VIT-AP UNIVERSITY

# CYBER SECURITY PROJECT

## FINAL REPORT

## PASSWORD ATTACKS

Using CeWL, Hashcat, Burp suite tools

**Submitted by**

Vedantha Suddula

20BCN7115

**Submitted to**

Dr.Shaik Kareemulla

**ABSTRACT:**

Password attacks are one of the most common ways for threat actors to gain access to your data, by stealing your password. Now in this project we do password attacks using CeWL, Burp suite, Hashcat. Password attacks involve exploiting a broken authorization vulnerability in the system combined with automatic password attack tools that speed up the guessing and cracking passwords. The attacker uses various techniques to access and exposethe credentials of a legitimate user, assuming their identity and privileges. The username-password combination is one of the oldest known account authenticationtechniques, so adversaries have had time to craft multiple methods of obtaining guessable passwords. Additionally, applications that use passwords as the sole authentication factor are vulnerable to password attacks since the vulnerabilities are well understood.

Weaker passwords have been a serious issue in today's world of cybercrime. By the estimation from various sources, it can be clear that any weaker password that is simply alphabets and less
than 8 characters can be cracked within 15 minutes of time depending upon the CPU and GPU that a person has. The report demonstrates why weaker passwords in WPA2-PSK at home Wi-Fi networks is a nightmare. A live demonstration inside a lab environment is done showing one of the password attacks, dictionary attacks. However, a solution to mitigate this problem is also documented in this report to make readers aware of the threats that they may invite at some point in their life.

# INDEX

## INTRODUCTION:

Most password-cracking programs are only as good as the wordlist that you provide them. Brute-force password cracking is very tedious and time consuming, but if you can find an appropriate and well-designed wordlist that is specific to the user whose password you are trying to crack, you can save yourself hours, maybe even days of password cracking.

Password attacks are one of the most common ways for threat actors to gain access to your data, by stealing your password. Now in this project we do password attacks using CeWL, Burp suite, Hashcat.

CeWl tool is great at creating wordlists based upon a set of rules such as the number of characters, the character set, etc., but doesn't enable us to choose a wordlist thatis particular to a business or industry or interests. We humans are not always very creative and often fall victim to the familiar, especially when generating passwords. If we understand that, it can be helpful to finding potential passwords and generating a relevant password list.

For instance, employees at a construction company are more likely to use words for passwords that are used in their industry, such as lumber, girder, build, soffit, eave, etc. People in the drug industry are more likely to have passwords such as prescription, drug, narcotic, barbiturate, etc. You get the idea.

## CEWL TOOL:

CeWL is a Ruby program that crawls a URL to a defined depth, optionally following external links, and produces a list of keywords that password crackers such as John the Ripper can use to crack passwords. can. FAB (Files Already Bagged) is a command-line program that generates author/producer lists from already downloaded files using information extraction algorithms similar to CeWL.

It is custom wordlist generator and ruby program that crawls a specific URL to a defined depth and returns a list of keywords, which password crackers like John the Ripper, Medusa, and WFuzz can use to crack the passwords. CeWL also has an associated command-line app FAB, which uses the same metadata extraction techniques to generate author/producer lists from already downloaded files using information extraction algorithms like CeWL.

CeWL comes preinstalled with Kali Linux. With this tool, we can easily collect words and phrases from the target page. It is a robust program that can quickly scrape the webserver of any website.

Open the terminal of Kali Linux and type “cewl -h” to see the lists of all the options it accepts, with a complete description.

## BURP SUITE:

Burp Suite is an integrated platform for performing security testing of web applications. Its various tools work seamlessly together to support the entire testing process, from initial mapping and analysis of an application's attack surface, through to finding and exploiting security vulnerabilities.

Burp gives you full control, letting you combine advanced manual techniques with state-of-the-art automation, to make your work faster, more effective, and more fun. Importantly, it gives us another way to manage our attacks as the alternative to Metasploit.

Brute force plays a vital role in web penetration testing because is the simplest method to gain access to a site or server by checking the correct username or password by calculating every possible combination that could generate a username or password.

## HASHCAT:

Hashcat is a popular and effective password cracker widely used by both penetration testers and sysadmins as well as criminals and spies.

Cracking passwords is different from guessing a web login password, which typically only allows a small number of guesses before locking your account. Instead, someone who has gained access to a system with encrypted passwords ("hashes") will often try to crack those hashes to recover those passwords.

Cracking passwords has many legitimate uses, besides the obvious criminal and espionage ones. A sysadmin may wish to pre-emptively check the security of user passwords. If hashcat can crack them, so can an attacker.

## BACKGROUND:

Password attacks involve exploiting a broken authorization vulnerability in the system combined with automatic password attack tools that speed up the guessing and cracking passwords. The attacker uses various techniques to access and expose the credentials of a legitimate user, assuming their identity and privileges. The username-password combination is one of the oldest known account authentication techniques, so adversaries have had time to craft multiple methods of obtaining guessable passwords. Additionally, applications that use passwords as the sole authentication factor are vulnerable to password attacks since the

vulnerabilities are well understood**.**

This type of password attack employs trial-and-error methods to guess a user's authentication information. The bad actor uses automated scripts to work through as many permutations as possible to guess the user's password correctly. While it is a relatively old method that requires a lot of patience and time, a Brute force attack is still standard in account breach attempts since they are automated and straightforward.

## Problem Definition:

Password attacks involve exploiting a broken authorization vulnerability in the system combined with automatic password attack tools that speed up the guessing and cracking passwords. The attacker uses various techniques to access and expose the credentials of a legitimate user, assuming their identity and privileges. The username-password combination is one of the oldest known account authentication techniques, so adversaries have had time to craft multiple methods of obtaining guessable passwords. Additionally, applications that use passwords as the sole authentication factor are vulnerable to password attacks since the vulnerabilities are well understood.

Password attacks have far-reaching consequences since malicious users only require unauthorized access to a single privileged account or a few users accounts to compromise the web application. Depending on the data hosted by the application, compromised passwords can pave the way for the exposure of sensitive information, distributed denial of service, financial fraud, and other sophisticated attacks.

## Objectives:

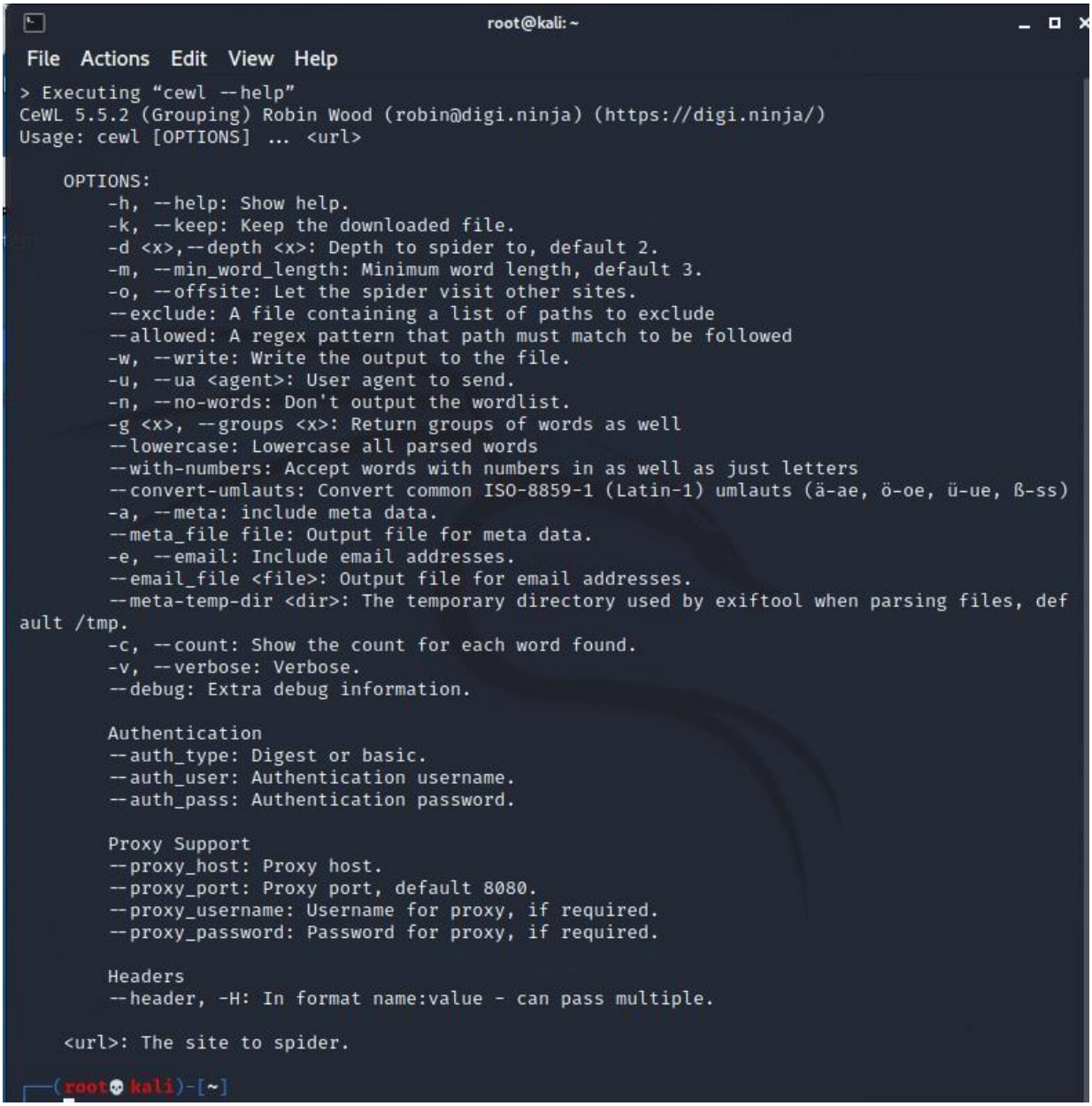
The aim of this report is to show readers how the implementation of a weaker and common guessable password using the following tools.
To fulfill the aim, certain scenarios are made to show how it is a blunder mistake. The
objectives to fulfill the aim of this report are:

- Selecting a wireless security protocol.
- Using weaker password for the demonstration purpose.
- Performing attack to crack the password(using cewl,burpsuite,hashcat)

After the attack is successfully carried out, in this report, a way to mitigate the risk is also
shown and verified that the mitigation technique works.

