# ALBUS SECURITY

PRIVACY IS THE BEST FREEDOM

# PENETRATION-LIST

Become an Expert in Applications Security

THE VULNERABILITY IS LIFE, NEVER MESS WITH VULNERABILITIES.

**WRITTEN BY ANIKET TYAGI**

# Penetration-list

Penetration-List is the supplement for Bug Bounty Hunters


ANIKET TYAGI

"Destroy those feelings that demotivate you for your career"

ALBUS SECURITY

Privacy is the best Freedom

# Albus Security

At AlbusSecurity, our goal is to help other ambitious organizations and their people to succeed and for this, we all will work hard. Work with us as we give our 100% to our clients also soon we will launch our all services in our working website. It is our responsibility to grow our client's businesses and make their businesses secure. Our unwavering commitment to our clients makes it a success

In our Education Service, we provide education about all the subjects that come with cybersecurity. But How? We make blogs, and some repositories on Github, and also we upload some courses related to Cybersecurity, Programming, and Networking also so that service is available right now then you can check our Linkedin profile to get more information.

# About Author

Hello

I'm Aniket Tyagi and I'm an Information Technology Security officer at the 5f eco foundation of India, an Information Security Researcher, and the founder of Albus Security,

Also, I'm a CTF player on hackthebox, And I really love exploring new things and playing with Web applications, Mobile Security. and Networking devices like cisco catalyst 2960-XR, 2960, and Cisco 4321 Integrated-Service-Router(ISR).

Something I'll do developing things like tools, web applications, android applications, and software.

# INTRODUCTION TO PENETRATION-LIST

The Penetration-list has become a dream of every bug bounty hunter. Because The reason to make this is that I just want one list that contains a theory section with steps to find vulnerabilities. So for the theory section, I've decided that I'll make a book, However, In that book, We'll learn about vulnerabilities in very depth, which Mean We'll go zero to hero level about vulnerability, Then what about materials that we use to find a vulnerability like Payload, Malicious files, fuzzing lists, dorks lists and Malicious scripts that were used to find a vulnerability in your Target. So for this material part, I'll upload materials on the GitHub repository.

# 1

# Information Disclosure

In this chapter, you'll learn about Information Disclosure Vulnerability, Also You'll learn how We'll find this vulnerability in your target

Without further do let's get started.

# Information-Disclosure-Vulnerability

## What Is Information Disclosure?

Information disclosure is when there is an information leakage or in simple words when a website unpremeditatedly reveals sensitive information to its users. Websites may leak all kinds of information to an attacker.

But then the question arises that what kind of information is revealed by the website?

Ans:- 1. Data about other users, such as usernames or financial information.

2. Sensitive commercial or business data.

3. Technical details about the website like application version, Plugin names, and also its infrastructure.

The dangers of leaking user or business data aren't always a bad situation but disclosing technical information can sometimes be serious. Why? Because it can potentially work as a starting point for planning an additional attack surface that may contain other amazing vulnerabilities. The knowledge that you get

from Information disclosure could even help you to find some critical vulnerabilities however, an attacker needs to bring out the information disclosure by interacting with the website in a hunting way. They will then carefully study the website's responses and try to identify the interesting behavior of the application.

Follow these steps to find information disclosure:-

1. Directory brute forcing
2. Information-Disclosure through error
3. Google Dorking
4. Shodan Dorking
5. GitHub Dorking

## Directory-brute forcing:-

A brute force attack is a hacking method that uses a trial and error method to crack passwords hash, login credentials, and encryption keys. An attacker can use the brute-force technique to find hidden web files and directories on a web-sever. There are several tools for doing this, but all tools use a list of sensitive files. So for this, we upload the list of names of some most sensitive files, but the question is about how that list will help you? So, I'm telling my own experience that how I use the list to find an information disclosure.

