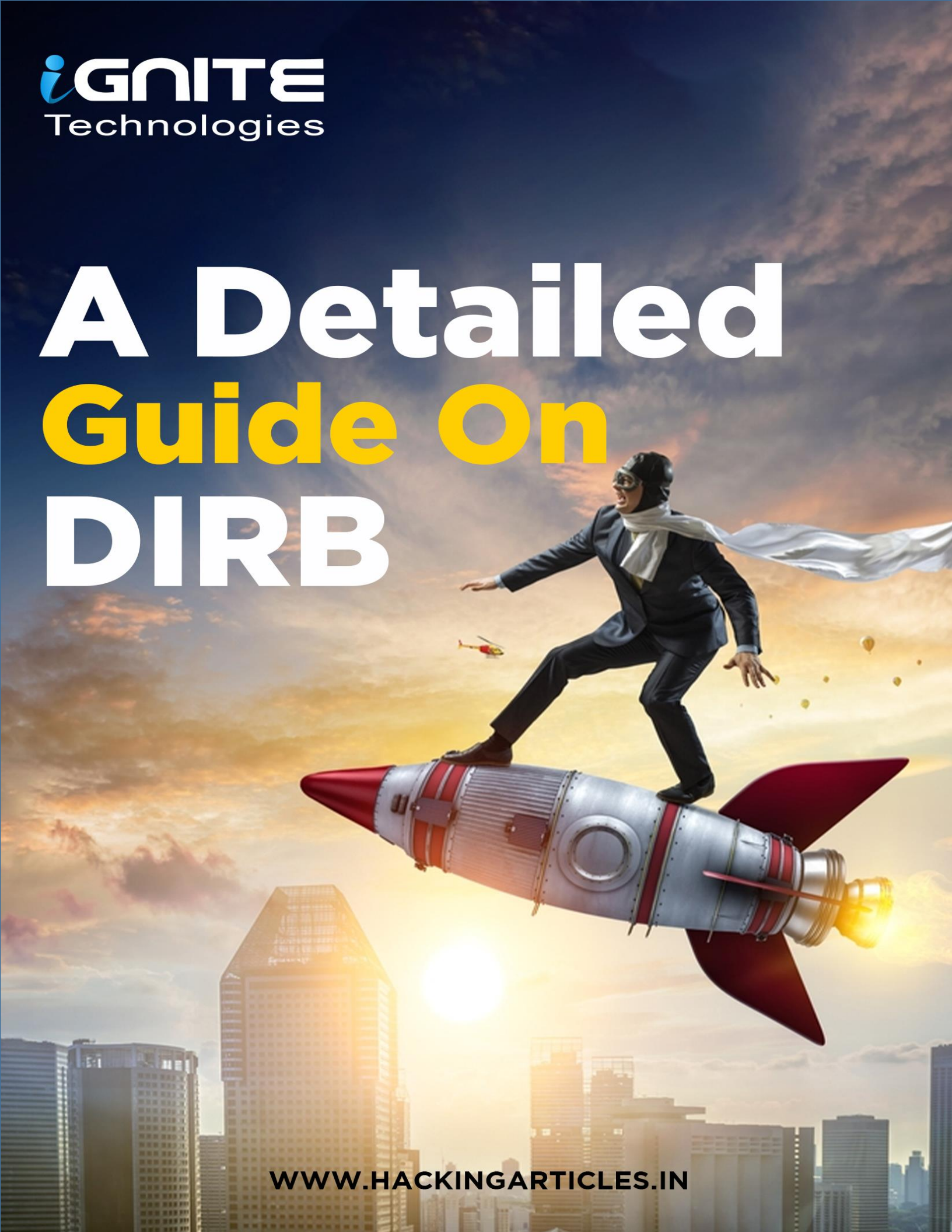
iGNITE Technologies

A Detailed Guide On DIRB



WWW.HACKINGARTICLES.IN

Contents

Introduction

We are focusing on the transient directory using the Kali Linux tool DIRB and trying to find hidden files and directories within a web server.

A path traversal attack, also known as "directory traversal", aims to access files and directories that are stored outside the web root folder. By manipulating variables with reference files with "dot-dot-slash (…/)" sequences and its variations or by using absolute file paths, it may be possible to access arbitrary files and directories stored on file system including application source code, configuration, and critical system files.

Source: https://www.owasp.org/index.php/Path_Traversal

Requirements

Target	BWAPP Labs, DVWA Labs,
Attacker	Kali Linux

What is DIRB?

DIRB is a Web Content Scanner. It looks for existing (and/or hidden) web objects. It basically works by launching a dictionary-based attack against a web server and analysing the response.

It comes with a set of preconfigured attack wordlists for easy usage, but you can use your custom wordlists. Also, DIRB can sometimes be used as a classic CGI scanner, but remember that it is a content scanner, not a vulnerability scanner.

The main purpose is to help in professional web application auditing. especially in security-related testing. It covers some holes not covered by classic web vulnerability scanners. DIRB looks for specific web objects that other generic CGI scanners can't look for. It doesn't search for vulnerabilities, nor does it look for web content that could be vulnerable.

Source: https://tools.kali.org/web-applications/dirb

The DIRB Tool is built-in to Kali Linux. Open the terminal and type the following command to get an overview of the tools included in the package:

Dirb

- -a <agent_string> : Specify your custom USER_AGENT.
- -c <cookie_string> : Set a cookie for the HTTP request.
- -f : Fine tunning of NOT_FOUND (404) detection.
- -H <header_string> : Add a custom header to the HTTP request.
- -i : Use case-insensitive search.
- -l : Print "Location" header when found.
- -N <nf_code>: Ignore responses with this HTTP code.
- -o <output_file> : Save output to disk.

- -p <proxy[:port]> : Use this proxy. (Default port is 1080)
- -P <proxy_username:proxy_password> : Proxy Authentication.
- -r : Don’t search recursively.
- -R : Interactive recursion. (Asks for each directory)
- -S : Silent Mode. Don’t show tested words. (For dumb terminals)
- -t : Don’t force an ending ‘/’ on URLs.
- -u <username:password> : HTTP Authentication.
- -v : Show also NOT_FOUND pages.
- -w : Don’t stop on WARNING messages.
- -X <extensions> / -x <exts_file> : Append each word with this extensions.
- -z : Add a milliseconds delay to not cause excessive Flood.