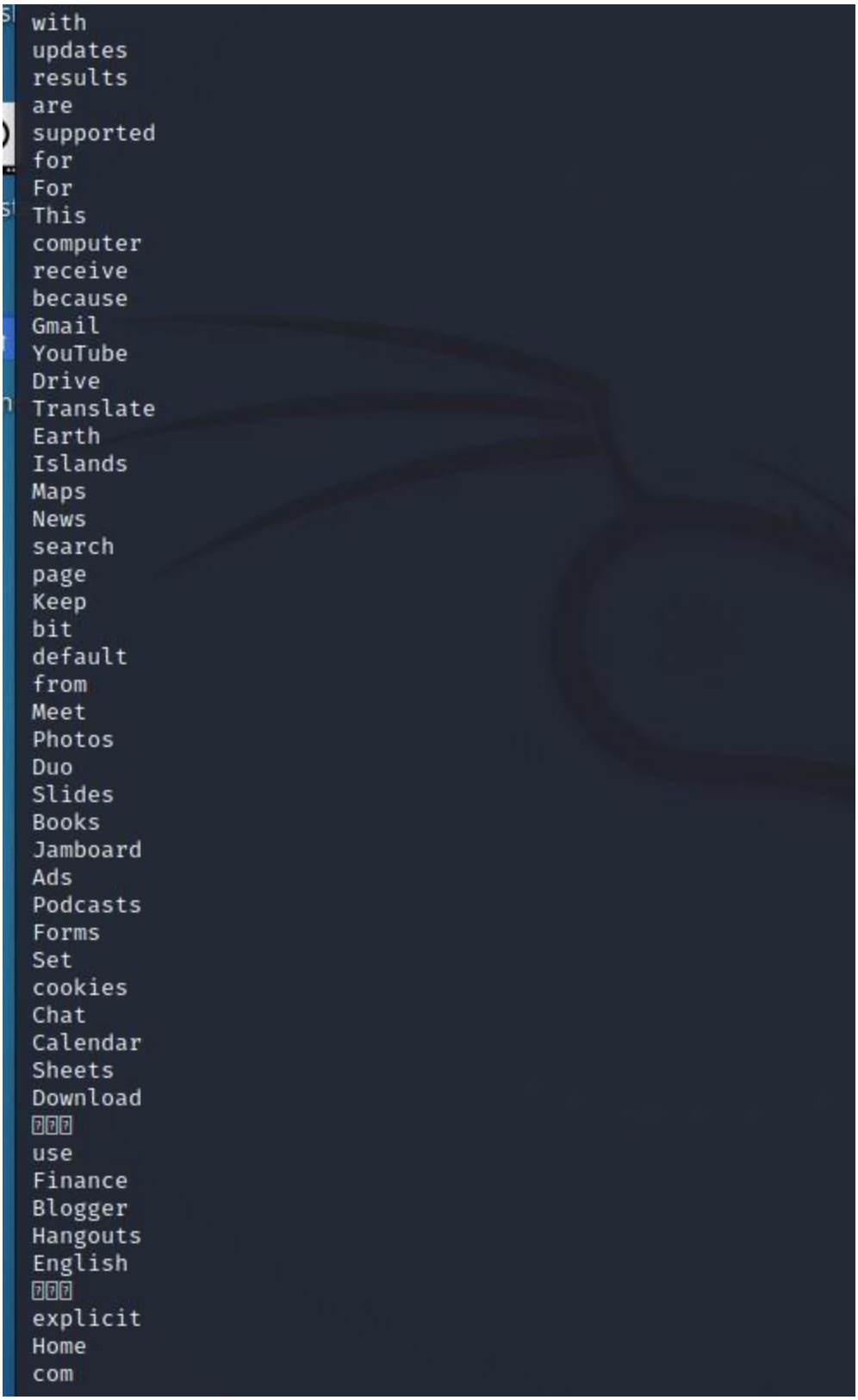
**Procedure:**

Command-1: cewl –help



Command-2: cewl website address

The given URL to a specified depth and prints a list of words which can then be used as a dictionary for cracking the password


with
updates
results
are
supported
for
For
This
computer
receive
because
Gmail
YouTube
Drive
Translate
Earth
Islands
Maps
News
search
page
Keep
bit
default
from
Meet
Photos
Duo
Slides
Books
Jamboard
Ads
Podcasts
Forms
Set
cookies
Chat
Calendar
Sheets
Download
use
Finance
Blogger
Hangouts
English
explicit
Home
com

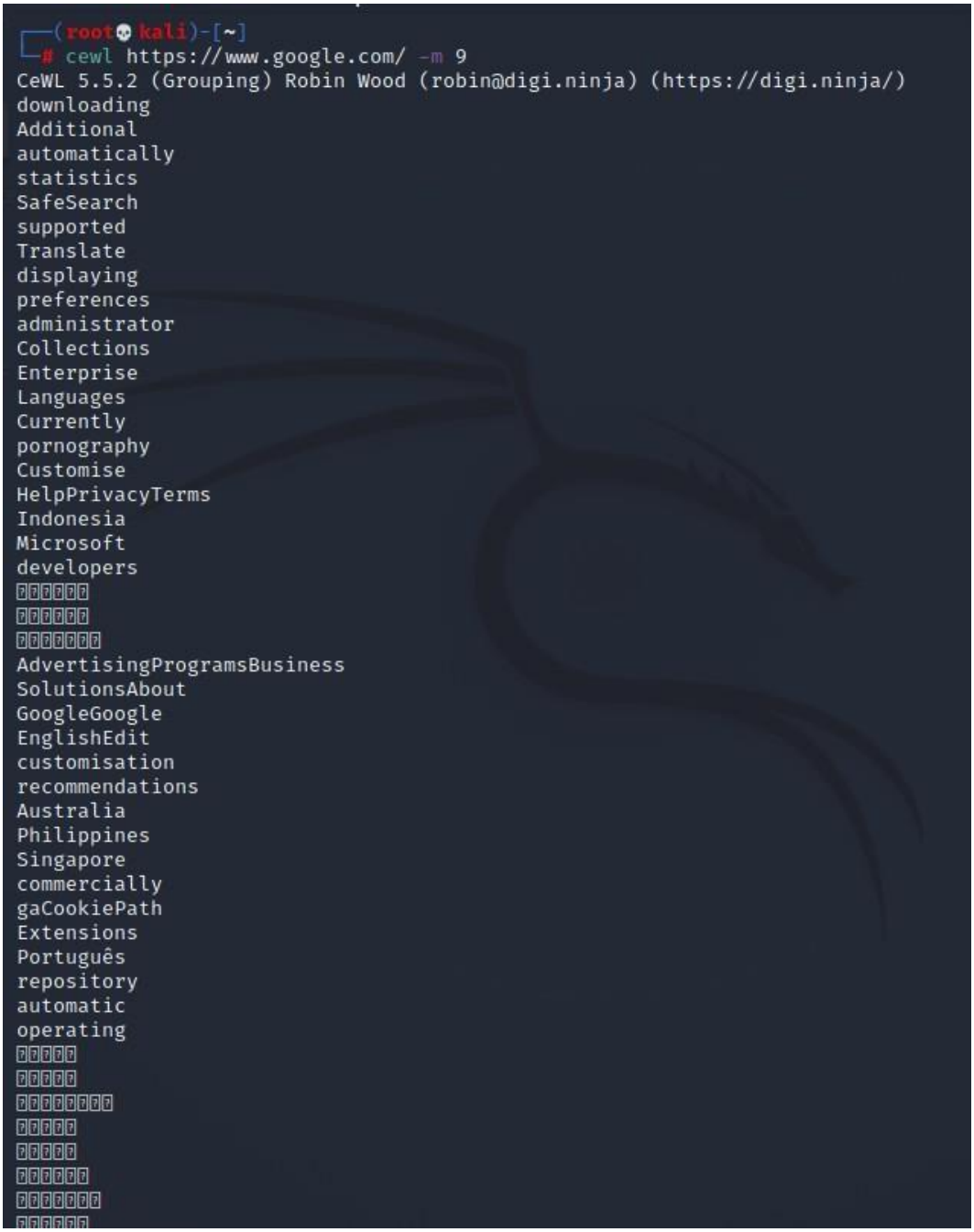
Command-3: cewl website address -w dict.txt

For the purpose of record maintenance, better readability, and future references, we save the print list of the word onto a file. To this we will use the parameter -w to save the output in a text file.

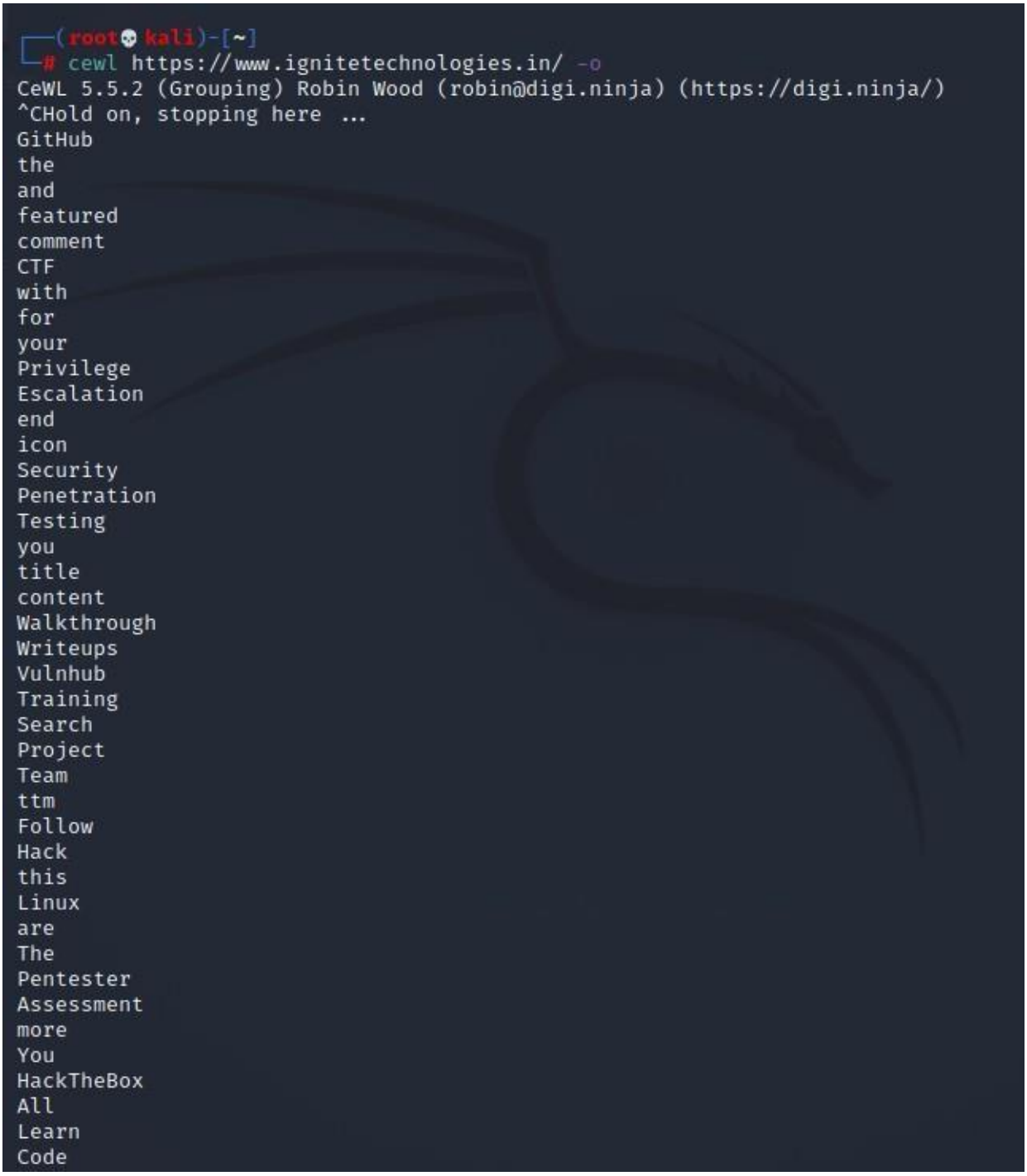

To display the output written in dict file.

Command-4: cewl website address -m9

If you want to generate a wordlist of a specific word length then use -m option as it enables minimum words to limit parameters.



