

Module:

DIRB-(A web content scanner)





What is Dirb

DIRB is a command line based tool to brute force any directory based on wordlists. DIRB will make an HTTP request and see the HTTP response code of each request

How it works

It internally has a wordlist file which has by default around 4000 words for brute force attack. There are a lot of updated wordlists available over the internet which can also be used. Dirb searches for the words in its wordlist in every directory or object of a website or a server. It might be an admin panel or a subdirectory that is vulnerable to attack. The key is to find the objects as they are generally hidden.

Table of Content:

- Introduction to DIRB
- Utilizing Multiple Wordlist for Directory Traversing
- Default working of Dirb
- Enumerating Directory with Specific Extension List
- Save Output to Disk
- Ignore Unnecessary Status-Code
- Default Working Vs Not stop on WARNING messages Working
- Speed delay
- > Not recursively (-r)
- Show NOT Existence Pages
- Extension List (-X parameter) Vs Extension Header (-H parameter)
- Not forcing an ending '/' on URLs (-t)
- HTTP Authentication (-u username: password)

How to get it?

Donwload Dirb via Github: <u>https://github.com/seifreed/dirb</u> <i>Download Dirb via Sourceforge: <u>https://sourceforge.net/projects/dirb/</u> *How to get it?*



Donwload Dirb via Github: <u>https://github.com/seifreed/dirbDownload</u> Dirb via Sourceforge: <u>https://sourceforge.net/projects/dirb/</u>

Note : I used Kali Linux and Dirb comes pre-installed with Kali.

Purpose of Dirb in Security testing:

Purpose of DIRB is to help in professional and web application auditing in security testing. DIRB looks for almost all the web objects that other generic CGI scanners can't look for. It doesn't look for vulnerabilities but it looks for the web contents that can be vulnerable.

Using Dirb:

Step 1 — Open Terminal

Step 2 — Start Dirb

Once we have a terminal open, go ahead and type dirb to get the help screen.

Kali> dirb

File Actions Edit View Help
<pre>(root@kali)-[~] # dirb</pre>
DIRB v2.22
By The Dark Raver
<pre>dirb <url_base> [<wordlist_file(s)>] [options]</wordlist_file(s)></url_base></pre>
NOTES
<pre><url_base> : Base URL to scan. (Use -resume for session resuming) <wordlist_file(s)> : List of wordfiles. (wordfile1,wordfile2,wordfile3)</wordlist_file(s)></url_base></pre>
'n' \rightarrow Go to next directory.
'q' → Stop scan. (Saving state for resume)
'r' → Remaining scan stats.
-a <agent string=""> : Specify your custom USER AGENT.</agent>
-b : Use path as is.
-c <cookie_string> : Set a cookie for the HTTP request.</cookie_string>



File Actions Edit View Help

OPTIONS
-a <agent string=""> : Specify your custom USER AGENT.</agent>
-b : Use path as is.
-c cookie strings: Set a cookie for the HTTP request.
-E coertificates : nath to the client certificate
$-f \cdot Eine tunping of NOT FOUND (404) detection$
-H chooder strings - Add a custom header to the HTTD request
i lise case incompletive coston header to the http request.
- 1 : Use case-insensitive search.
-l: Print Location neader when found.
-N <nf_code>: Ignore responses with this HTP code.</nf_code>
-o <output_file> : Save output to disk.</output_file>
<pre>-p <proxy[:port]> : Use this proxy. (Default port is 1080)</proxy[:port]></pre>
-P <proxy_username:proxy_password> : Proxy Authentication.</proxy_username:proxy_password>
-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.</username:password>
-v : Show also NOT FOUND pages.
-w : Don't stop on WARNING messages.
-X <extensions -x="" <exts="" append="" each="" extensions.<="" files:="" td="" this="" with="" word=""></extensions>
-z <millisecs> : Add a milliseconds delay to not cause excessive Flood.</millisecs>

As you can see in this screenshot above, DIRB's syntax is very simple with multiple options. In its simplest form, we only need to type the command dirb followed by the URLof the website we are testing.

Kali> dirb URL

Step 3 — Dirb for simple hidden object scan

with the Dirb's default word list file it searches the URL for 4612 Object types. Let's try it on test site, webscantest.com.

kali > dirb <u>http://testingsite.com</u>

DIRB begins the scan looking for those keywords among the website objects.

The results list with the response code and the size of the file for each ping. Also, dirb starts searching the files of the folder which returns the response code as 200. It searches the entire folders with the wordlist and displays the results.



File Actions Edit View Help
(root⊙ kali)-[~]
└# dirb https://wwwcom/
DIRB v2.22
By The Dark Raver
ly, another exception occurred while executing the custom error page for the first exception. The request has been terminated.
START_TIME: Fri Aug 6 01:32:29 2021
WORDLIST FILES: /usr/share/dirb/wordlists/common.txt
GENERATED WORDS: 4612
Scopping UDL: https://www.com/
=> DIRECTORY: https://www.left.com/app/
+ https://wwwcom/aspnet_client (CODE:403 SIZE:58)
+ https://www.com/aux (CODE:500 SIZE:1763)
+ https://wwwcom/com1 (CODE:500 SIZE:1763)
+ https://wwwcom/com2 (CODE:500 SIZE:1763)
+ https://www.iiom/app/lpt2 (CODE:500 SIZE:1763)
+ https://www.iiom/app/nul (CODE:500 SIZE:1763)
+ https://www.in
—— Entering directory: https://www.itcp/careers/ ——
(1) FATAL: Too many errors connecting to host
(Possible cause: OPERATION TIMEOUT)
END TIME: Eri Aug. 6.02:24:20.2021
DOWNLOADED: 9328 - FOUND: 24

Finally, when DIRB is done, it reports back the number of found objects (24 in this case). Note that in the help screen above, we can use the -o switch to send the results to an output file to save the results to a text file.

Testing for Special Vulnerable list

We can use DIRB to test for specific vulnerable objects within specific types of web technologies. Each web technology has different vulnerabilities. They are *NOT all the same. DIRB can help us look for specific vulnerable objects specific to the particular technology.*

In Kali, DIRB has specific wordlists to search for these vulnerable often hidden objects. You can find them at:

kali > cd /usr/share/dirb/wordlists/vuln

Then list the contents of that directory:

Created By: Vikram Solanki