dirb

```
root@kali:~# dirb

-----------------
DIRB v2.22
By The Dark Raver
-----------------

dirb <url_base> [<wordlist_file(s)>] [options]

========================= NOTES =========================
 <url_base> : Base URL to scan. (Use -resume for session resuming)
 <wordlist_file(s)> : List of wordfiles. (wordfile1,wordfile2,wordfile3...)

======================== HOTKEYS ========================
 'n' -> Go to next directory.
 'q' -> Stop scan. (Saving state for resume)
 'r' -> Remaining scan stats.

======================== OPTIONS ========================
 -a <agent_string> : Specify your custom USER_AGENT.
 -b : Use path as is.
 -c <cookie_string> : Set a cookie for the HTTP request.
 -E <certificate> : path to the client certificate.
 -f : Fine tunning of NOT_FOUND (404) detection.
 -H <header_string> : Add a custom header to the HTTP request.
 -i : Use case-insensitive search.
 -l : Print "Location" header when found.
 -N <nf_code>: Ignore responses with this HTTP code.
 -o <output_file> : Save output to disk.
 -p <proxy[:port]> : Use this proxy. (Default port is 1080)
 -P <proxy_username:proxy_password> : Proxy Authentication.
 -r : Don't search recursively.
 -R : Interactive recursion. (Asks for each directory)
 -S : Silent Mode. Don't show tested words. (For dumb terminals)
 -t : Don't force an ending '/' on URLs.
 -u <username:password> : HTTP Authentication.
 -v : Show also NOT_FOUND pages.
 -w : Don't stop on WARNING messages.
 -X <extensions> / -x <exts_file> : Append each word with this extensions.
 -z <millisecs> : Add a milliseconds delay to not cause excessive Flood.

======================== EXAMPLES ========================
 dirb http://url/directory/ (Simple Test)
 dirb http://url/ -X .html (Test files with '.html' extension)
 dirb http://url/ /usr/share/dirb/wordlists/vulns/apache.txt (Test with apac
 dirb https://secure_url/ (Simple Test with SSL)
```

Utilizing Multiple Wordlist for Directory Traversing

The above attack works by using the default wordlist files common.txt, but we can change this word list and could select another wordlist for directory traversal. You must follow the following path to view all available wordlists.

```
cd /usr/share/wordlists/dirb
ls -la
cd /usr/share/wordlists/vulns
ls -la
```

You can see from the image below that there are so many text files as wordlist; we can use them as required.

```
root@kali:/usr/share/wordlists/dirb# ls -la
total 268
drwxr-xr-x 5 root root   4096 Aug 21 06:48 .
drwxr-xr-x 3 root root   4096 Aug 21 06:48 ..
-rw-r--r-- 1 root root 184073 Jan 24  2012 big.txt
-rw-r--r-- 1 root root   1292 Jan 27  2012 catala.txt
-rw-r--r-- 1 root root  35849 Nov 17  2014 common.txt
-rw-r--r-- 1 root root   1492 May 23  2012 euskera.txt
-rw-r--r-- 1 root root    142 Dec 29  2005 extensions_common.txt
-rw-r--r-- 1 root root     75 Mar 16  2012 indexes.txt
-rw-r--r-- 1 root root    244 Dec 29  2005 mutations_common.txt
drwxr-xr-x 2 root root   4096 Aug 21 06:48 others
-rw-r--r-- 1 root root   6561 Mar  4  2014 small.txt
-rw-r--r-- 1 root root   3731 Nov 12  2014 spanish.txt
drwxr-xr-x 2 root root   4096 Aug 21 06:48 stress
drwxr-xr-x 2 root root   4096 Aug 21 06:48 vulns
root@kali:/usr/share/wordlists/dirb# cd vulns/
root@kali:/usr/share/wordlists/dirb/vulns# ls -la
total 500
drwxr-xr-x 2 root root   4096 Aug 21 06:48 .
drwxr-xr-x 5 root root   4096 Aug 21 06:48 ..
-rw-r--r-- 1 root root    230 Jun 29  2004 apache.txt
-rw-r--r-- 1 root root    259 Dec 30  2011 axis.txt
-rw-r--r-- 1 root root 122829 Aug 30  2007 cgis.txt
-rw-r--r-- 1 root root    706 Jun  7  2005 coldfusion.txt
-rw-r--r-- 1 root root   4648 Oct 26  2011 domino.txt
-rw-r--r-- 1 root root 135331 May 29  2013 fatwire_pagenames.txt
-rw-r--r-- 1 root root   1869 May 17  2011 fatwire.txt
-rw-r--r-- 1 root root    523 Apr  8  2010 frontpage.txt
-rw-r--r-- 1 root root   3896 Mar 16  2012 hpsmh.txt
-rw-r--r-- 1 root root  20644 May 13  2009 hyperion.txt
-rw-r--r-- 1 root root    485 May 31  2004 iis.txt
-rw-r--r-- 1 root root    365 May 24  2004 iplanet.txt
-rw-r--r-- 1 root root    395 Oct  9  2013 jboss.txt
-rw-r--r-- 1 root root   2148 Apr 29  2013 jersey.txt
-rw-r--r-- 1 root root    306 Jun  7  2005 jrun.txt
-rw-r--r-- 1 root root    465 Nov  9  2008 netware.txt
-rw-r--r-- 1 root root  29182 Sep 20  2013 oracle.txt
-rw-r--r-- 1 root root   2442 Jun 29  2012 ror.txt
-rw-r--r-- 1 root root  33300 Oct  1  2013 sap.txt
-rw-r--r-- 1 root root  44075 Sep 15  2011 sharepoint.txt
-rw-r--r-- 1 root root    970 Sep  7  2004 sunas.txt
-rw-r--r-- 1 root root    220 Oct 19  2003 tests.txt
-rw-r--r-- 1 root root   2474 Feb  1  2012 tomcat.txt
-rw-r--r-- 1 root root    536 Feb  6  2007 vignette.txt
-rw-r--r-- 1 root root   7117 Aug 27  2013 weblogic.txt
-rw-r--r-- 1 root root  12564 Jun 27  2013 websphere.txt
root@kali:/usr/share/wordlists/dirb/vulns#
```

Default working of Dirb

In this attack the common.txt is set as a default word list for directory traversal, the protester can use the following command. Open the terminal and type the following command to start the Brussels Directory attack.

```
dirb http://192.168.1.106/dvwa/
```

Using the common.txt file, the DIRB returns the enumerated directories found within the target URL as shown in the below image.