Command-5: cewl webserver name -0

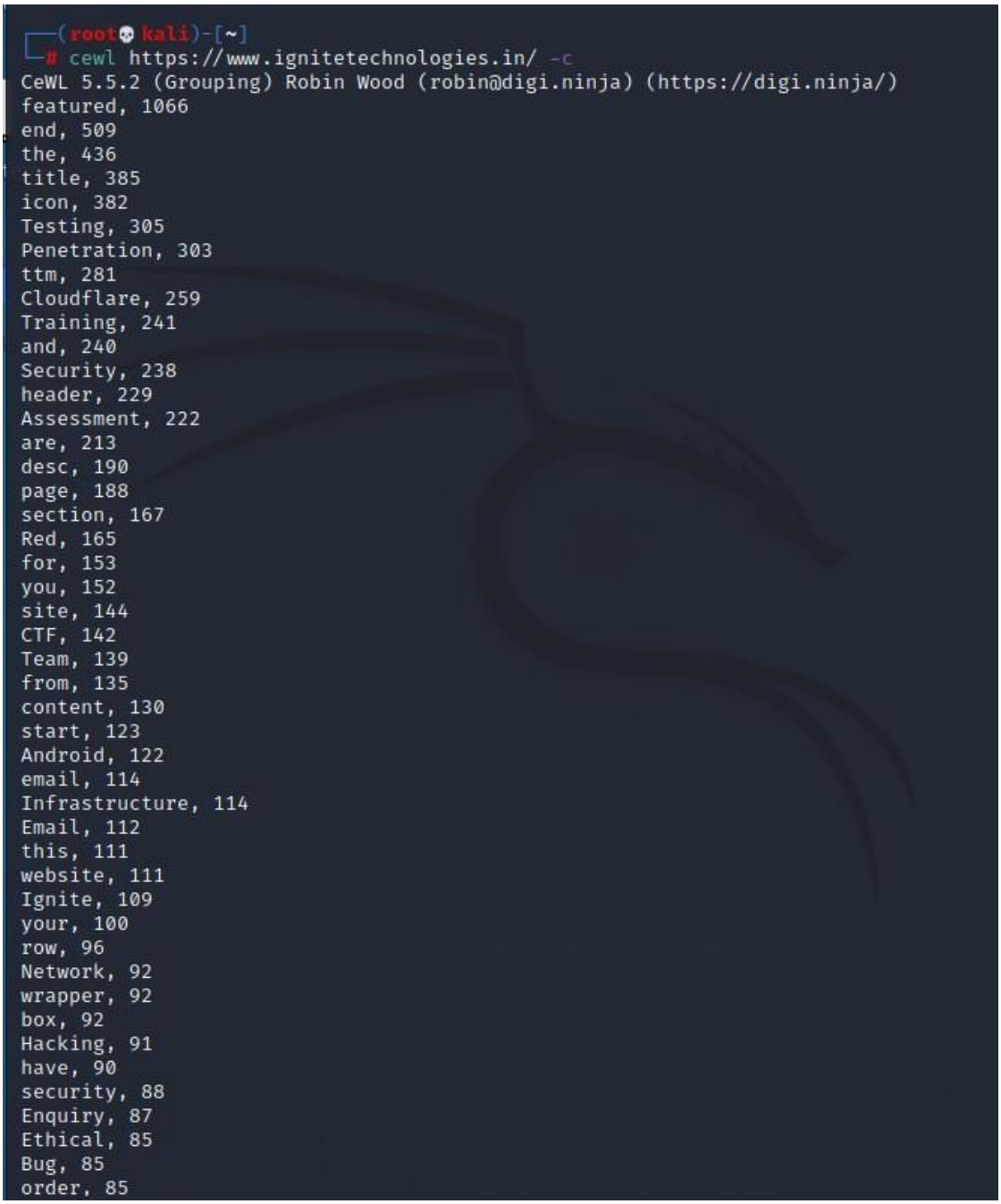


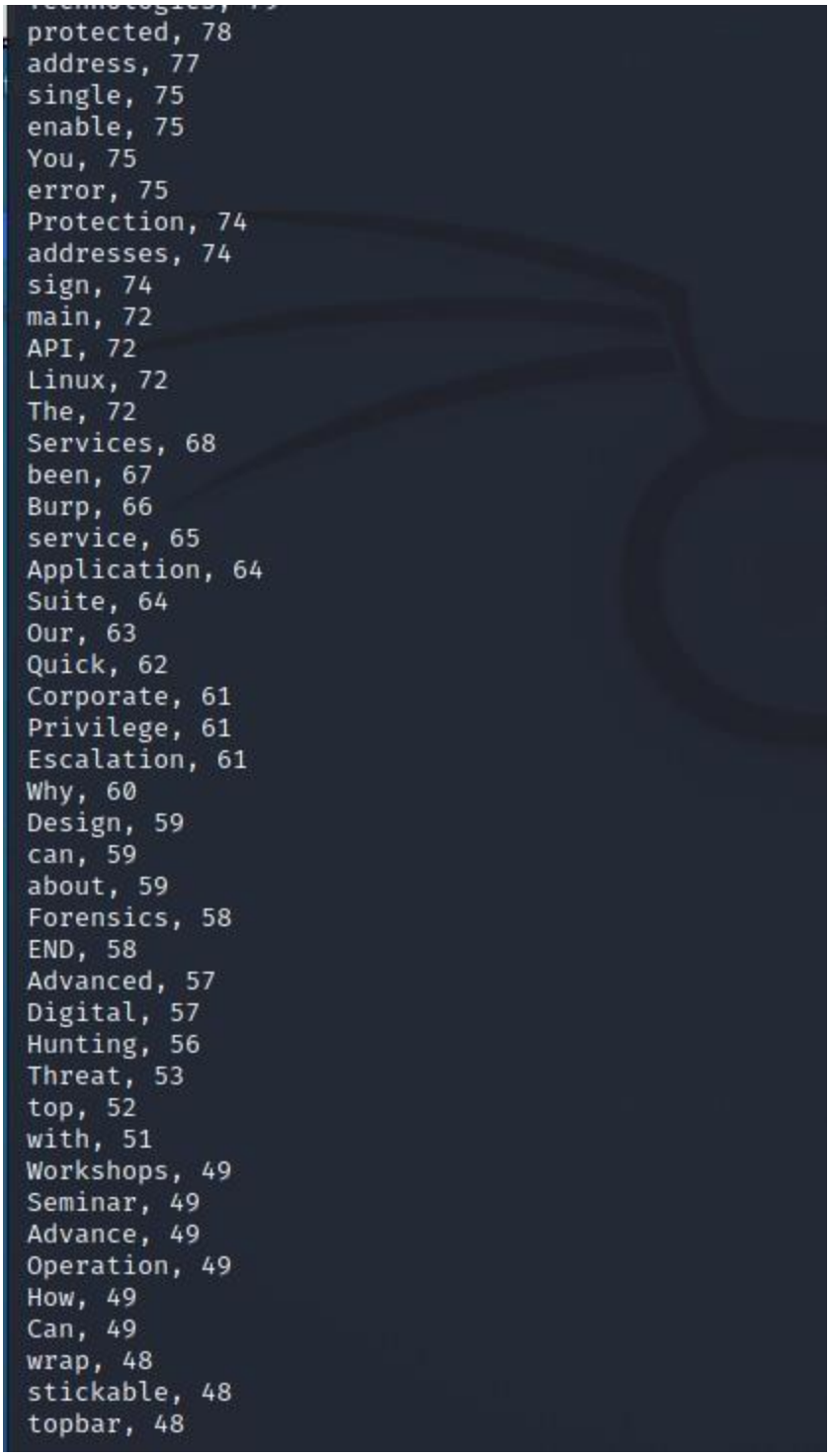
Command-6: cewl webserver -n -e

You can use **-e option** that enables email parameter along with **-n option** that hides the list of the word generated while crawling the given website.

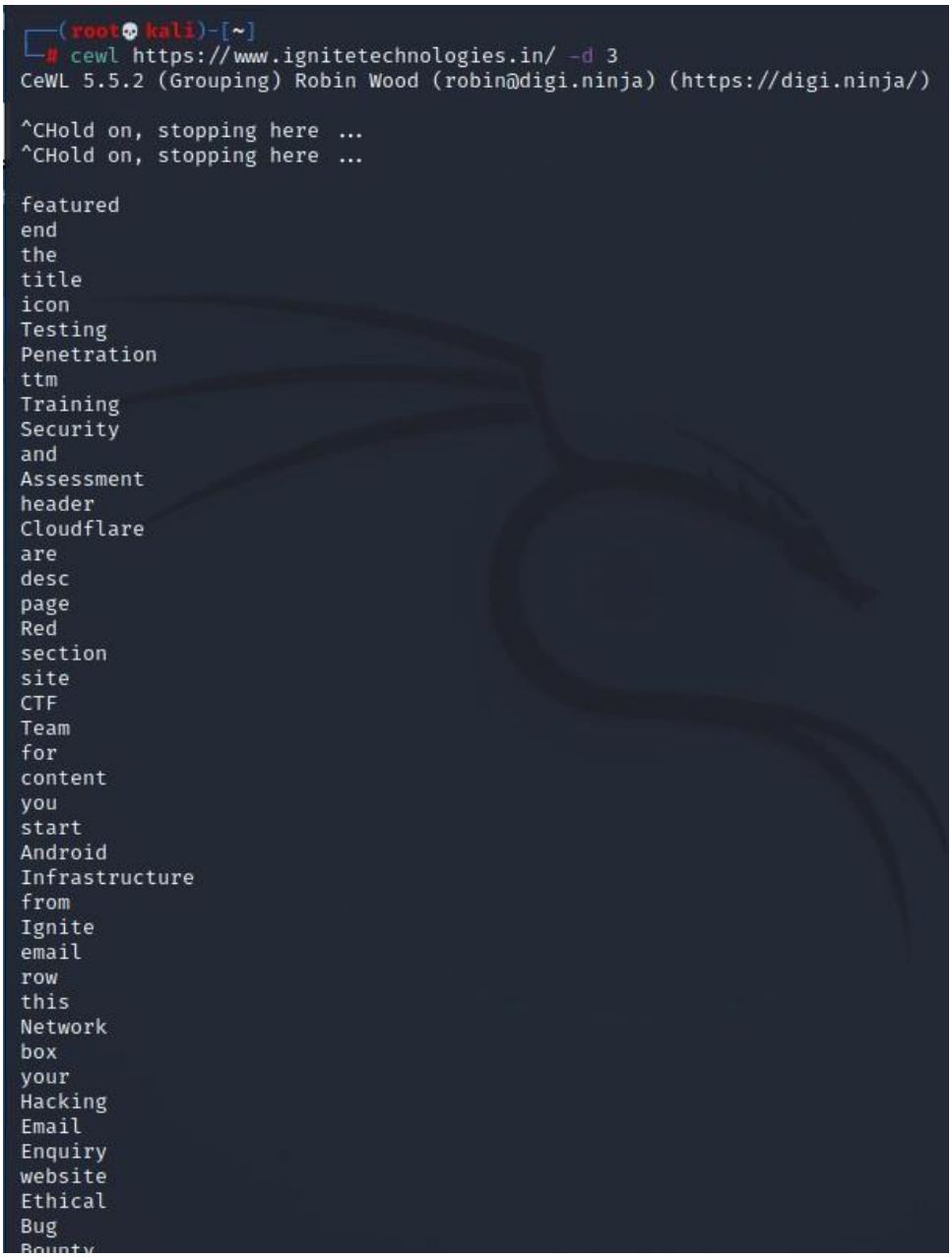
Command-7: cewl webserver name -c

If you want to count the number of words repeated several times in a website,then use **-c options** that enable count parameter.




protected, 78
address, 77
single, 75
enable, 75
You, 75
error, 75
Protection, 74
addresses, 74
sign, 74
main, 72
API, 72
Linux, 72
The, 72
Services, 68
been, 67
Burp, 66
service, 65
Application, 64
Suite, 64
Our, 63
Quick, 62
Corporate, 61
Privilege, 61
Escalation, 61
Why, 60
Design, 59
can, 59
about, 59
Forensics, 58
END, 58
Advanced, 57
Digital, 57
Hunting, 56
Threat, 53
top, 52
with, 51
Workshops, 49
Seminar, 49
Advance, 49
Operation, 49
How, 49
Can, 49
wrap, 48
stickable, 48
topbar, 48

Command-8: cewl webserver name -d depthno




If you want to increase the level of spider for generating a larger list of the word by enumerating more new words from the website then use **-d option** along withdepth level number that enables depth parameter for making more intense creeping. By Default it the depth level set is 2.

