Directory and file enumeration of the Chameleon website

I used following python program to find hidden Directories in Chameleon website.

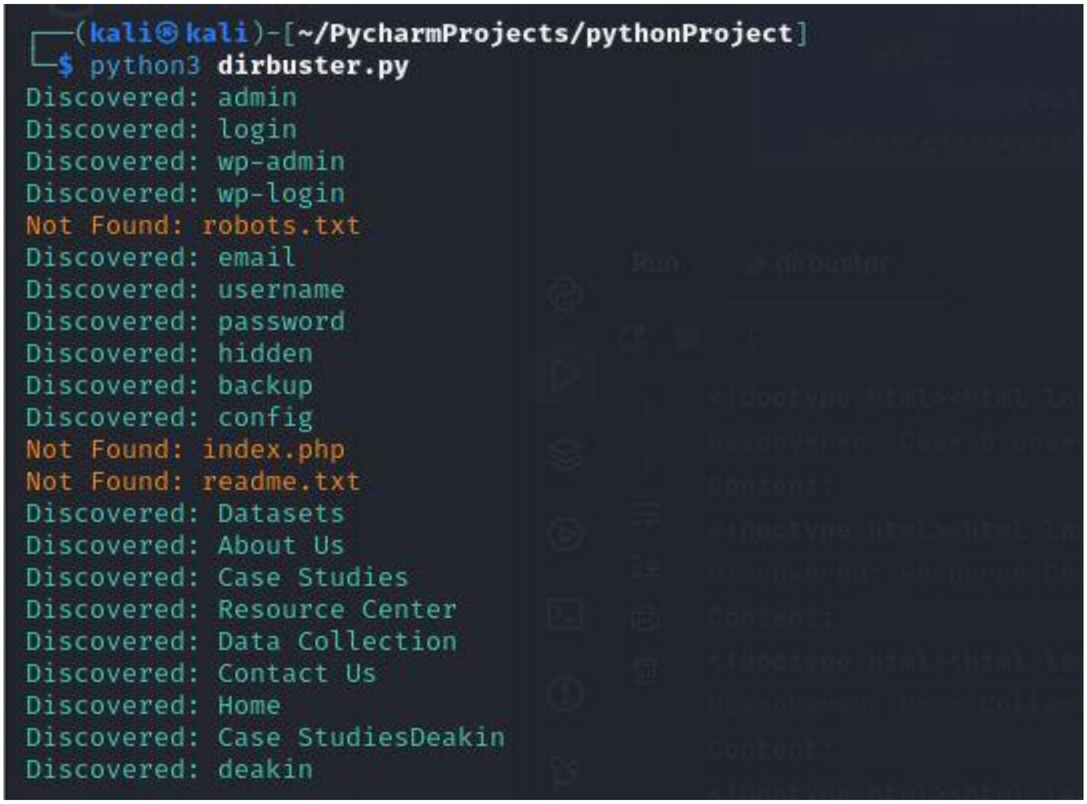
```
import requests

class colors:
    GREEN = '\033[92m'
    YELLOW = '\033[93m'
    RED = '\033[91m'
    RESET = '\033[0m'
def dirbuster(url, wordlist):
    discovered = []
    for word in wordlist:
        response = requests.get(url + '/' + word)
        if response.status_code == 200:
            discovered.append(word)
            print(colors.GREEN + "Discovered:", word + colors.RESET)
        elif response.status_code == 404:
            print(colors.YELLOW + "Not Found:", word + colors.RESET)
        else:
            print(colors.RED + "Error:", word + colors.RESET)
    return discovered

# Example wordlist containing common directory and file names
wordlist = [
    'admin', 'login', 'wp-admin', 'wp-login', 'robots.txt', 'email', 'username', 'password', 'hidden',
    'backup', 'config', 'index.php', 'readme.txt', 'Datasets', 'About Us', 'Case Studies', 'Resource Center',
'Data Collection', 'Contact Us', 'Home', 'Case Studies'
    'Deakin', 'deakin'
]

dirbuster('https://sit-chameleon-website-0bc2323.ts.r.appspot.com/', wordlist)
```

I got the following result.



```
┌──(kali㉿kali)-[~/PycharmProjects/pythonProject]
└─$ python3 dirbuster.py
Discovered: admin
Discovered: login
Discovered: wp-admin
Discovered: wp-login
Not Found: robots.txt
Discovered: email
Discovered: username
Discovered: password
Discovered: hidden
Discovered: backup
Discovered: config
Not Found: index.php
Not Found: readme.txt
Discovered: Datasets
Discovered: About Us
Discovered: Case Studies
Discovered: Resource Center
Discovered: Data Collection
Discovered: Contact Us
Discovered: Home
Discovered: Case StudiesDeakin
Discovered: deakin
```

Directory enumeration can reveal directories such as "admin," "login," "wp-admin," "wp-login," "email," "username," "password," and others, which may represent security problems. Here's how.

1. **Directory enumeration** discloses the directory structure of a web application or website. Attackers can utilize this information to pinpoint possible targets for further exploitation, such as well-known administrative interfaces like "admin" or login pages like "login."
2. **Brute Force Attacks:** Once attackers have identified directories such as "wp-admin" or "wp-login," they may try brute force credentials to gain unauthorized access to administrative interfaces or user accounts.
3. **Resource Enumeration:** Revealing data storage folders, such as "backup" or "config," can let attackers identify sensitive files or configurations that they can use to undermine the system's security.

To reduce these risks and prevent information leakage via directory enumeration, consider the following measures:

1. **Access Control:** Use appropriate access controls to limit access to sensitive directories and resources. Use authentication measures to prevent unauthorized users from accessing administrative interfaces or sensitive data.
2. **Directory Whitelisting:** Rather than depending exclusively on directory enumeration, use directory whitelisting to specify which directories should be accessible. By default, access to directories is denied, and only those required for the application's functionality are allowed.

3. **Custom Error Pages**: Set up custom error pages to display generic error messages rather than giving specific information like directory existence or accessibility. This helps to keep attackers from assessing the legitimacy of detected directories.
4. **Security by Obscurity:** While not a main defence approach, renaming important directories or adopting non-standard directory structures can make it more difficult for attackers to identify and target them.
5. **Conduct frequent security audits and vulnerability assessments**: to discover and address any potential security flaws, such as difficulties with directory enumeration and information exposure.

   Implementing these procedures reduces the risk of exploitation caused by directory enumeration and improves the overall security posture of your online application or website.