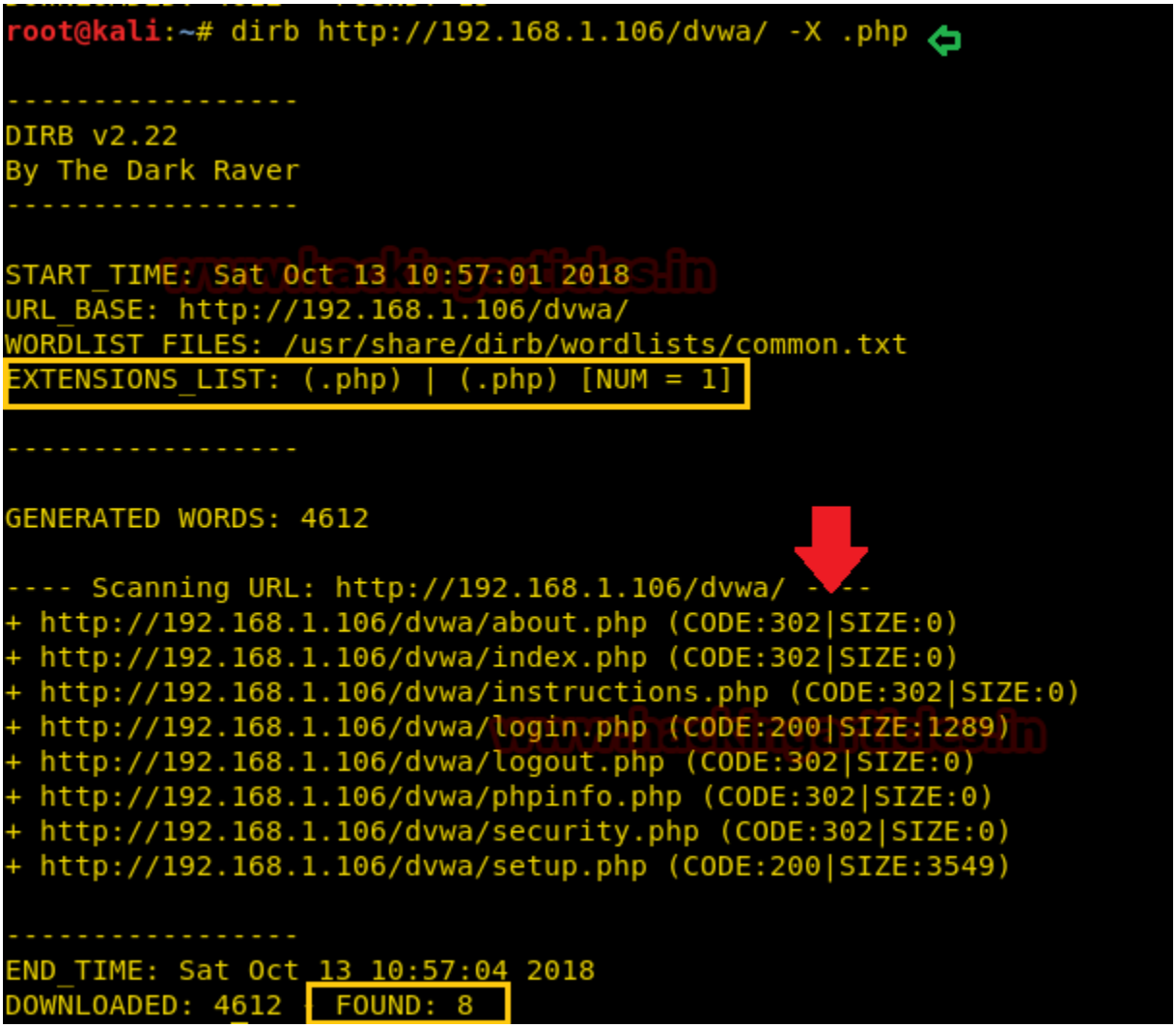
```
root@kali:~# dirb http://192.168.1.106/dvwa/

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sat Oct 13 10:55:25 2018
URL_BASE: http://192.168.1.106/dvwa/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.1.106/dvwa/ ----
+ http://192.168.1.106/dvwa/about (CODE:302|SIZE:0)
==> DIRECTORY: http://192.168.1.106/dvwa/config/
==> DIRECTORY: http://192.168.1.106/dvwa/docs/
==> DIRECTORY: http://192.168.1.106/dvwa/external/
+ http://192.168.1.106/dvwa/favicon.ico (CODE:200|SIZE:1406)
+ http://192.168.1.106/dvwa/index (CODE:302|SIZE:0)
+ http://192.168.1.106/dvwa/index.php (CODE:302|SIZE:0)
+ http://192.168.1.106/dvwa/instructions (CODE:302|SIZE:0)
+ http://192.168.1.106/dvwa/login (CODE:200|SIZE:1289)
+ http://192.168.1.106/dvwa/logout (CODE:302|SIZE:0)
+ http://192.168.1.106/dvwa/php.ini (CODE:200|SIZE:148)
+ http://192.168.1.106/dvwa/phpinfo (CODE:302|SIZE:0)
+ http://192.168.1.106/dvwa/phpinfo.php (CODE:302|SIZE:0)
+ http://192.168.1.106/dvwa/README (CODE:200|SIZE:4934)
+ http://192.168.1.106/dvwa/robots (CODE:200|SIZE:26)
+ http://192.168.1.106/dvwa/robots.txt (CODE:200|SIZE:26)
+ http://192.168.1.106/dvwa/security (CODE:302|SIZE:0)
+ http://192.168.1.106/dvwa/setup (CODE:200|SIZE:3549)

---- Entering directory: http://192.168.1.106/dvwa/config/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.106/dvwa/docs/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.106/dvwa/external/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

-----------------
END_TIME: Sat Oct 13 10:55:28 2018
DOWNLOADED: 4612 - FOUND: 15
```

Enumerating Directory with Specific Extension List

There are a lot of situations where we need to extract the directories of a specific extension from the target server, and then we can use the **-X** parameter of the dirb scan. This parameter accepts the file extension name and then searches for the given extension files on the target server or machine.

```
dirb http://192.168.1.106/dvwa/ -X .php
```

The above command will extract all directory path related to .php extension as shown the following image.



Save Output to Disk

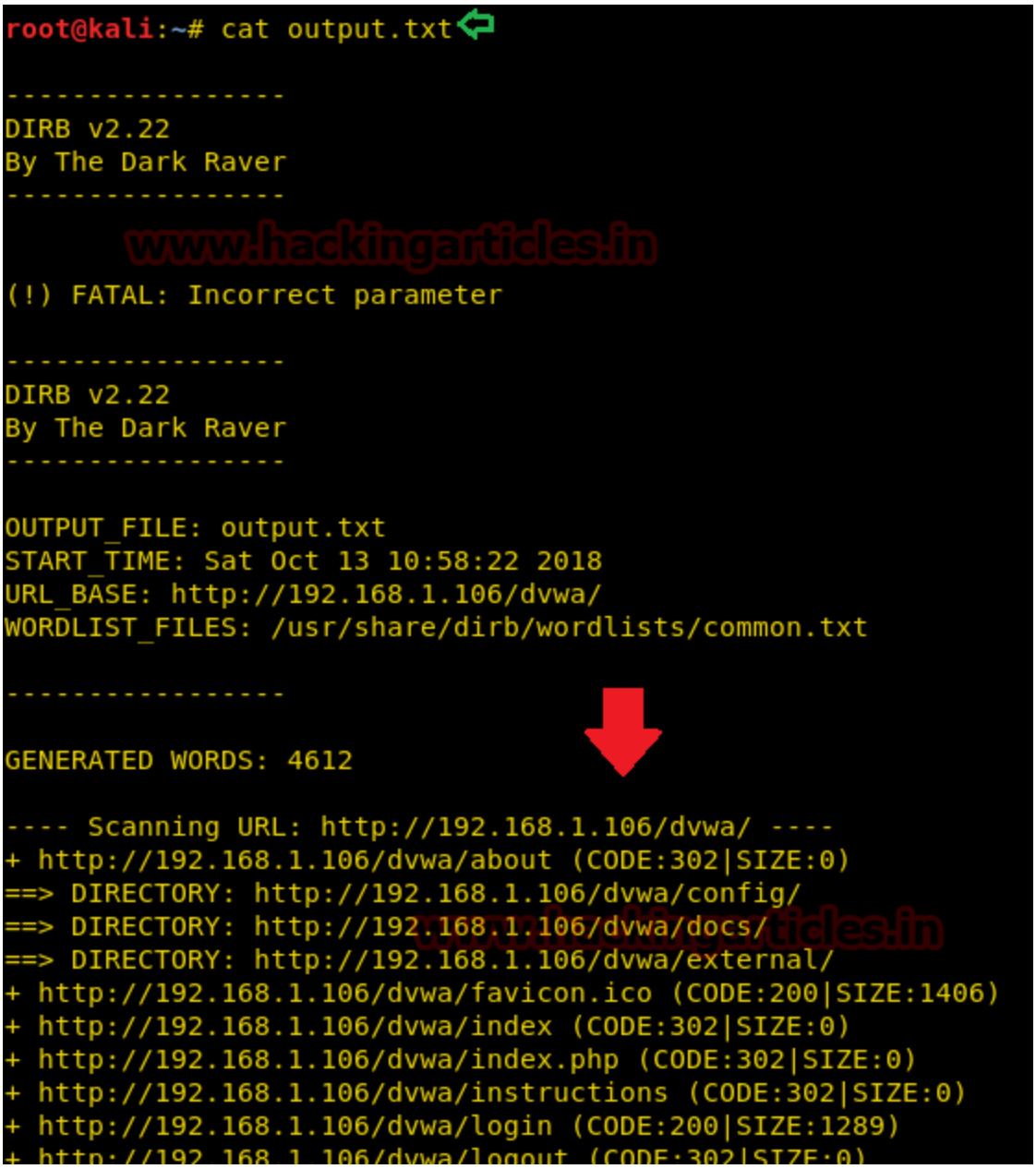
For the purpose of record maintenance, better readability, and future references, we save the output of the dirb scan on a file. To do this, we will use the parameter -o of the dirb scan. We can save the output of the dirb scan in a text file.

```
dirb http://192.168.1.106/dvwa/  -o output.txt
```