Command-9: cewl webserver –debug

You can use –debug option that enables debug mode and shows error and raw detail of the website while crawling.

```
┌──(root💀kali)-[~]
└─# cewl https://www.ignitetechnologies.in/ --debug
CeWL 5.5.2 (Grouping) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
Pushing {nil⇒"https://www.ignitetechnologies.in/"}
Checking page https://www.ignitetechnologies.in/
Comparing https://www.ignitetechnologies.in/ with https://www.ignitetechnologies.in/
<!DOCTYPE html>
<html lang="en">

<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="keywords" content="Ethical Hacking | Bug Bounty Training | Android Pentest| Net
work Pentest | CTF Challenges | Red
        Teaming | Infrastructure Security Assessment |Digital Forensic | Corporate Training | S
eminar &amp;
        Workshops">
    <meta name="description" content="Ignite Technologies is a worldwide name in the Informatio
n Technology field. As we provide high-
quality cybersecurity training and consulting services that fulfil students, government and cor
porate
requirements. The training course contains 20+ advanced snippets of all modules and a hyper-
realistic (Cyber Range) virtual lab which allows individuals to combat cyber-attacks in a contr
olled
environment in real situations.">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
    <title>Ignite Technologies – Ethical Hacking | Bug Bounty Training | Red Teaming | Network
Pentest | CTF Challenges</title>

    <!— favicon icon —>
    <link rel="shortcut icon" href="images/favicon.png">

    <!— bootstrap —>
    <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">

    <!— animate —>
    <link rel="stylesheet" type="text/css" href="css/animate.css">

    <!— owl-carousel —>
    <link rel="stylesheet" type="text/css" href="css/owl.carousel.css">

    <!— fontawesome —>
    <link rel="stylesheet" type="text/css" href="css/font-awesome.css">

    <!— themify —>
    <link rel="stylesheet" type="text/css" href="css/themify-icons.css">
```

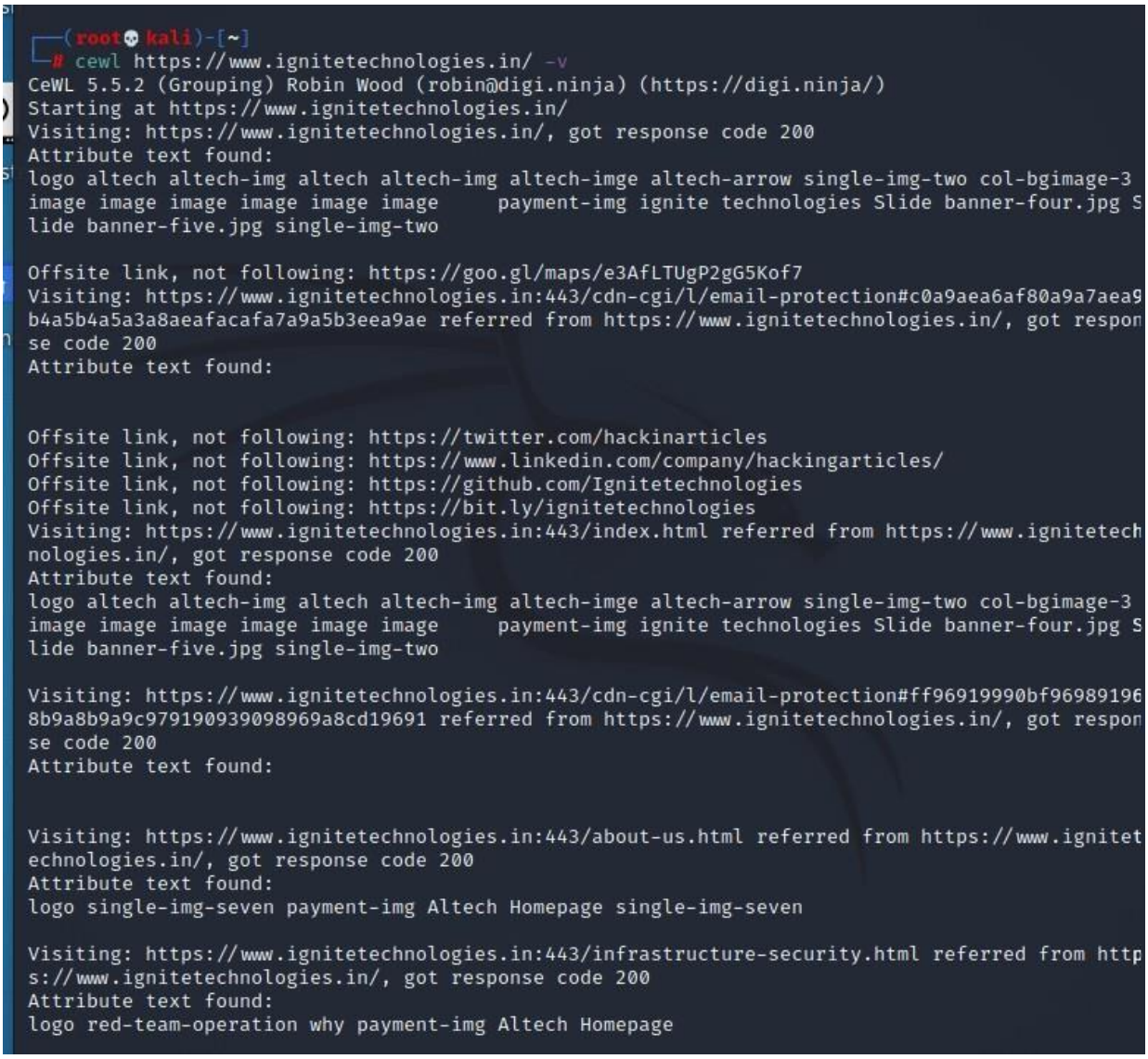
```
Checking page https://www.linkedin.com/company/hackingarticles
Comparing https://www.linkedin.com/company/hackingarticles with https://www.ignitetechnologies.
in/
Checking page https://www.linkedin.com/company/hackingarticles
Comparing https://www.linkedin.com/company/hackingarticles with https://www.ignitetechnologies.
in/
Checking page https://github.com/Ignitetechnologies
Comparing https://github.com/Ignitetechnologies with https://www.ignitetechnologies.in/
Checking page https://github.com/Ignitetechnologies
Comparing https://github.com/Ignitetechnologies with https://www.ignitetechnologies.in/
Checking page https://github.com/Ignitetechnologies
Comparing https://github.com/Ignitetechnologies with https://www.ignitetechnologies.in/
End of main loop
featured
end
the
title
icon
Testing
Penetration
ttm
Cloudflare
Training
and
Security
header
Assessment
are
desc
page
section
Red
for
you
site
CTF
Team
from
content
start
Android
email
Infrastructure
Email
this
```

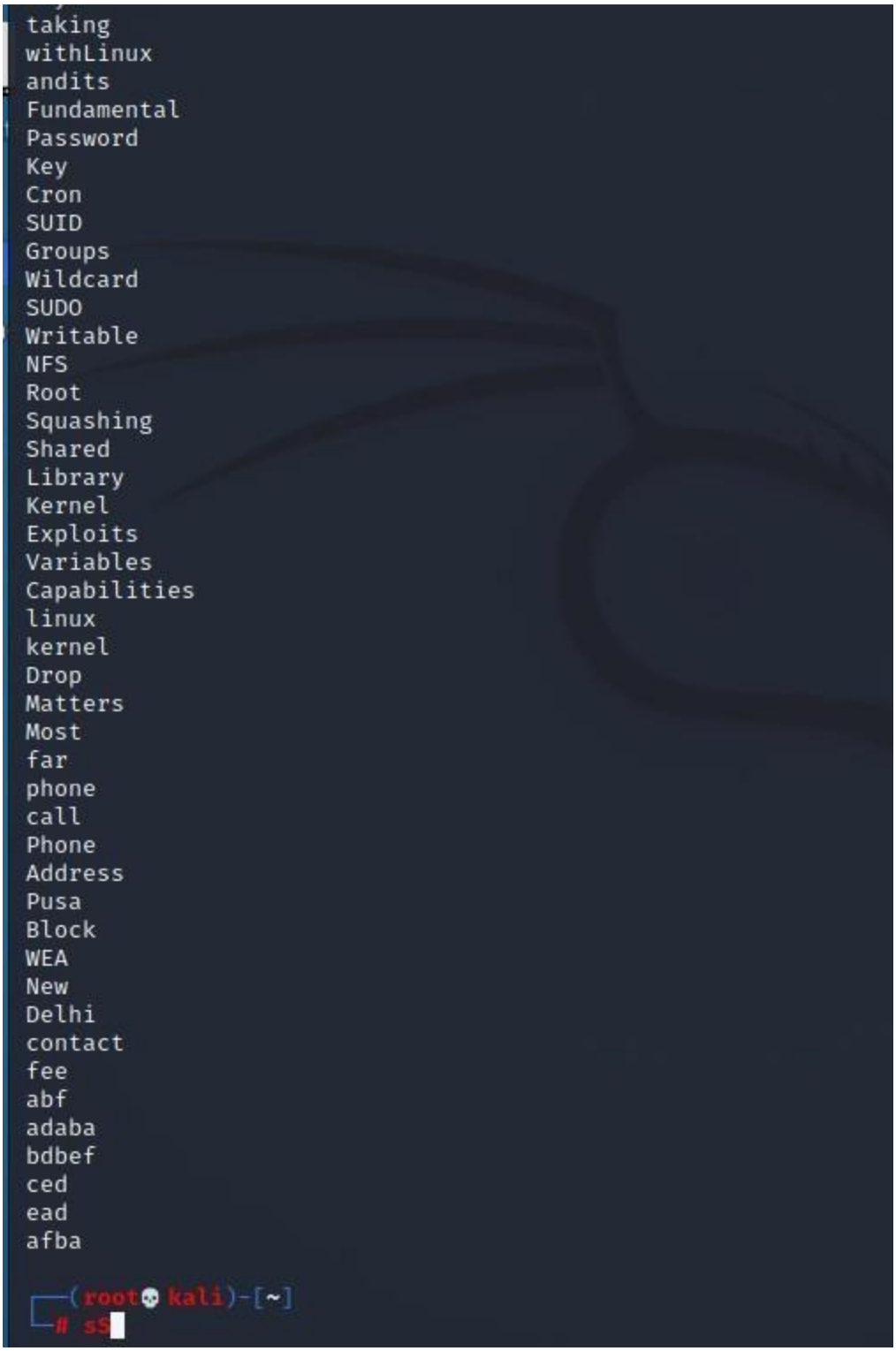
```
Pusa
Block
WEA
New
Delhi
contact
fdcb
ded
bcc
dec
efb
dfb
aff
cbcb
afbeb
eda
cdf
End of wordlist loop
End of email loop
End of meta loop

┌──(root💀kali)-[~]
└─# cewl https://www.ignitetechnologies.in/ --debug
```

