Name:Ashriya

Date:07/03/2023

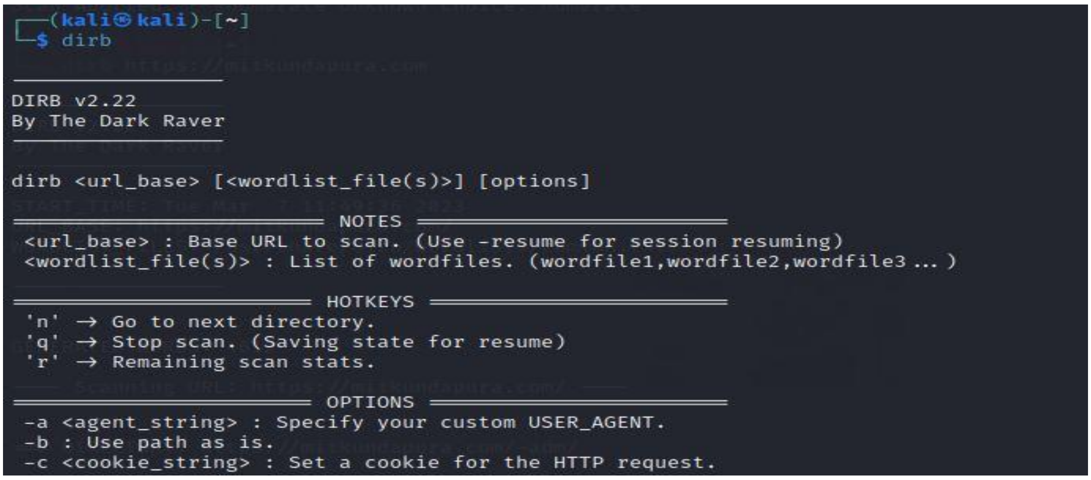
Task:3

1.Drib

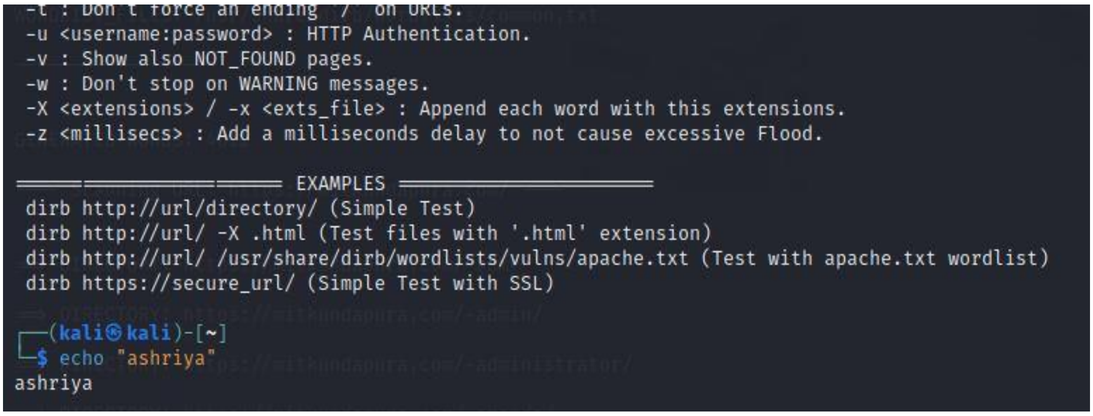
DIRB is a Web Content Scanner. It looks for existing (and/or hidden) Web Objects. It basically works by launching a dictionary based attack against a web server and analyzing the responses. DIRB comes with a set of preconfigured attack wordlists for easy usage but you can use your custom wordlists.

