([http:// 192.168.142.205/dvwa](http://192.168.142.205/dvwa)) in DirBuster's GUI.



- Choose a wordlist and configure thread settings for the scan.

2. Run the Scan:

- Start the enumeration process.
- Monitor the progress and results in real-time as DirBuster brute-forces directories and files.

3. Compare the Results:

- Compare the results from DirBuster with those from Dirb to identify any additional findings.

Step 5: Documenting Indicators of Compromise (IoCs)

1. Analyze the Enumeration Impact:

- Identify IoCs such as unauthorized access to hidden directories or files that could indicate potential vulnerabilities.
- Common IoCs include:
 - Discovery of administrative directories (/admin/).
 - Access to upload directories (/uploads/) where files could be injected.
 - Exposure of sensitive configuration files (/config.php).

2. Document Potential Risks:

- Highlight the security risks associated with the findings, such as the potential for unauthorized access or data leakage.

Step 6: Final Documentation and Reporting

1. Compile the Report:

- Document each step in detail, including setup, execution, and analysis.
- Include all relevant screenshots to illustrate the process.
- Summarize the findings, focusing on the key directories and files discovered.

2. Prepare the Presentation:

- Create slides summarizing the purpose, tools, methodology, results, and lessons learned.
- Include key screenshots from the process to visually support your findings.

This structured implementation process ensures a comprehensive understanding of web application enumeration using Dirb and DirBuster, highlighting both the technical steps and the importance of analyzing potential vulnerabilities.

CONCLUSION & RECOMMENDATIONS:

In this project, we focused on web application enumeration using Dirb and DirBuster against the Damn Vulnerable Web Application (DVWA). The key findings and outcomes from the project are as follows:

1. Effective Enumeration Tools:

- **Dirb** and **DirBuster** proved to be effective tools for web application enumeration. They identified various directories and files within the DVWA application, demonstrating their capability in discovering hidden resources and potential vulnerabilities.

2. Configuration Challenges:

- Several issues were encountered related to the configuration of DVWA, including undefined constants and array keys. These issues highlighted the importance of ensuring that all configuration settings and constants are correctly defined and compatible with the tools being used.

3. Database Connection Issues:

- There were challenges with establishing a connection to the database, including undefined constants and missing configuration settings. These issues emphasized the need for proper database setup and configuration in web application environments.

4. Tool Performance:

- **Dirb** provided a quick enumeration with basic capabilities, while **DirBuster** offered a more comprehensive approach with advanced options and a graphical interface. Both tools complemented each other well in the enumeration process.

To enhance security and mitigate the vulnerabilities identified during the project, the following countermeasures and best practices are recommended:

1. Regular Updates and Patching:

- **Keep Software Updated:** Ensure that web applications, tools, and server software are regularly updated to the latest versions to protect against known vulnerabilities and bugs.

2. Strengthen Configuration and Error Handling:

- **Verify Configuration Settings:** Regularly review and test configuration files to ensure all necessary settings are correctly defined and that there are no missing keys or constants.
- **Implement Error Handling:** Implement robust error handling and validation to manage undefined variables and constants gracefully.

3. Improve Database Security:

- **Secure Database Credentials:** Use strong, unique passwords for database users and ensure that credentials are securely stored and managed.
- **Limit Database Access:** Restrict database access to only necessary applications and users. Use least privilege principles to minimize exposure.

4. Enhance Web Application Security:

- **Perform Regular Security Assessments:** Conduct regular security assessments and penetration testing to identify and address vulnerabilities.
- **Implement Security Best Practices:** Follow security best practices for web applications, such as input validation, output encoding, and proper authentication and authorization controls.

5. Educate and Train:

- **Cybersecurity Training:** Provide training for developers and administrators on secure coding practices and common vulnerabilities to prevent security issues.

6. Leverage Security Tools:

- **Use Comprehensive Security Tools:** In addition to Dirb and DirBuster, consider using other security tools for vulnerability scanning and assessment to cover a broader range of potential issues.

LIST OF REFERENCES:

1. [A Comparative Study of Directory Fuzzing Tools | IEEE Conference Publication | IEEE Xplore](#)
2. [Electronics | Free Full-Text | Teaching a Hands-On CTF-Based Web Application Security Course \(mdpi.com\)](#)