The above command will generate an output.txt file at the desktop of the enumerated directories.

Now that we have successfully executed the command, now let's traverse to the location to ensure whether the output has been saved on the file on not. In this case, our location for output is /root/output.txt

```
cat output.txt
```

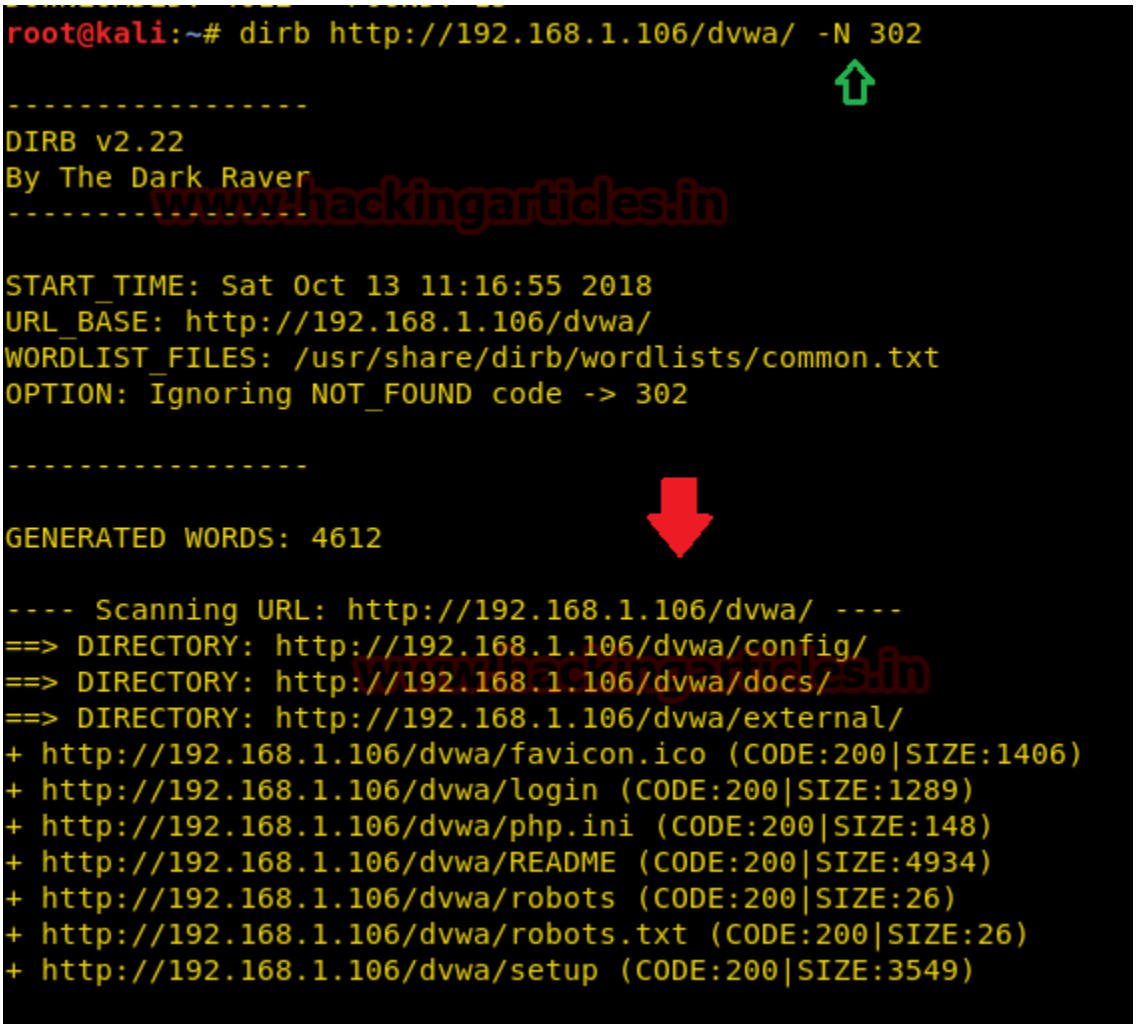


Ignore Unnecessary Status-Code

The Status-Code element is a 3-digit integer where the first digit of the Status-Code defines the class of response and the last two digits do not have any categorization role. In this attack, we are using the **-N** parameter on code 302 as shown below.

```
dirb http://192.168.1.106/dvwa/  -N 302
```

As you can grasp from the given screenshot that the dirb scan is ignoring the NOT FOUND code that is., 302.



Default working vs. Nonstop on WARNING messages working

During the normal dirb scan as shown below, some of the pages generate warnings; the dirb scan skips those directories where it encounters any warnings.