Command:

$ drib

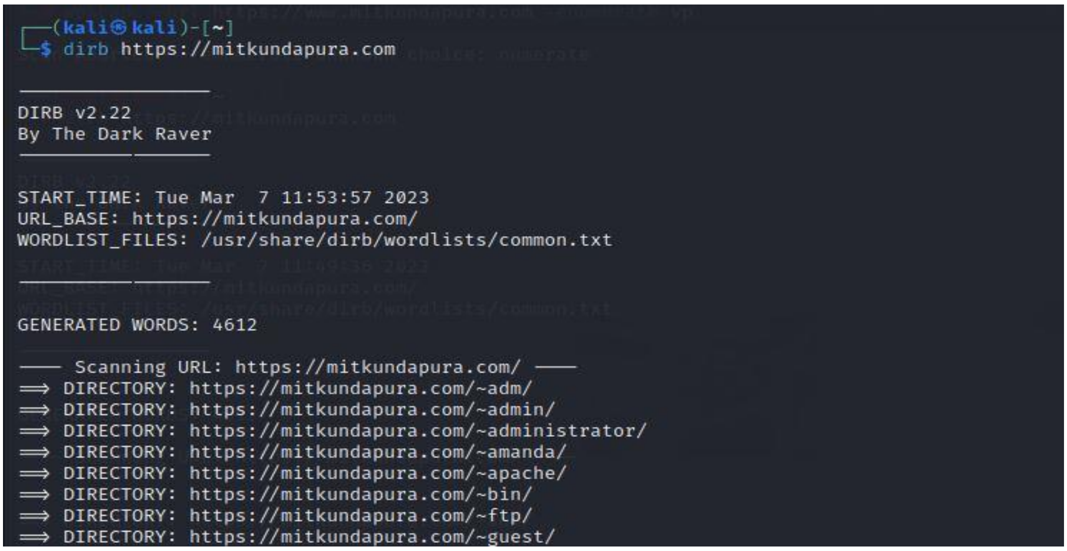


```
┌──(kali㉿kali)-[~]
└─$ dirb

-----------------
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.
```

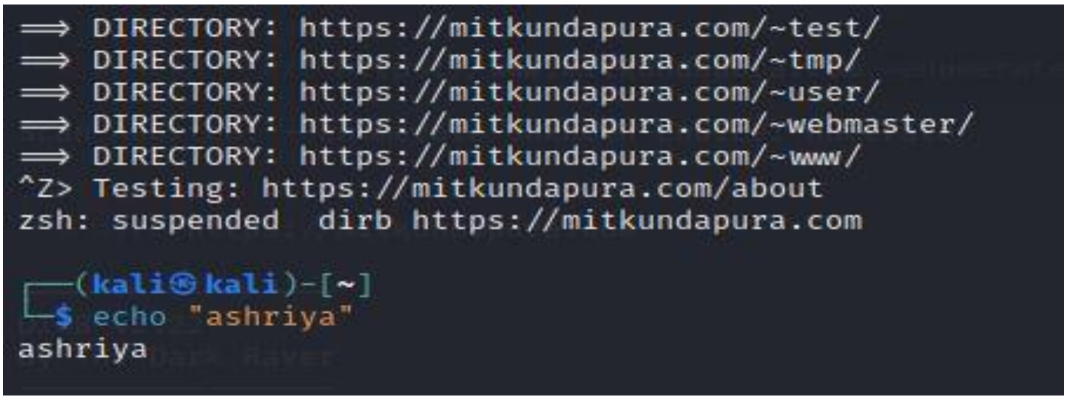


```
 -t : Don't force an ending '/' on URLs.
 -u <username:password> : HTTP Authentication.
 -v : Show also NOT_FOUND pages.
 -w : Don't stop on WARNING messages.
 -X <extensions> / -x <exts_file> : Append each word with this extensions.
 -z <millisecs> : Add a milliseconds delay to not cause excessive 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 apache.txt wordlist)
 dirb https://secure_url/ (Simple Test with SSL)

┌──(kali㉿kali)-[~]
└─$ echo "ashriya"
ashriya
```