*Find Hidden files with the help of burp suite?*

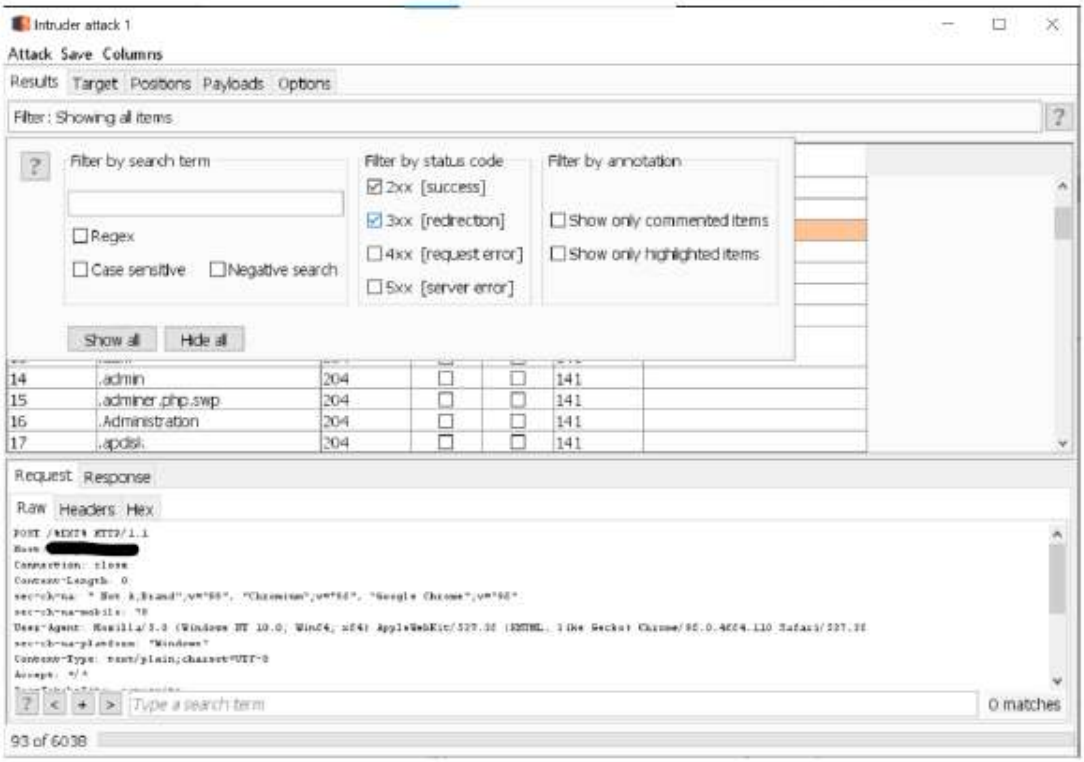
1. Open your burp-suite, intercept a request then type any random words on the file path.


2. Send the request to Burp Intruder.

Go to Burp Intruder. On the "Positions" tab, clear all the default payload positions and set the random words, So that will be your injecting point and Intruder should be used in Sniper mode.

3. On the payload tab, Now Select a Simple list on payload type and then load our sensitive file list.

4. Start Attack, Now turning point is that the Burpsuite will provide us with the filter options which means you could customize your response by clicking the "filter by status code", then for this, you disable ticks on 4x 5x then you will only see those requests that will give you response 200 301.