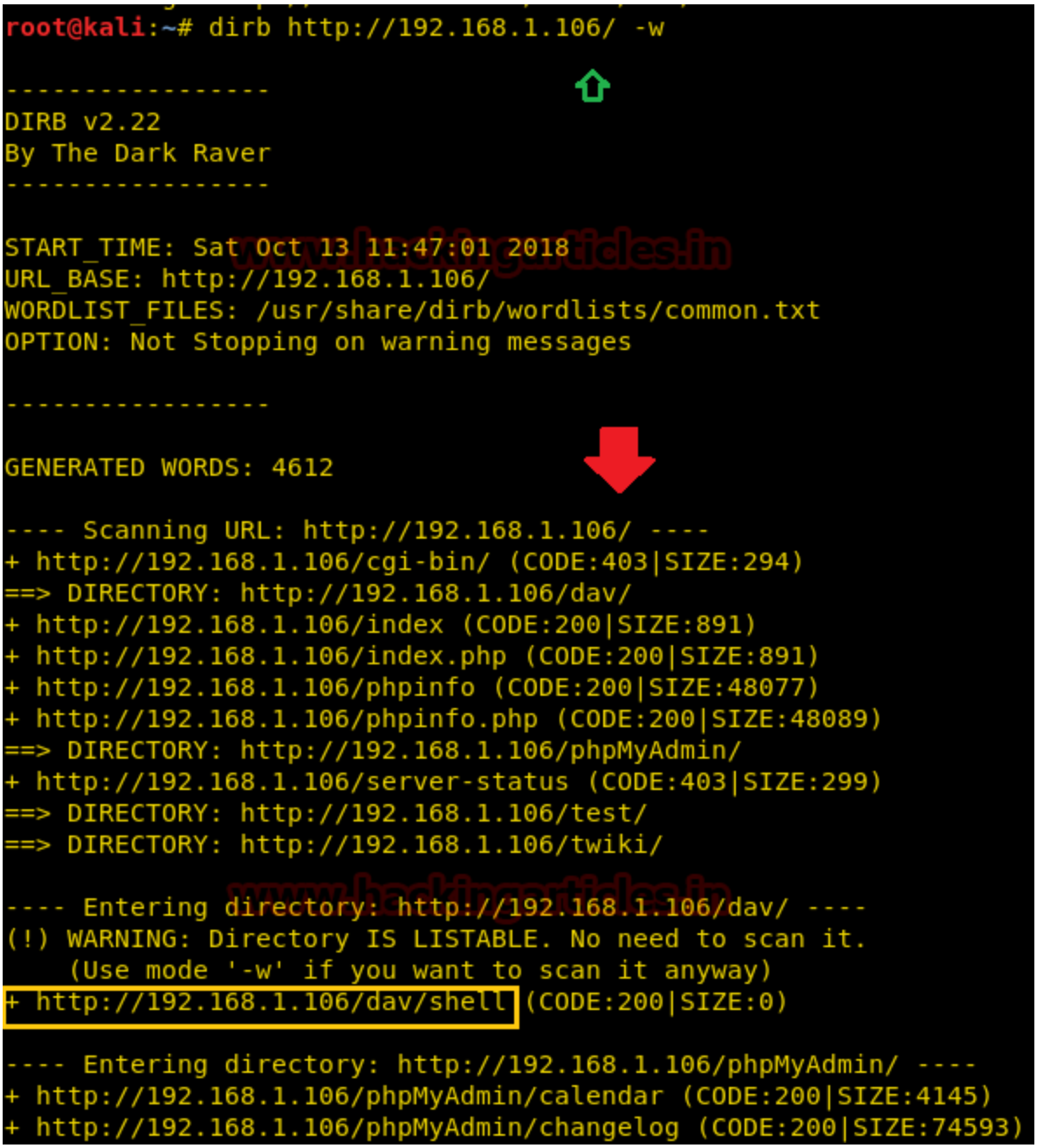
dirb http://192.168.1.106/

```
root@kali:~# dirb http://192.168.1.106/

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sat Oct 13 11:46:45 2018
URL_BASE: http://192.168.1.106/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.1.106/ ----
+ http://192.168.1.106/cgi-bin/ (CODE:403|SIZE:294)
==> DIRECTORY: http://192.168.1.106/dav/
+ http://192.168.1.106/index (CODE:200|SIZE:891)
+ http://192.168.1.106/index.php (CODE:200|SIZE:891)
+ http://192.168.1.106/phpinfo (CODE:200|SIZE:48077)
+ http://192.168.1.106/phpinfo.php (CODE:200|SIZE:48089)
==> DIRECTORY: http://192.168.1.106/phpMyAdmin/
+ http://192.168.1.106/server-status (CODE:403|SIZE:299)
==> DIRECTORY: http://192.168.1.106/test/
==> DIRECTORY: http://192.168.1.106/twiki/

---- Entering directory: http://192.168.1.106/dav/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
```

While doing a scan that is to be done very deeply and verbosely, we want that the dirb scan to not avoid these warnings and do an in-depth scan, hence we use the **-w** parameter of the dirb scan.

dirb http://192.168.1.106/ -w

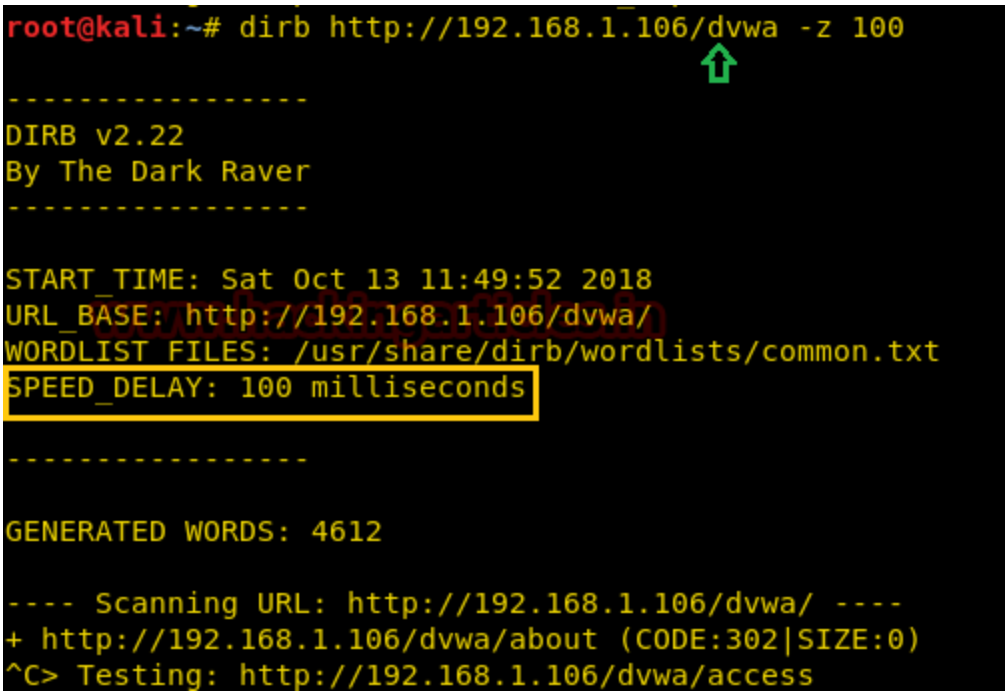
As you can observe the highlighted directory **/dev/shell** is enumerated even after the warning message which is missing in the default scan.



Speed delay

While working in different scenarios, there is some environment we come across that cannot handle the flood created by the dirb scan, so in those environments, it is important that we delay the scan for some time. This can be done easily with the **-z** parameter of the dirb scan. In this parameter, the time is provided on the scale of milliseconds. Like as shown in our given example, we have given 100 seconds delay to dirb.

