CPanelWalker Help File

# Overview

The `CpanelWalker.py` script is designed to brute-force passwords and usernames for CPanel and similar services. It utilizes proxies, random User-Agent headers, and dynamic IP detection to maximize the chance of success while avoiding detection and blocking.

The script:

- Fetches proxies dynamically from ProxyScrape.
- Uses a provided password file to attempt brute-forcing credentials.
- Includes retries, delays, and external IP management.
- Logs found credentials and failed attempts.

# Usage

```bash
python CpanelWalker.py <password_file> [--delay DELAY] [--retries RETRIES]
```

### Arguments

1. `<password_file>`: Path to the file containing passwords (e.g., `passwords.txt`). This is the main file used for brute-forcing.

### Optional Arguments

- `--delay DELAY`: Specify the delay (in seconds) between password attempts. Default is 1 second. You can increase it to avoid getting blocked by the server.

  Example:

```bash
  python CpanelWalker.py passwords.txt --delay 2
  ```

- `--retries RETRIES`: Number of retry attempts on failure to connect (default is 3). You can increase this if there are frequent connection issues.
  Example:
  ```bash
  python CpanelWalker.py passwords.txt --retries 5
  ```

# Features
### 1. Detect External IP
The script automatically detects the external IP address using `https://api.ipify.org`. You will be prompted whether to use this detected IP or enter a custom one.

### 2. Dynamic Proxy Fetching
The script fetches proxies dynamically using the ProxyScrape API:
- Protocol: `http`
- Timeout: 10,000 ms
- Country: All
- SSL: All
- Anonymity: All

The proxies are then randomly selected during brute-force attempts.

### 3. Random User-Agent Headers

To evade detection, the script uses a list of User-Agent strings simulating different browsers and operating systems.

### 4. Brute-force Password Attempts

For each username-password combination, the script attempts to log in by sending HTTP requests to the provided URL. It checks for the following HTTP status codes:

- **200**: Indicates success.
- **401**: Indicates failure (invalid credentials).

If valid credentials are found, they are saved to `found.txt`.

### 5. Retries and Delay

- **Retries**: If a proxy fails or the connection is lost, the script will retry up to the number specified by `--retries`.
- **Delay**: The script includes a delay between password attempts (adjustable via `--delay`).

# Sample Run

```bash
python CpanelWalker.py passwords.txt --delay 1.5 --retries 4
```

- **Password file**: `passwords.txt`
- **Delay**: 1.5 seconds between each attempt.
- **Retries**: 4 retries on failure.

# Output Files

1. **`found.txt`**: Stores the valid username-password combinations found during the brute-force attack.

   Example format:

   ```

   Username: admin, Password: 123456

   ```

# Error Handling

The script handles common connection errors such as:

- **ProxyError**

- **Timeout**

- **SSLError**

- **ConnectionError**

- **ChunkedEncodingError**

In case of such errors, the script will retry with a different proxy.

# Exit Codes

- `0`: Success (valid credentials found).

- `1`: Failure (no valid credentials found).

# Dependencies

- **Python 3.x**

- **`requests` library**: Used for sending HTTP requests and handling proxies. Install via:

  ```bash
  ```

```
pip install requests
```

# Contact

For further assistance or questions, feel free to contact the developer.