Dirbuster Script Documentation

Purpose of the Scripts

Dirbuster (or Dirb) is a directory brute forcing tool commonly used to find hidden directories and files on web servers. The provided scripts automate tasks such as performing basic directory scans, using custom wordlists, and scanning with specific file extensions, enabling security teams to identify hidden resources efficiently.

The first script, Basic Directory Bruteforce, scans a target URL using a default wordlist to discover hidden directories and files. The second script, Custom Wordlist Scan, allows the user to specify a custom wordlist for targeted scanning. The third script, Scan with Extensions, performs directory scanning with additional file extensions, increasing the likelihood of finding hidden resources.

Usage Instructions

General Setup

1. Ensure Dirb is installed on your system. You can install it using your package manager.

   "sudo apt install dirb"

2. Install Python if not already installed.
3. Save each script as a .py file and run them from the command line or any Python IDE.

Script 1: Basic Directory Bruteforce

Command:

python basic_dirb_scan.py

Required Parameters:

- ✓ target_url: The URL of the website to scan (e.g., http://example.com).

Script 2: Custom Wordlist Scan

Command:

python custom_wordlist_scan.py

Required Parameters:

- ✓ target_url: The URL of the website to scan (e.g., http://example.com).
- ✓ wordlist: The path to the custom wordlist file (e.g., /usr/share/wordlists/custom.txt).

Script 3: Scan with Extensions

Command:

python dirb_scan_with_extensions.py

Required Parameters:

- ✓ target_url: The URL of the website to scan (e.g., http://example.com).
- ✓ wordlist: The path to the wordlist file (e.g., /usr/share/wordlists/common.txt).
- ✓ extensions: A comma-separated list of file extensions to scan (e.g., .php,.html,.txt).

Expected Outputs or Results

Basic Directory Bruteforce

- ✓ Outputs the discovered directories and files in the terminal.
- ✓ Results are displayed with HTTP response codes indicating accessibility.

Custom Wordlist Scan

- ✓ Performs a targeted scan using the specified wordlist.
- ✓ Outputs discovered directories and files to the terminal.

Scan with Extensions

- ✓ Scans directories and files using additional file extensions.
- ✓ Displays results in the terminal, showing matched directories/files with response codes.

Dependencies Needed

- ✓ Dirb - Required for directory brute-forcing.
- ✓ Python - Required to run the scripts.
- ✓ Subprocess Module - Used to interact with the command-line interface.

Line-by-Line Explanation

Basic Directory Bruteforce

```
import subprocess
```

- subprocess: Used to run Dirb commands from the script.

```
def basic_dirb_scan(target_url):
```

- Defines a function to perform a basic directory scan on the target URL.

```
print(f"Running Dirb scan on {target_url}...")
subprocess.run(["dirb", target_url])
```

- Prints a message indicating the start of the scan and executes the Dirb command for the target URL.

Custom Wordlist Scan

```
import subprocess
```

- subprocess: Used to run Dirb commands from the script.

```
def custom_wordlist_scan(target_url, wordlist):
```

- Defines a function to scan the target URL using a custom wordlist.

```
print(f"Running Dirb scan on {target_url} with wordlist {wordlist}...")
subprocess.run(["dirb", target_url, wordlist])
```

- Prints a message indicating the start of the scan and executes the Dirb command with the specified wordlist.

Scan with Extensions

```
import subprocess
```

- subprocess: Used to run Dirb commands from the script.

```
def dirb_scan_with_extensions(target_url, wordlist, extensions):
```

- Defines a function to scan the target URL with additional file extensions.

```
print(f"Running Dirb scan on {target_url} with extensions {extensions}...")
subprocess.run(["dirb", target_url, wordlist, "-X", extensions])
```

- Prints a message indicating the start of the scan and executes the Dirb command with the specified extensions.

Why These Scripts Are Needed ?

Automating Dirb tasks simplifies the process of discovering hidden directories and files on web servers, which is a critical part of web application security assessments. The Basic Directory Bruteforce script provides a quick and easy way to identify common hidden resources. The Custom Wordlist Scan script allows for targeted scanning based on specific requirements, increasing the chances of finding less common directories and files. The Scan with Extensions script extends the search scope to include specific file types, making it invaluable for uncovering sensitive files like configuration files, scripts, and logs.