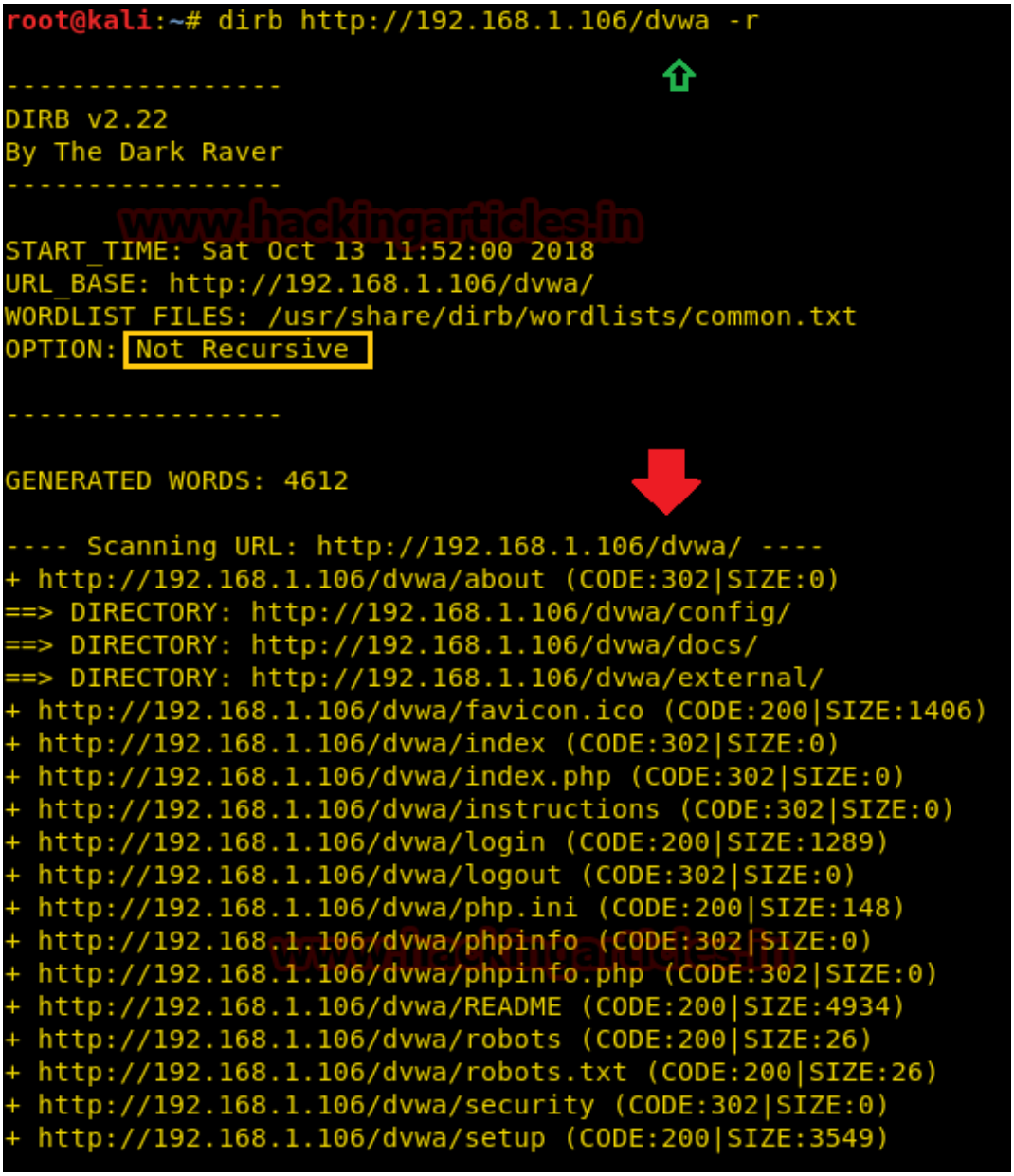
```
dirb http://192.168.1.106/dvwa  -z 100
```



Not recursively (-r)

The dirb scan, by default, scans the directories recursively. It means it scans a directory and then traverses inside that directory to scan for more subdirectories. But in some scenarios, where time is insufficient, we set the dirb to not scan recursively. This can be achieved using the **-r** parameter.

dirb http://192.168.1.106/dvwa -r

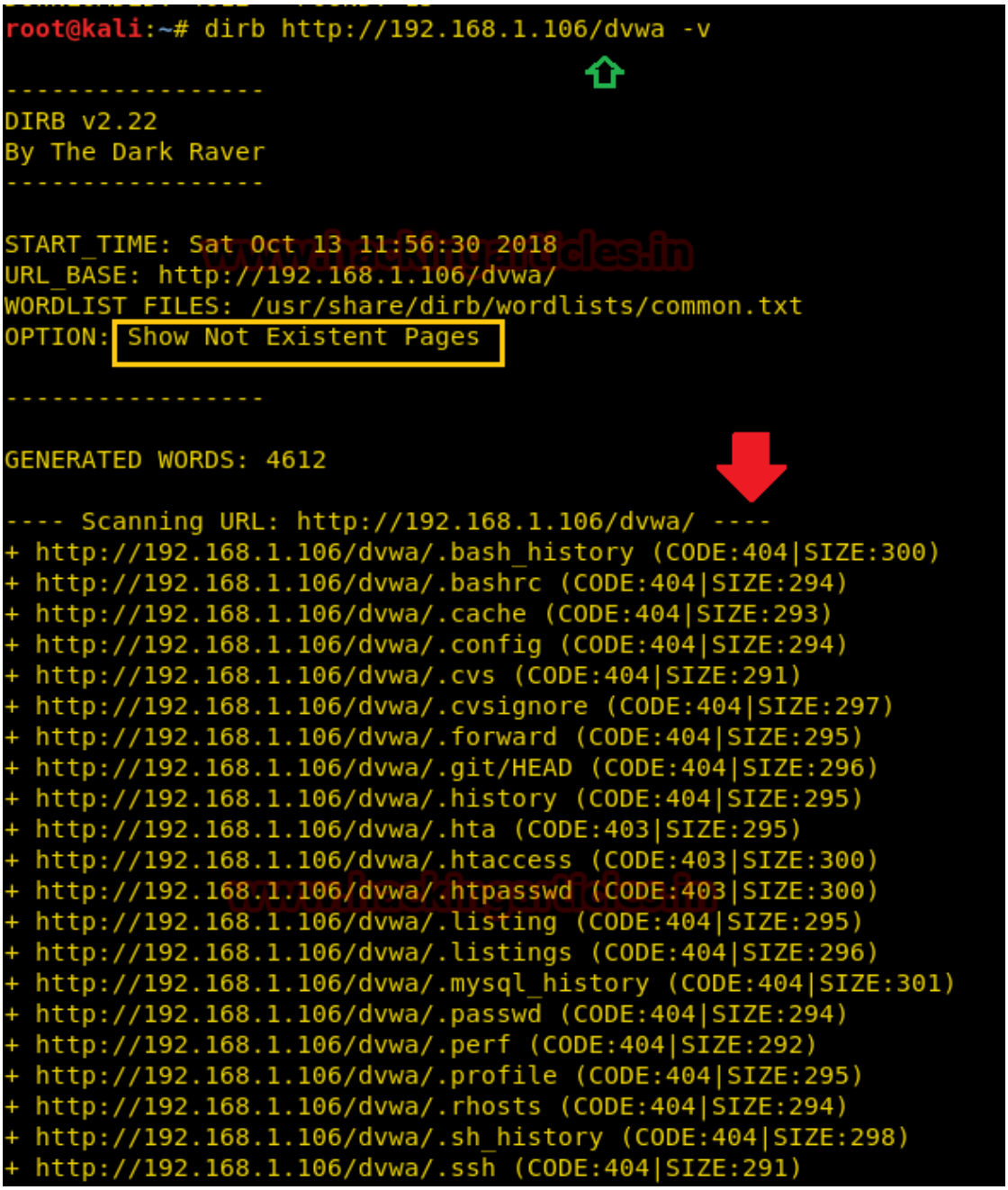


Show NOT Existence Pages

A 404 error is an HTTP status code that means that the page you were trying to reach on a website couldn't be found on their server. 404 Not Found error messages are frequently customized by individual websites. In some scenarios we need to find the 404 pages too, which dirb skips by default? To find those pages we will use **-v** parameter.

dirb http://192.168.1.106/dvwa -v

From given below the image you can observe it has also extracted all those directories are relevant to 404 errors.