$ drib https://mitkundapura.com

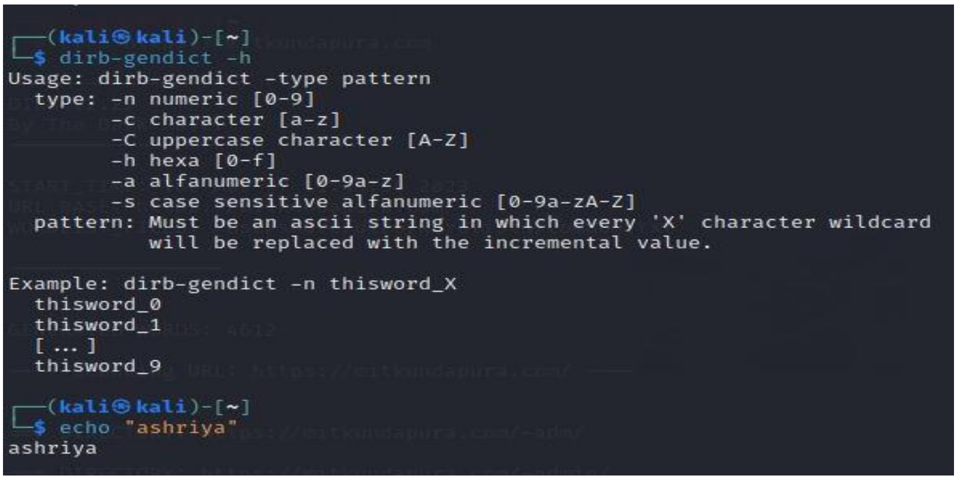


```
┌──(kali㉿kali)-[~]
└─$ dirb https://mitkundapura.com

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

START_TIME: Tue Mar  7 11:53:57 2023
URL_BASE: https://mitkundapura.com/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

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

GENERATED WORDS: 4612

---- Scanning URL: https://mitkundapura.com/ ----
==> DIRECTORY: https://mitkundapura.com/~adm/
==> DIRECTORY: https://mitkundapura.com/~admin/
==> DIRECTORY: https://mitkundapura.com/~administrator/
==> DIRECTORY: https://mitkundapura.com/~amanda/
==> DIRECTORY: https://mitkundapura.com/~apache/
==> DIRECTORY: https://mitkundapura.com/~bin/
==> DIRECTORY: https://mitkundapura.com/~ftp/
==> DIRECTORY: https://mitkundapura.com/~guest/
```



```
==> DIRECTORY: https://mitkundapura.com/~test/
==> DIRECTORY: https://mitkundapura.com/~tmp/
==> DIRECTORY: https://mitkundapura.com/~user/
==> DIRECTORY: https://mitkundapura.com/~webmaster/
==> DIRECTORY: https://mitkundapura.com/~www/
^Z> Testing: https://mitkundapura.com/about
zsh: suspended  dirb https://mitkundapura.com

┌──(kali㉿kali)-[~]
└─$ echo "ashriya"
ashriya
```

$ drib-gendict –h



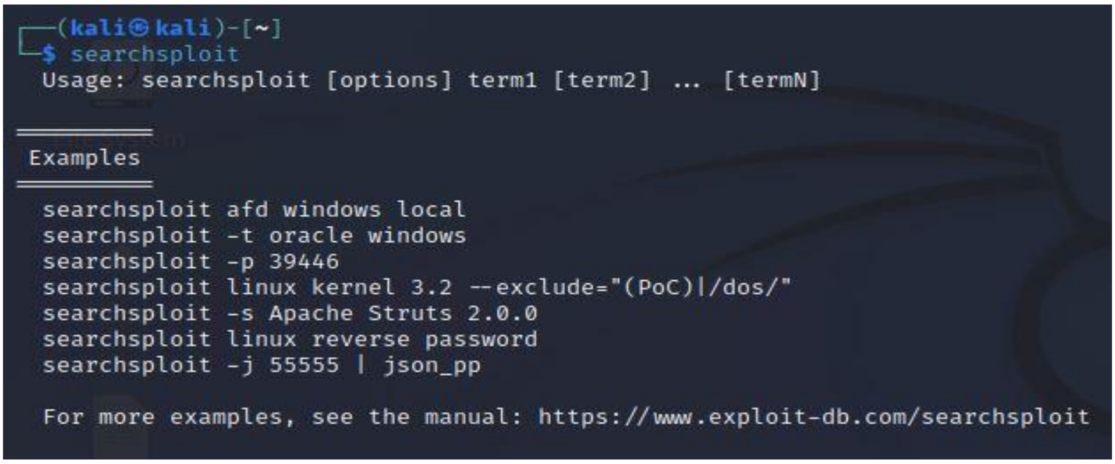
```
┌──(kali㉿kali)-[~]
└─$ dirb-gendict -h
Usage: dirb-gendict -type pattern
  type: -n numeric [0-9]
        -c character [a-z]
        -C uppercase character [A-Z]
        -h hexa [0-f]
        -a alfanumeric [0-9a-z]
        -s case sensitive alfanumeric [0-9a-zA-Z]
  pattern: Must be an ascii string in which every 'X' character wildcard
           will be replaced with the incremental value.

Example: dirb-gendict -n thisword_X
  thisword_0
  thisword_1
  [...]
  thisword_9

┌──(kali㉿kali)-[~]
└─$ echo "ashriya"
ashriya
```