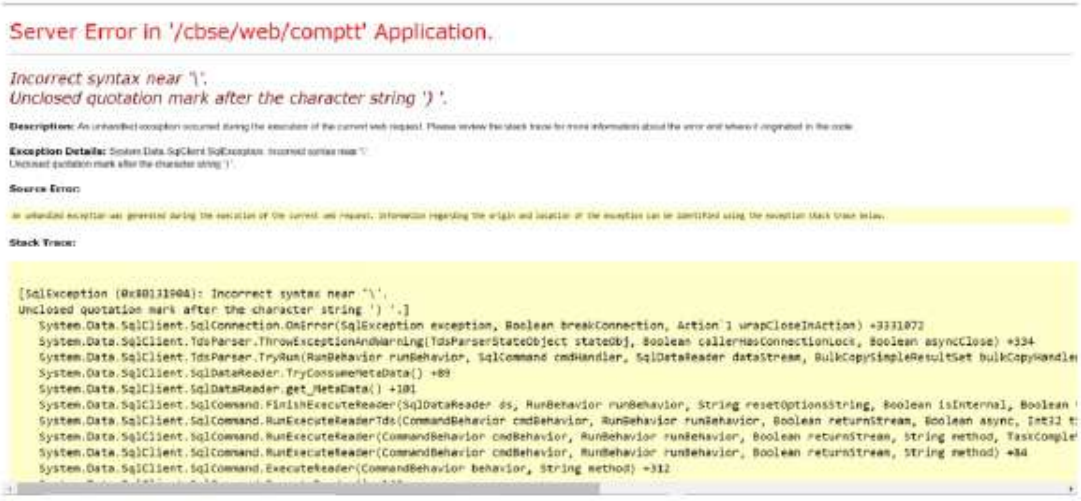



You can also use Some Command-line Interface Tool like dirsearch, go- buster,ffuf, etc dirsearch is also good. Why I'm not telling you to use dirsearch or any other tool is because these tools also use sensitive file lists in the backend, As per circumstances we make one list from it and remove unwanted file names and add some more files names to the list to make it unique. So Please check our Penetration list on Github.

## Information Disclosure Through Error:-

We already know about Error but "How did it occur?" the reason is that the Developer does not configure the website and function appropriately so the website displays a verbose error message in response. But "How do we force a website to generate an error?"

1. Add some malicious input like this "?>?}|@#$%'." So there is a chance that the application will give you an error, for example- If I add some malicious input parameter on the CBSE website then it gives me an error message.




2. The second way is to Intercept your Request and then remove some parameters, then if the server is not configured probably it will give you an error in your response, For example-

months ago I reported the same vulnerability in bug-crowd, unfortunately, it duplicated.

## Google Dorking:-

Google Dorking is one of the famous techniques to filter your search, But "how do we just add some queries in our input?". Those queries are also known as dorks. This will help you to find whatever you want from the deep ocean of the internet. For example:- Assuming that the Developer may not have applied some security rules, you can be looking for some confidential files to a particular website, those files which are not intended for the public of the Internet. Due to some misconfiguration when you use some special queries in your search for that website, you can get to that particular file in your search. Now, We will learn about some important dorks for information disclosure these all dorks have followed the same syntax like- {**dorkname: parameter}**.

First, we will learn basic dorks then secondly we will learn about advanced dorks to find information disclosure.

## Basic dork:-

**site: {domain}** → search results will contain files and pages in the domain specified by *the domain name, For example,* → ***" site: target.com"***.

*if you want subdomains of your target then* this dork can be used

to → "**site: *.*target.com***"

**intitle: {words}** → the search results will contain the word specified by you in the intitle, For example, → "**target.com intitle: admin login**".

**inurl: {word}** → the search results will contain in the URL the *word* after the dork but it is only useful for one single keyword, For example, →"**site:target.com inurl: ?id=".**

**intext: {word}** →the search result will be used to locate a specific word within the returned pages that is entered in intext, For example, → "**site: target.com intext: login".**

**filetype: {type}** →search results will contain files of the extension specified by *type, For example* , → ***site: target.com filetype: pdf | csv | bak | wab.***

## Advanced dorks:-

**allintext: {word}** → searches for specific text contained on web page. For example, → **site: target.com allintext: "Incorrect syntax near"**.

**allinurl: {word}** → this is exactly the same as inurl: but it is used to fetch results whose URL contains all the specified characters, For example, → **site: target.com allinurl: "client area".**

**+{word}** → use this operator to add one more specific word in your query. For example, → **"site:target.com intext: Login**

**+password".**

- **{word}** → use this operator to remove specific words in your query. For example → **site: target.com allintext: "Adminlogs" -403**.

**{word} | {word}** → use this operator to find multiple terms in your result, For example, → **"site: *.target.com -intext:username | password | logs"**.

We also have a Google dork list in our Github profile, To get this please visit our Penetration-list repository.

## Shodan Dorking:-

Shodan is a gigantic search engine that gives you information about specific types of computers connected to the internet using a variety of Dorks. Shodan dorks are the first step to success for penetration testing because it is not only used to get a piece of information about web servers but are also used to get information about any kind of internet-connected devices as it helps identify vulnerable systems. Shodan dork is harder than google, But it gives you a lot of information about your target. With the help of these dorks, you can find an Information disclosure. But first, you need to learn about some dorks so I will tell you some dorks that will help you to find information disclosure, all dorks have to follow the same syntax that we learn in google Dorking.