Extension List (-X parameter) vs. Extension Header (-H parameter)

By using the **–X** parameter along with target URL with a specific extension, for example, .php, it enumerates all file or directory with .php extension, but by using **–H** parameter with specific extension, for example .php along with target URL it will enumerate all files or directories named with php as shown in the given below image.

```
dirb http://192.168.1.106/dvwa -X .php
dirb http://192.168.1.106/dvwa -H .php
```

```
EXTENSIONS_LIST: (.php) | (.php) [NUM = 1]

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.1.106/dvwa/ ----
+ http://192.168.1.106/dvwa/about.php (CODE:302|SIZE:0)
+ http://192.168.1.106/dvwa/index.php (CODE:302|SIZE:0)
+ http://192.168.1.106/dvwa/instructions.php (CODE:302|SIZE:0)
+ http://192.168.1.106/dvwa/login.php (CODE:200|SIZE:1289)
+ http://192.168.1.106/dvwa/logout.php (CODE:302|SIZE:0)
+ http://192.168.1.106/dvwa/phpinfo.php (CODE:302|SIZE:0)
+ http://192.168.1.106/dvwa/security.php (CODE:302|SIZE:0)
+ http://192.168.1.106/dvwa/setup.php (CODE:200|SIZE:3549)

-----------------
END_TIME: Sat Oct 13 12:00:20 2018
DOWNLOADED: 4612 - FOUND: 8
root@kali:~# dirb http://192.168.1.106/dvwa -H .php

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sat Oct 13 12:01:32 2018
URL_BASE: http://192.168.1.106/dvwa/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
ADDED_HEADERS:
--
.php
--

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.1.106/dvwa/ ----
+ http://192.168.1.106/dvwa/about (CODE:302|SIZE:0)
==> DIRECTORY: http://192.168.1.106/dvwa/config/
==> DIRECTORY: http://192.168.1.106/dvwa/docs/
==> DIRECTORY: http://192.168.1.106/dvwa/external/
+ http://192.168.1.106/dvwa/favicon.ico (CODE:200|SIZE:1406)
+ http://192.168.1.106/dvwa/index (CODE:302|SIZE:0)
+ http://192.168.1.106/dvwa/index.php (CODE:302|SIZE:0)
+ http://192.168.1.106/dvwa/instructions (CODE:302|SIZE:0)
+ http://192.168.1.106/dvwa/login (CODE:200|SIZE:1289)
+ http://192.168.1.106/dvwa/logout (CODE:302|SIZE:0)
+ http://192.168.1.106/dvwa/php.ini (CODE:200|SIZE:148)
+ http://192.168.1.106/dvwa/phpinfo (CODE:302|SIZE:0)
+ http://192.168.1.106/dvwa/phpinfo.php (CODE:302|SIZE:0)
```

Not forcing an ending '/' on URLs (-t)

From the attacks used in the previous situations, in order to run the dirb tool, we will have to add a forward slash (/) at the end of the URL to be accepted in dirb. In order to check that we need to try one attack on URL ending without any forward slash.

```
dirb http://192.168.1.105/bWAPP/portal.php
```

You will observe that the scan doesn't get executed successfully because of the lack of the forward slash, the importance of which we discussed earlier in this article.

Try this attack once again with the same command with some changes so in order to run that command we have to add **–t** in the previous command.

```
dirb http://192.168.1.105/bWAPP/portal.php -t
```

As now we can observe that the even in the absence of the forward slash, we still have successfully executed the dirb scan.

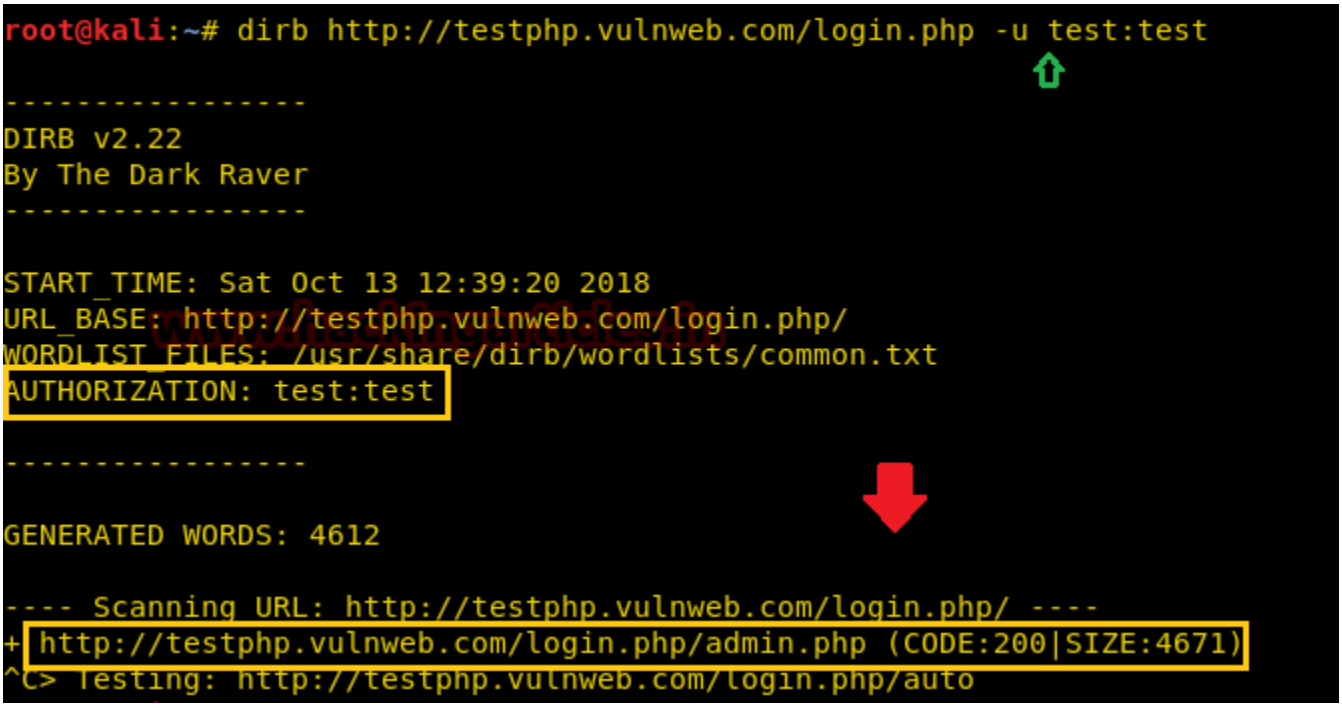
```
root@kali:~# dirb http://192.168.1.105/bWAPP/portal.php

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sat Oct 13 12:34:11 2018
URL_BASE: http://192.168.1.105/bWAPP/portal.php/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.1.105/bWAPP/portal.php/ ----

-----------------
END_TIME: Sat Oct 13 12:34:14 2018
DOWNLOADED: 4612 - FOUND: 0
root@kali:~# dirb http://192.168.1.105/bWAPP/portal.php -t

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sat Oct 13 12:34:22 2018
URL_BASE: http://192.168.1.105/bWAPP/portal.php
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
OPTION: NOT forcing an ending '/' on URLs

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.1.105/bWAPP/portal.php ----
+ http://192.168.1.105/bWAPP/portal.phps (CODE:403|SIZE:297)

-----------------
END_TIME: Sat Oct 13 12:34:25 2018
DOWNLOADED: 4612 - FOUND: 1
```