Command-10: cewl webserver name -v

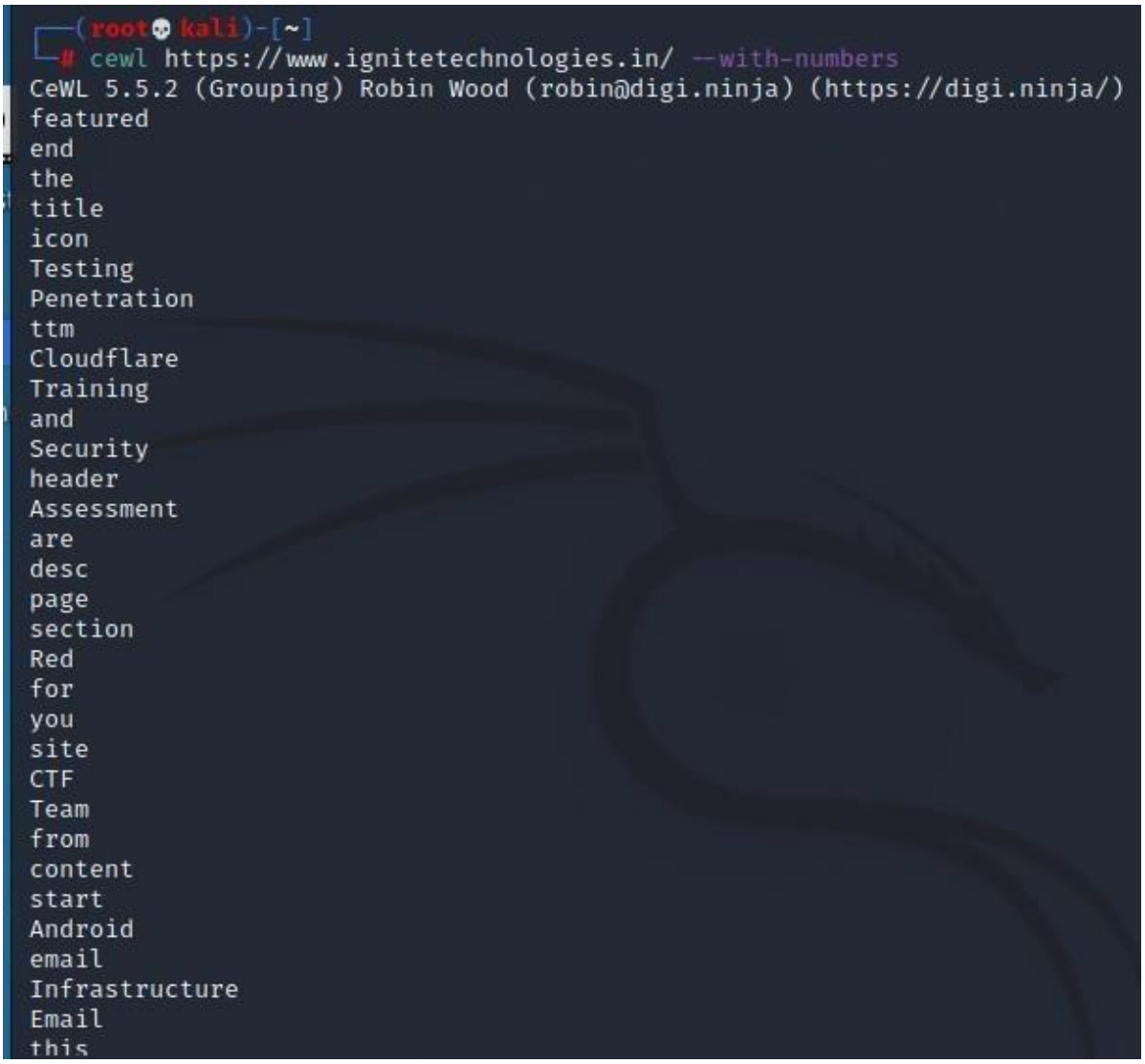
To expand the website crawling result and for retrieving complete detail of a website, you can use **-v option** for verbose mode. Rather than generating awordlist, it will dump the information available on the website.

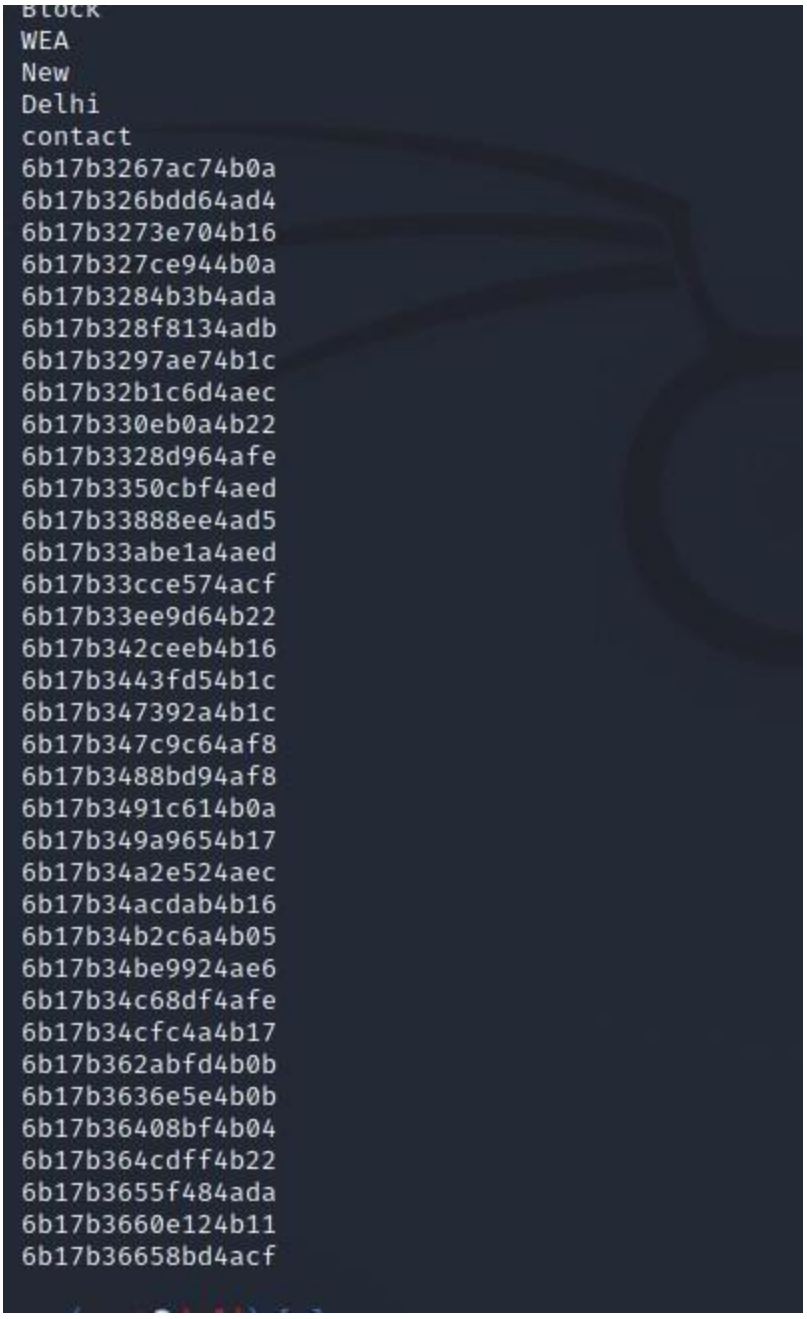



taking
withLinux
andits
Fundamental
Password
Key
Cron
SUID
Groups
Wildcard
SUDO
Writable
NFS
Root
Squashing
Shared
Library
Kernel
Exploits
Variables
Capabilities
linux
kernel
Drop
Matters
Most
far
phone
call
Phone
Address
Pusa
Block
WEA
New
Delhi
contact
fee
abf
adaba
bdbef
ced
ead
afba
(root kali)-[~]
# ss

Command-11: cewl websername –with-numbers

If you want to generate an alpha-numeric wordlist then you can use **–with-numbers** option along with the command.



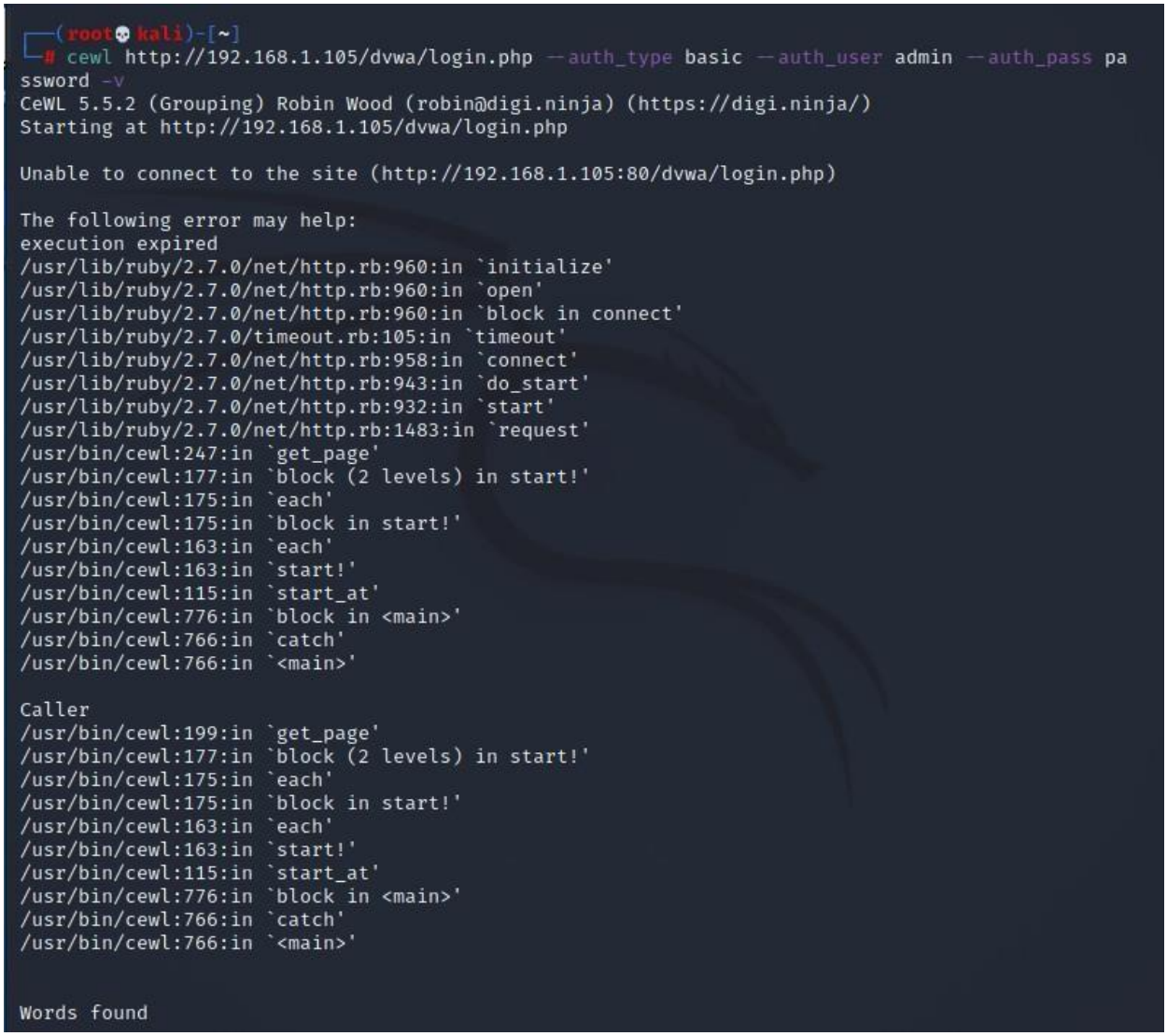

Block
WEA
New
Delhi
contact
6b17b3267ac74b0a
6b17b326bdd64ad4
6b17b3273e704b16
6b17b327ce944b0a
6b17b3284b3b4ada
6b17b328f8134adb
6b17b3297ae74b1c
6b17b32b1c6d4aec
6b17b330eb0a4b22
6b17b3328d964afe
6b17b3350cbf4aed
6b17b33888ee4ad5
6b17b33abe1a4aed
6b17b33cce574acf
6b17b33ee9d64b22
6b17b342ceeb4b16
6b17b3443fd54b1c
6b17b347392a4b1c
6b17b347c9c64af8
6b17b3488bd94af8
6b17b3491c614b0a
6b17b349a9654b17
6b17b34a2e524aec
6b17b34acdab4b16
6b17b34b2c6a4b05
6b17b34be9924ae6
6b17b34c68df4afe
6b17b34cfc4a4b17
6b17b362abfd4b0b
6b17b3636e5e4b0b
6b17b36408bf4b04
6b17b364cdff4b22
6b17b3655f484ada
6b17b3660e124b11
6b17b36658bd4acf

Command-12: If there is page authentication for login into the website then above default willnot work properly, in order to generate a wordlist you need to bypass the authentication page by using the following parameter:

–auth_type: Digest or basic.