Some Shodan dorks:-

**ssl: {word}** → the search results will contain ssl certificate information about your target, For example, → **ssl.cert.subject.cn:"target.com".**

**hostname: {word}** → the search results will contain some values that match the hostname, for example, → **hostname: "target.com".**

**port: {word}** → use this dork to search for a specific port in your query, For example, → **hostname: "target.com" port:"11211" product:" Memcached".Note:-** port 11211 is Basically used for Memcached

**http.html: {word}** → the search result will contain your provided keyword in the entire Html page of your target, For example, → **"ssl: target.com port: 2082 http.html: WordPress"**.

**org:{word}** →the search results will contain different organizations in your query, For example, → **hostname:"target.com" org:"Apache httpd" http.html: ": username, password"**. You also use **product: {word}** dork to find what service is used by the target, For example, →

**ssl.cert.subject.CN:"target.com" http.html:"Login, username, password", "Admin" 200 -http.html:" Not Found"**.

We also have a Shodan dork list in our Github profile, To get this please visit our Penetration-list repository.

## GitHub Dorking:-

First, I would like to explain to you about GitHub, then How we use GitHub normally in our day-to-day life, then we'll learn how we use GitHub to find information disclosure. In GitHub, hackers and developers share their knowledge means sharing their codes with others, and also sometimes people publish some other material like cheatsheets for testing, material lists, etc. Where Git is a command-line interface tool and Github is a WBG(Web-BASED-GRAPHICAL interface) but it's not our topic that we discuss GitHub in deep. I'll just explain to you about GitHub in short, Now How Normally we use GitHub in our day-to-day life, for example, if you want to use the Github-docker tool on your machine so you should need to know how? Answer- Firstly you go to the developer repository where he uploads the code of the tool then you can clone the tool from their repository for use. Now the question arises that how we use GitHub to find some sensitive information about our target. Answer — In Simple words, Company developers share their code and they leave sometimes strange things about your target on their code then publish their code publicly, but what do you mean by strange things like API_KEYS, Password, ACCESS_TOKEN, etc, However, Google dork, shodan dork can be used to scan websites for sensitive data. But GitHub dork is used for critical data such as usernames and passwords, database credentials, API data, cryptographic keys, etc.

I hope you got a little bit of understanding about Github. Now it's time to discuss how we extract sensitive information from our target. You must need to add some special's keywords to your

search to extract information about your target. for example, **"org: target.com language: bash ftp"** and you can also directly use → **"target.com language: bash ftp"**.

Github-Dorks:-

***org: {word}*** → the search results will contain your entered organization repo that makes your attack surface small but you neglect this also in your search just enter your target name. For example, → **"org: target.com".**

***filename: {word}*** → search results will contain files of the extension specified by word. For example, → **"target.com filename:.netrc password".**

***NOT {word}*** → use this word to remove specific words in your query,

For example, →

**"target.com Password twilio_api_key NOT FAKE NOT TEST NOT prod. secret.exs".**

You can add lots of dorks in your search.

**— — — — — — — — Some special dorks — — — — — — —**

bintray_key
AWSSecretKey
digitalocean_ssh_key_body
npm_api_key

sqsaccesskey
mailgun_password
github_deploy_hb_doc_pass
filename:.env
filename:prod.exs
filename:.npmrc _auth
ilename:WebServers.xml
msg
nickserv
identify
filename:config
path: sites
databases
password private
filename:passed
path:etc
sandbox_aws_secret_access_key
filename:vim_settings.xml
filename:sftp.json
path:vscode
filename:secrets.yml
filename:configuration.php

# Contact Us

## TALK TO US

***Phone no:-***

***+91 7983001181***
***+91 9027035367***

***Drop a mail:-***

***as745591@gmail.com***
***prakrati2004@gmail.com***

***Linkedin:-***

***https://www.linkedin.com/in/aniket-tyagi-cyber-world/***