HTTP AUTHORIZATION (-u username: password)

HTTP Authentication/Authentication Mechanisms are all based on the use of the 401-status code and the WWW-Authenticate response header. The most widely used HTTP authentication mechanism is Basic. The client sends the user name and password as unencrypted base64 encoded text.

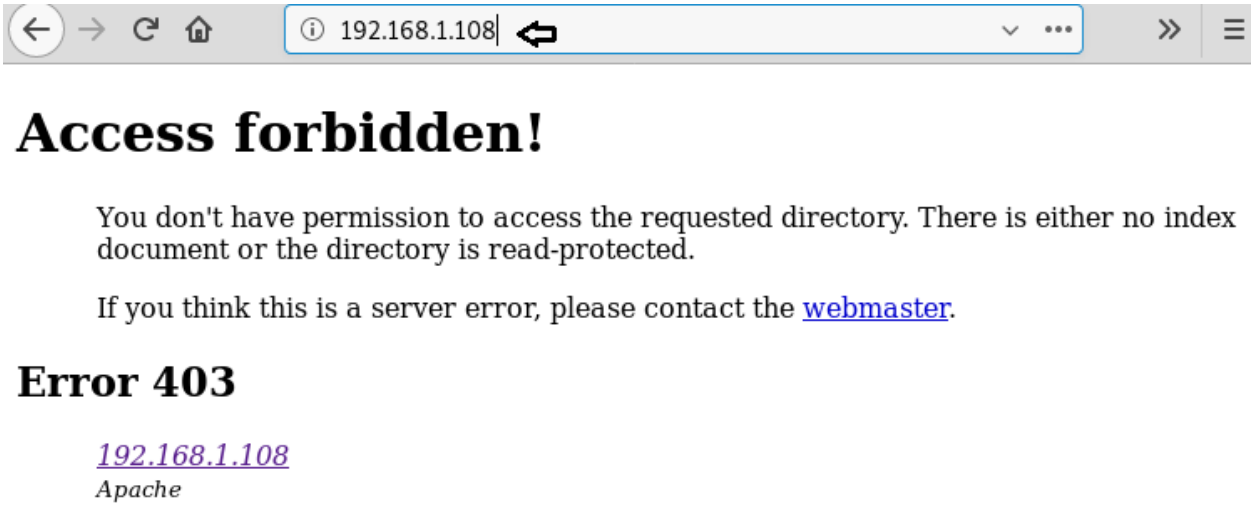
So, in order to bypass this kind of authentication with the help of Dirb, we have used the command below: As a result, it is shown Status –code 200 for the test: test and authorized credential on target URL.

dirb http://testphp.vulnweb.com/login.php -u test:test



Proxy URL

Using **-p option** enables proxy URL to be used for all requests, by default it works on port 1080. As you can observe, on exploring target network IP in the web browser it put up "Access forbidden error" which means this web page is running behind some proxy.



To ensure this prediction, we run the dirb command twice, firstly on port 80 which is by default and further on port 3129 along with **–p option** which enables proxy parameter.

```
dirb http://192.168.1.108
dirb http://192.168.1.108/  -p 192.168.1.108:3129
```

From the given below image, you can take reference for the output result obtained for above commands, here we haven't obtained any directory or file on executing the first command where else in the second command executed successfully.

```
root@kali:~# dirb http://192.168.1.108

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Tue Oct 23 13:06:03 2018
URL_BASE: http://192.168.1.108/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.1.108/ ----
(!) WARNING: All responses for this directory seem to be CODE = 403.
    (Use mode '-w' if you want to scan it anyway)

-----------------
END_TIME: Tue Oct 23 13:06:04 2018
DOWNLOADED: 101 - FOUND: 0
root@kali:~# dirb http://192.168.1.108/ -p 192.168.1.108:3129

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Tue Oct 23 13:06:12 2018
URL_BASE: http://192.168.1.108/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
PROXY: 192.168.1.108:3129

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.1.108/ ----
==> DIRECTORY: http://192.168.1.108/blog/
+ http://192.168.1.108/index.html (CODE:200|SIZE:3181)

---- Entering directory: http://192.168.1.108/blog/ ----
+ http://192.168.1.108/blog/index.php (CODE:301|SIZE:0)
==> DIRECTORY: http://192.168.1.108/blog/wp-admin/
==> DIRECTORY: http://192.168.1.108/blog/wp-content/
==> DIRECTORY: http://192.168.1.108/blog/wp-includes/
+ http://192.168.1.108/blog/xmlrpc.php (CODE:405|SIZE:42)

---- Entering directory: http://192.168.1.108/blog/wp-admin/ ----
```

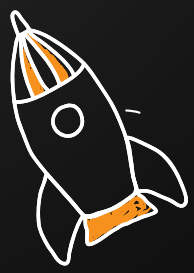
iGNITE Technologies

JOIN OUR TRAINING PROGRAMS

CLICK HERE

BEGINNER

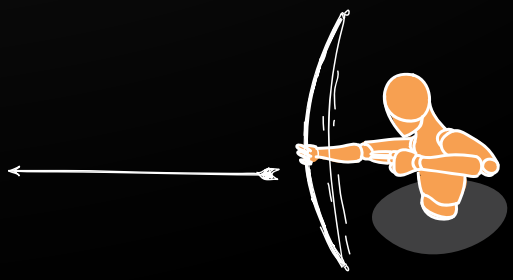
- Ethical Hacking
- Network Pentest
- Bug Bounty
- Wireless Pentest
- Network Security Essentials



ADVANCED

- Burp Suite Pro
- Android Pentest
- Web Services-API
- Advanced Metasploit
- Pro Infrastructure VAPT
- CTF
- Computer Forensics

EXPERT



- Red Team Operation
  - APT's - MITRE Attack Tactics
  - Active Directory Attack
  - MSSQL Security Assessment
- Privilege Escalation
  - Windows
  - Linux