–auth_user: Authentication username.

–auth_pass: Authentication password




**TOOL-2 : Hashcat**

```
┌──(kali㉿kali)-[~]
└─$ echo -n password | md5sum |tr -d "-">>hashh

┌──(kali㉿kali)-[~]
└─$ ls
'~'              Desktop                 Downloads    fac.sh      hashes.txt   hey1.txt
 ak1.txt         Dictionary_hashes.txt   exam         f.txt       hashh        hydra.restore
 all-2.0.tar.gz  Documents               example.db   ggg.txt     hello.txt    id_rsa
 des             done.txt                exampl.txt   hash1.txt   hey          'ix posix_encrypt posix_open posix_mkdir posix_

┌──(kali㉿kali)-[~]
└─$ cat hashh
5f4dcc3b5aa765d61d8327deb882cf99

┌──(kali㉿kali)-[~]
└─$ nano rock

┌──(kali㉿kali)-[~]
└─$ cat rock
password
hello
12345
honey
```

```
┌──(kali㉿kali)-[~]
└─$ hashcat -m 0 -a 0 -o output.txt hashh rock
hashcat (v6.1.1) starting ...

OpenCL API (OpenCL 2.0 pocl 1.8  Linux, None+Asserts, RELOC, LLVM 9.0.1, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
==========================================================================================================================================
* Device #1: pthread-Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz, 2540/2604 MB (1024 MB allocatable), 3MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0×0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Applicable optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Hash
* Single-Salt
* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.
Using pure kernels enables cracking longer passwords but for the price of drastically reduced performance.
If you want to switch to optimized backend kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

Host memory required for this attack: 64 MB

Dictionary cache built:
* Filename..: rock
* Passwords.: 4
* Bytes.....: 27
* Keyspace..: 4
* Runtime...: 0 secs

The wordlist or mask that you are using is too small.
This means that hashcat cannot use the full parallel power of your device(s).
Unless you supply more work, your cracking speed will drop.
For tips on supplying more work, see: https://hashcat.net/faq/morework

Approaching final keyspace - workload adjusted.


Session..........: hashcat
```

```
The wordlist or mask that you are using is too small.
This means that hashcat cannot use the full parallel power of your device(s).
Unless you supply more work, your cracking speed will drop.
For tips on supplying more work, see: https://hashcat.net/faq/morework

Approaching final keyspace - workload adjusted.


Session..........: hashcat
Status...........: Cracked
Hash.Name........: MD5
Hash.Target......: 5f4dcc3b5aa765d61d8327deb882cf99
Time.Started.....: Wed Nov 23 17:40:36 2022 (0 secs)
Time.Estimated...: Wed Nov 23 17:40:36 2022 (0 secs)
Guess.Base.......: File (rock)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:    10380 H/s (0.00ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests
Progress.........: 4/4 (100.00%)
Rejected.........: 0/4 (0.00%)
Restore.Point....: 0/4 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: password → honey

Started: Wed Nov 23 17:40:36 2022
Stopped: Wed Nov 23 17:40:38 2022
```

```
┌──(kali㉿kali)-[~]
└─$ cat output.txt
5f4dcc3b5aa765d61d8327deb882cf99:password
```

## TOOL-3: Burp Suite

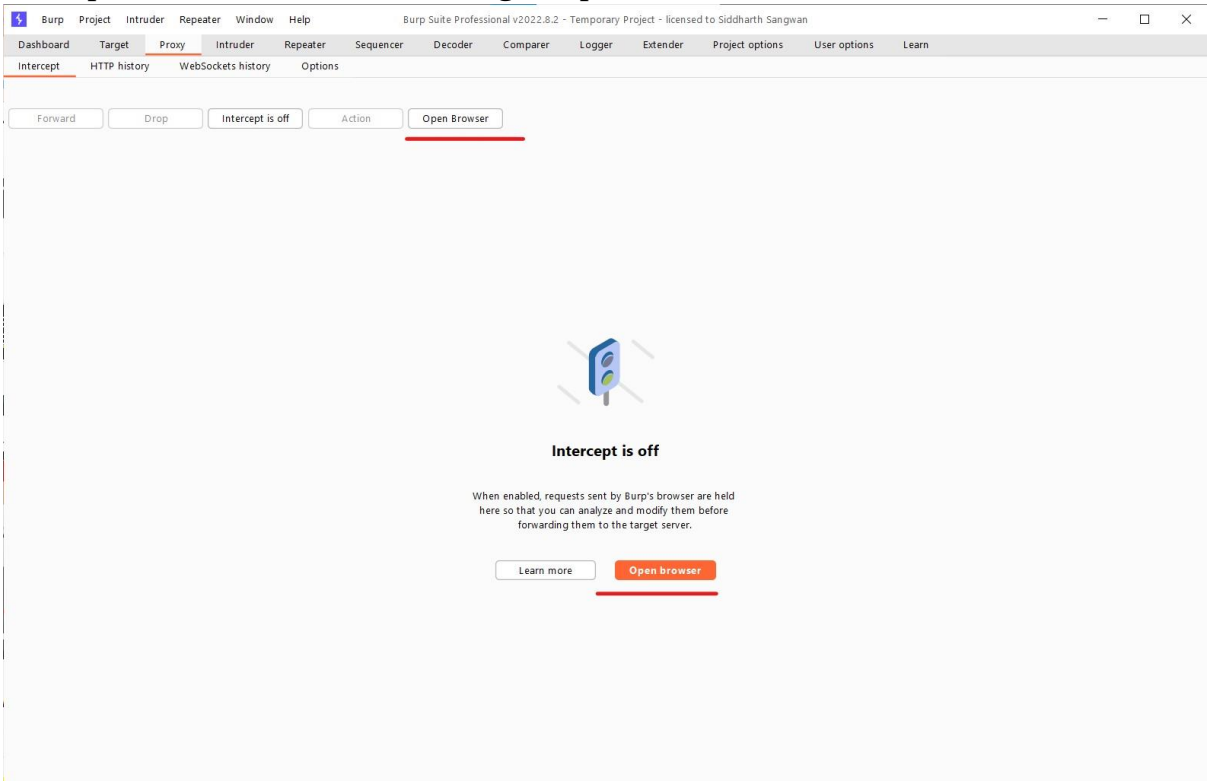
Brute force plays a vital role in web penetration testing because is the simplest method to gain access to a site or server by checking the correct username or password by calculating every possible combination that could generate a username or password

### Requirement:

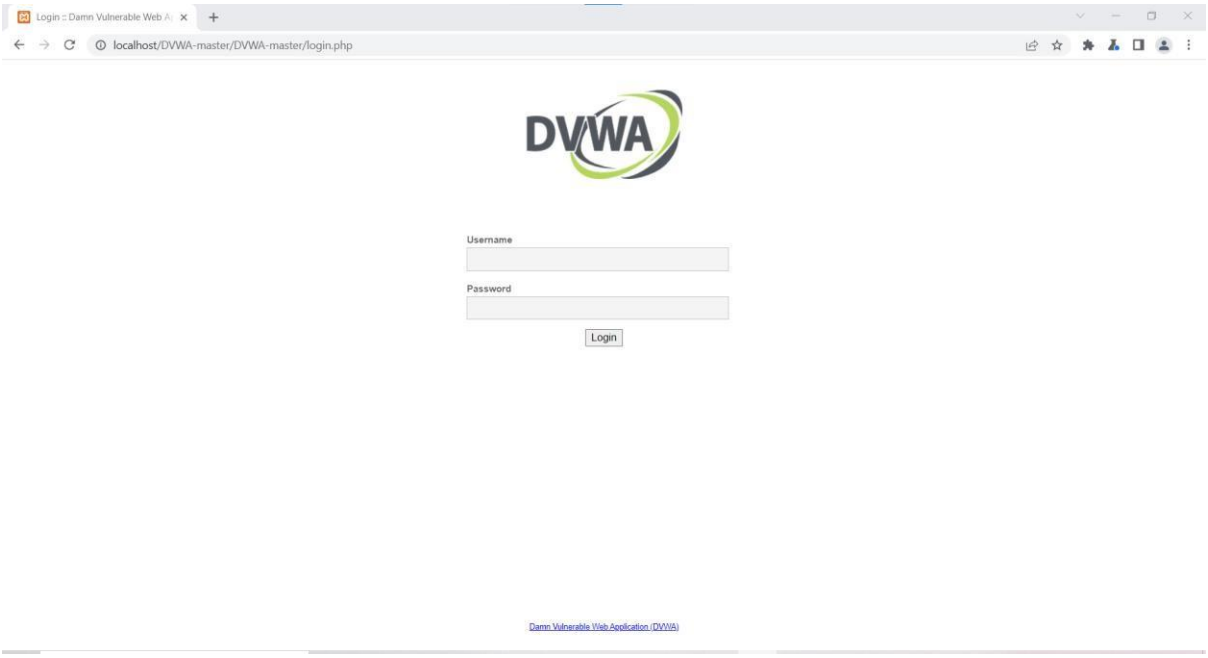
1. XAMPP for Windows 7.4.30, 8.0.23 & 8.1.10
2. Burp Suite Professional
3. DVWA File For testing
   Link: https://github.com/digininja/DVWA.git

### Process:

1. Open a Chrome browser using Burp Suite

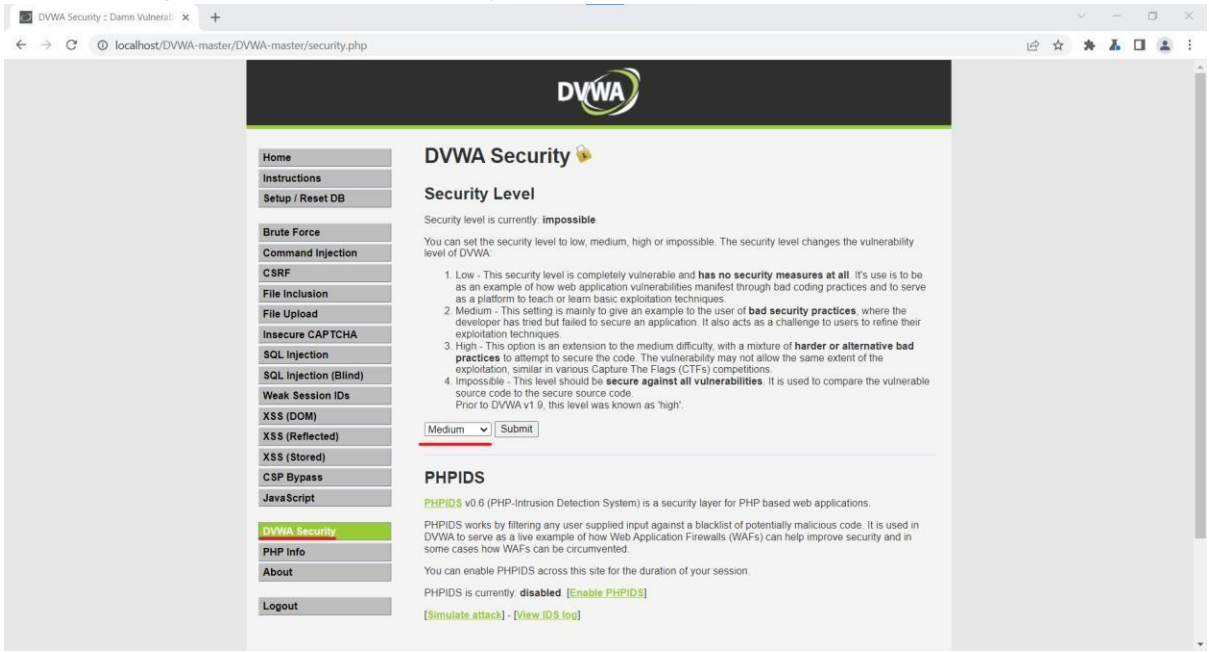



2. On your Xampp server and open localhost in Chrome




**Username:admin**
**Password: password**

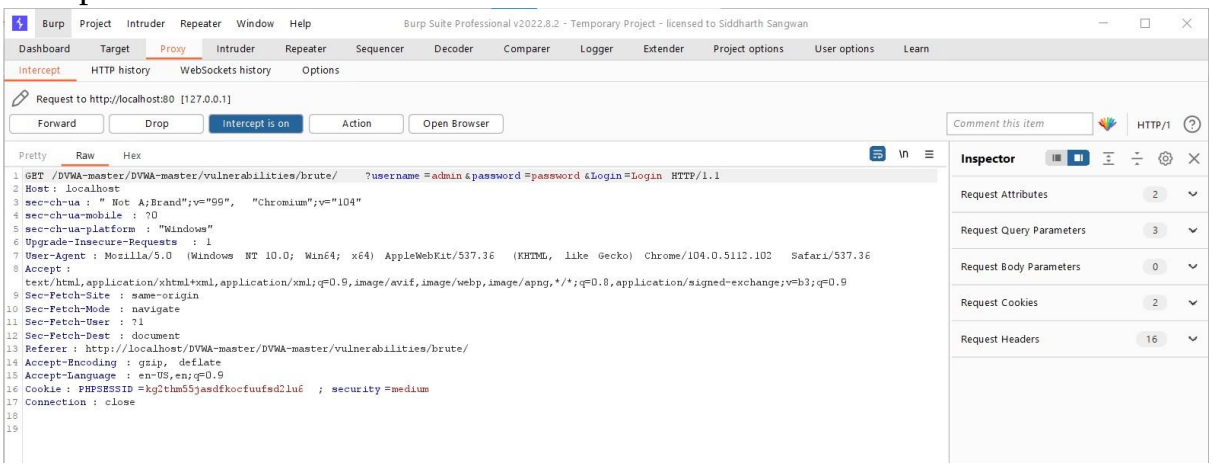
3. Change the DVWA Security level to medium



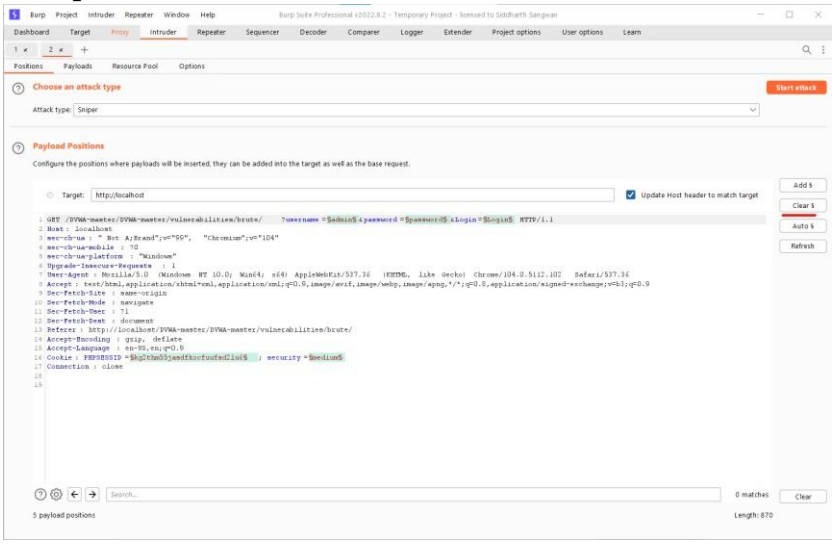

4. Open Brute Force to implement the attack.

5. Keep a dummy username and password to intercept the request in burp suite

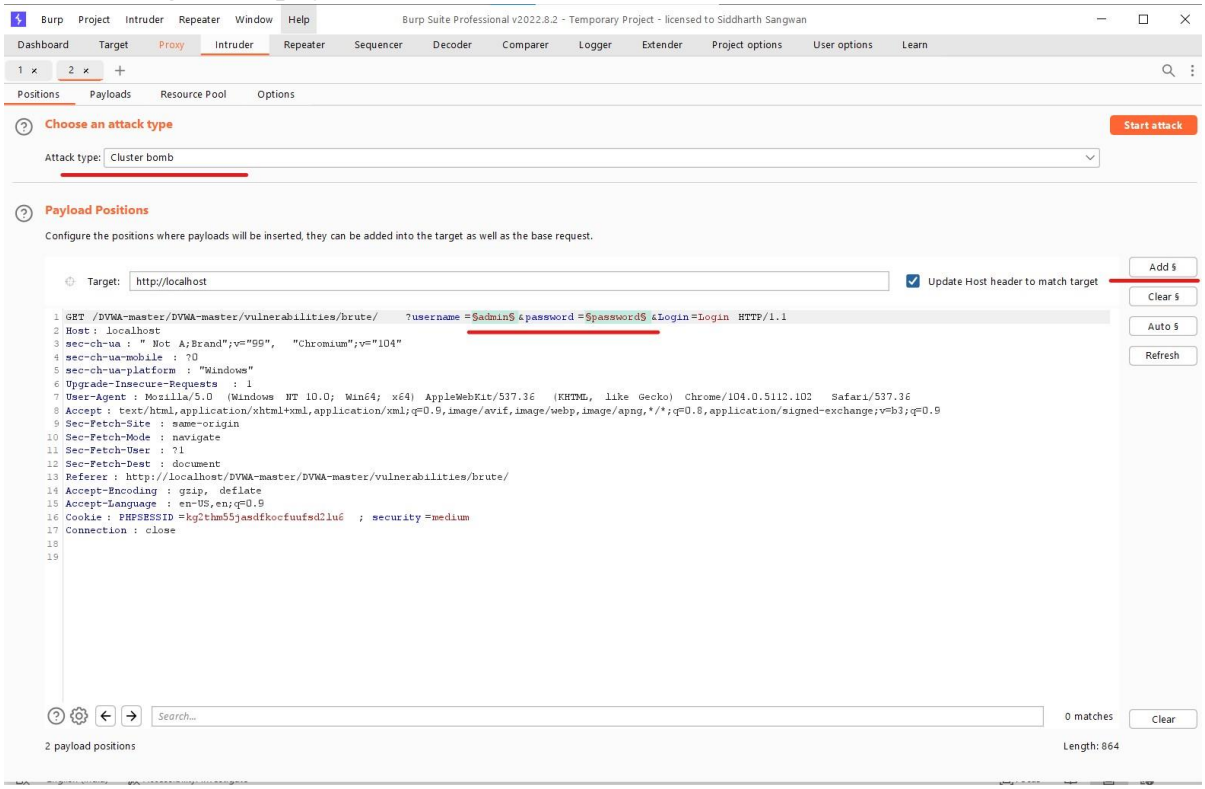

6. Request has been captured in burp suite and right click on the code and press on **send to Intruder**



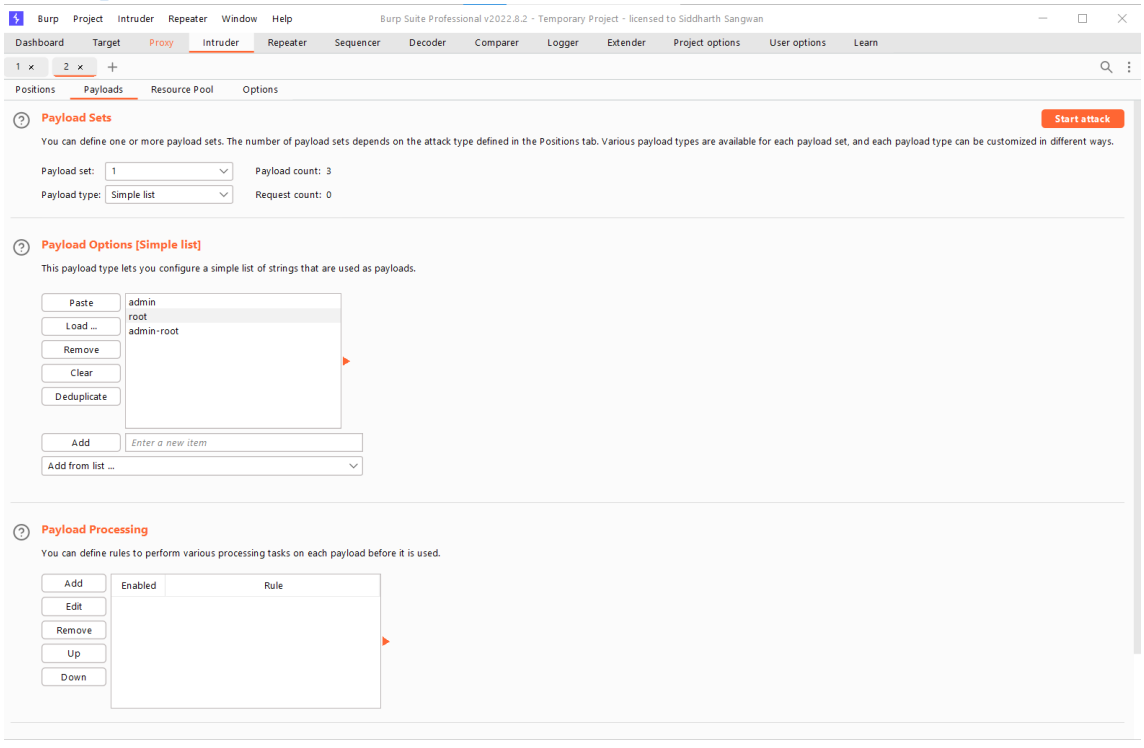

7. Clear all the options

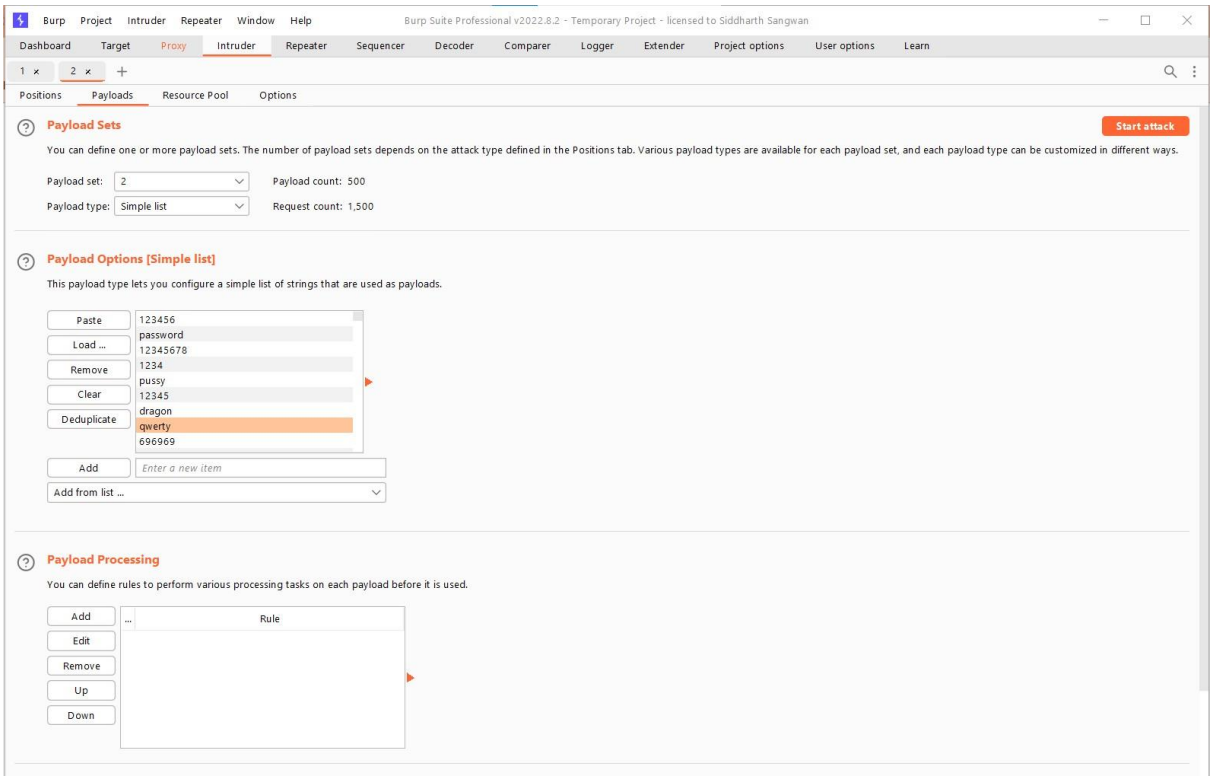


Add username and password to scan and change the attack type to cluster bomb and goto to payloads

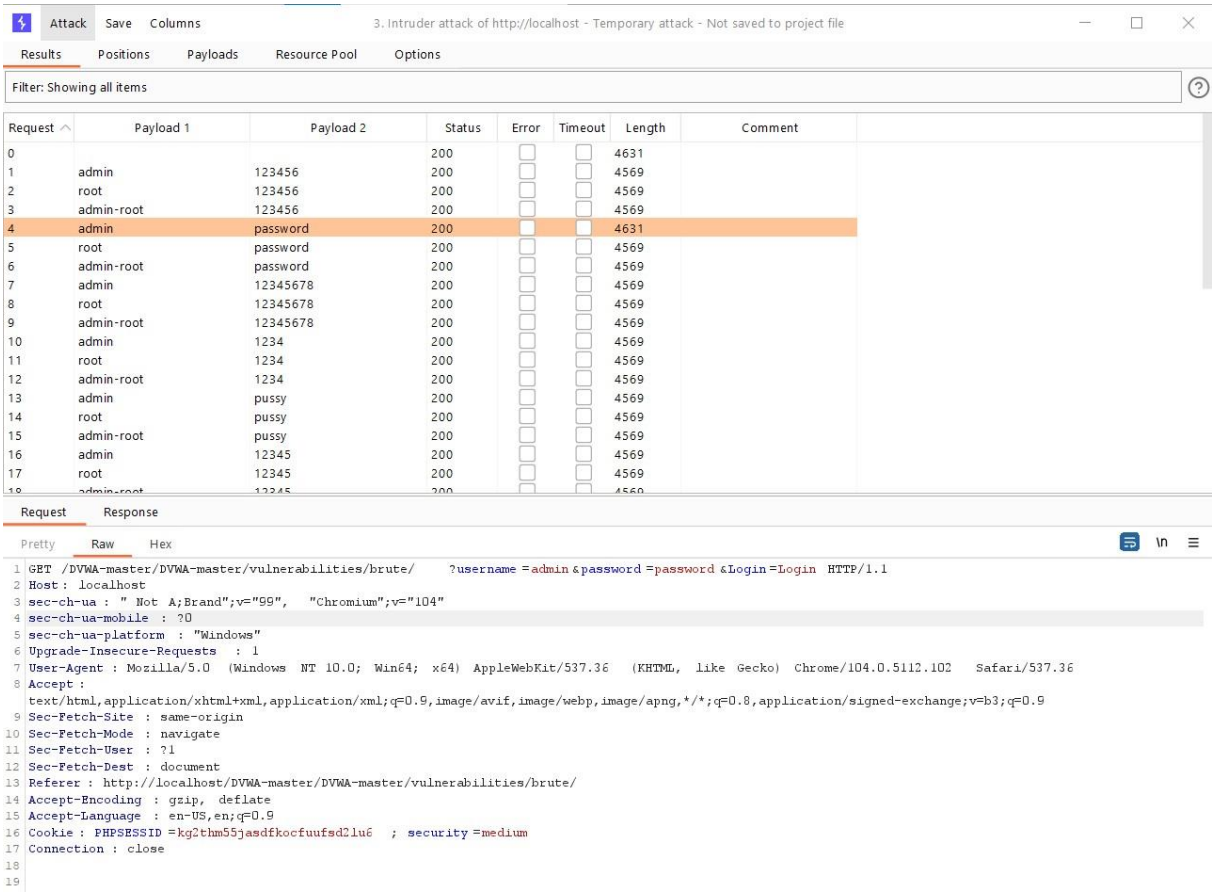



Add the username and password list to check implement the attack and for implementation I used the data from
Link:
https://github.com/danielmiessler/SecLists/blob/master/Passwords/500-worst-passwords.txt

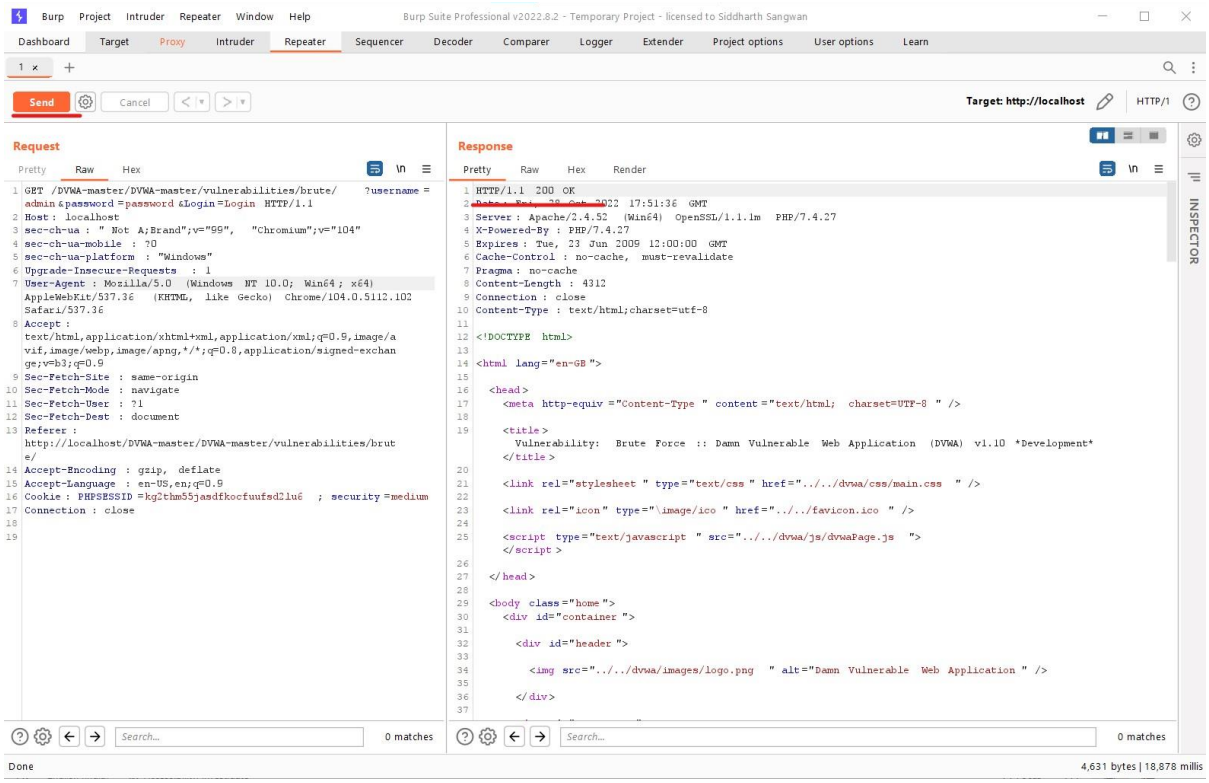




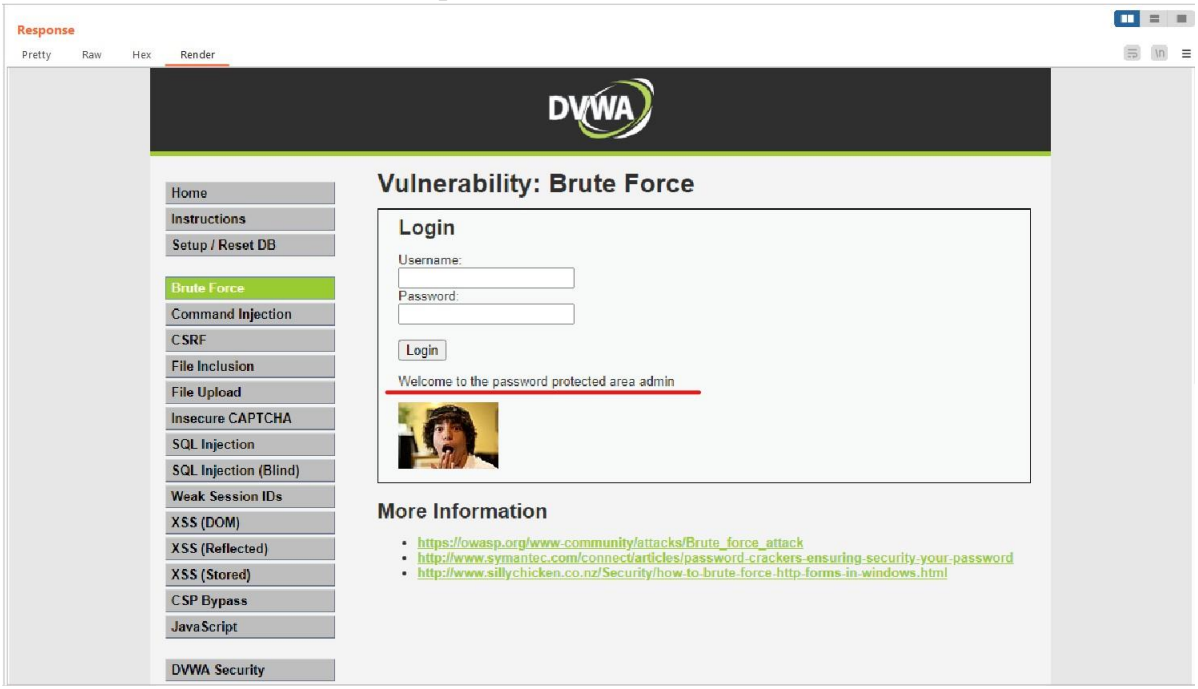

Start the attack It takes some time to check:




Right click on the code and send it to repeater and press the send!




Press on render to see the output:



**Results and Discussions:**
The results for this following report are by using the above tools we got to know how to create a proper word list and crack the passwords.

**Conclusion and Future scope:**
This report states that the implementation of tools like hashcat,cewl,burpsuite. Additionally, dictionary words, personal information, a special date, pet's name, etc. shall always be avoided while creating a password for your network. As demonstrated in the report, they are prone to dictionary and brute-force attacks. Various password generators can be used that can be found that generates and saves passwords securely to help people implement strong passwords. Hence, every individual must be conscious of these types of attacks to have a secured wireless communication in the present and future.

A single password requirement to get into an account is called single-factor authentication. This form has been relied on for many years but is now outdated. A newly formed best practice is multi-factor authentication, where two or more of the following are required for account access:

Something you know. This may be a password or PIN number.
Something you have. This may be an HID card or a server-generated, one-time code given to a user (most of the time on their cellphone), that must be keyed into the device being accessed.
Something you are. This consists of fingerprints, facial recognition, eye scans, and other biometrics.
It adds a second layer of complexity to log-in but provides another barrier of entry against ransomware and data thieves. This encourages them to move on to other, easier targets. While it's not foolproof, it deters attackers to look for another option, potentially saving you from a disaster.

**References:**
**https://www.researchgate.net/publication/337940573_A_Technical_Report_on_Password_Attacks_in_IEEE_80211WPA2_with_PSK**
**https://www.researchgate.net/publication/284609462_Password_Attacks_and_Generation_Strategies**