2.Searchsploit

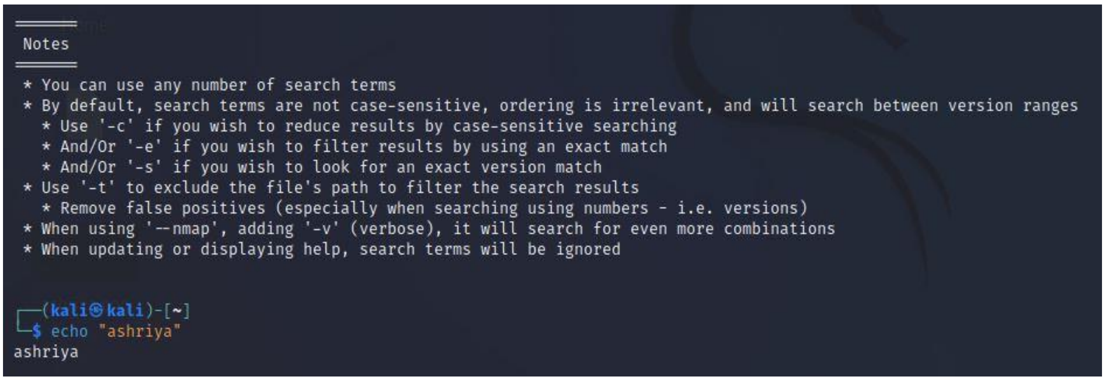
SearchSploit – The Manual. Included in our Exploit Database repository on GitLab is searchsploit, a command line search tool for Exploit-DB that also allows you to take a copy of Exploit Database with you, everywhere you go

Command:

$ searchsploit

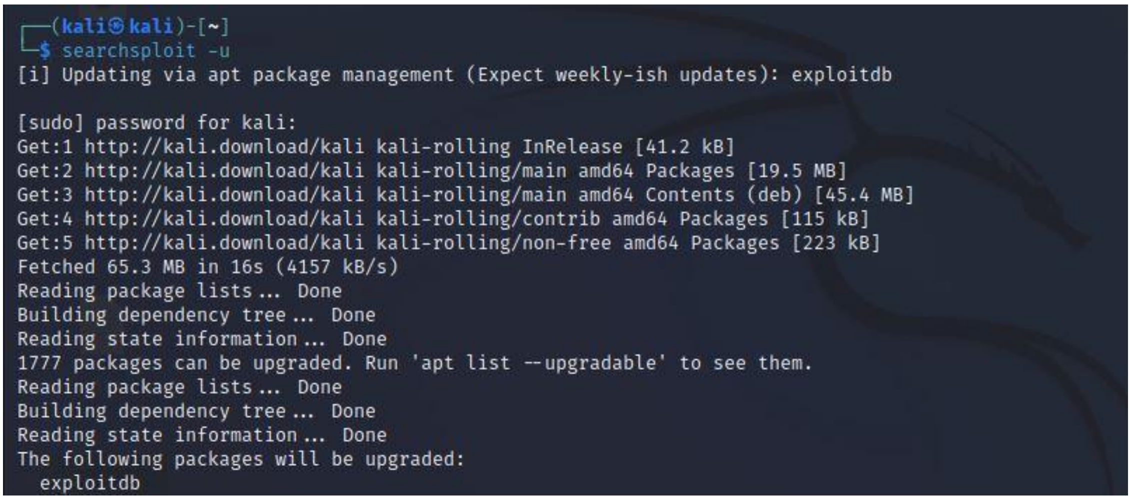


```
┌──(kali㉿kali)-[~]
└─$ searchsploit
  Usage: searchsploit [options] term1 [term2] ... [termN]

==========
 Examples
==========
  searchsploit afd windows local
  searchsploit -t oracle windows
  searchsploit -p 39446
  searchsploit linux kernel 3.2 --exclude="(PoC)|/dos/"
  searchsploit -s Apache Struts 2.0.0
  searchsploit linux reverse password
  searchsploit -j 55555 | json_pp

  For more examples, see the manual: https://www.exploit-db.com/searchsploit
```

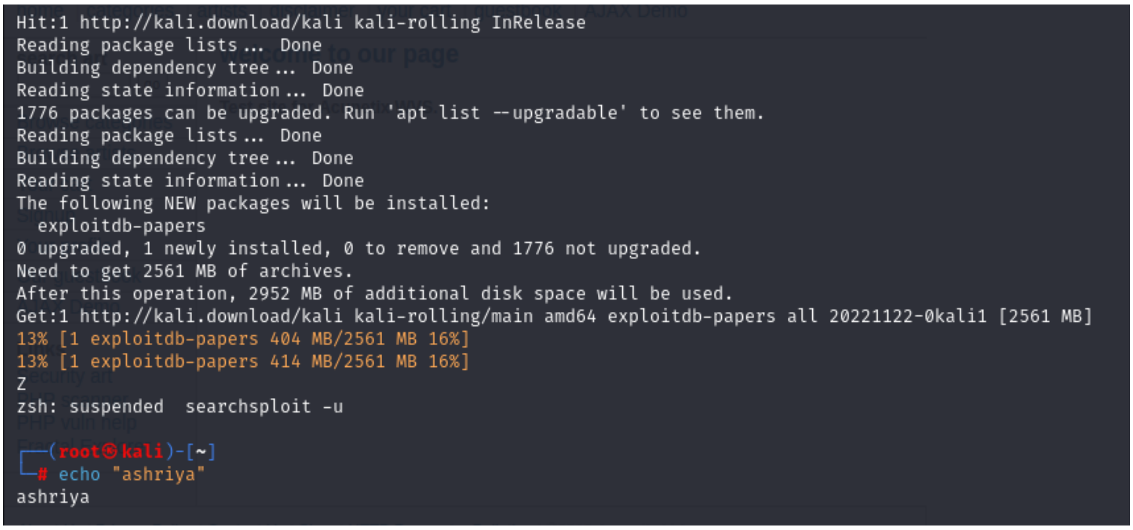


```
=======
 Notes
=======
 * You can use any number of search terms
 * By default, search terms are not case-sensitive, ordering is irrelevant, and will search between version ranges
   * Use '-c' if you wish to reduce results by case-sensitive searching
   * And/Or '-e' if you wish to filter results by using an exact match
   * And/Or '-s' if you wish to look for an exact version match
 * Use '-t' to exclude the file's path to filter the search results
   * Remove false positives (especially when searching using numbers - i.e. versions)
 * When using '--nmap', adding '-v' (verbose), it will search for even more combinations
 * When updating or displaying help, search terms will be ignored


┌──(kali㉿kali)-[~]
└─$ echo "ashriya"
ashriya
```

$ searchsploit –u



```
┌──(kali㉿kali)-[~]
└─$ searchsploit -u
[i] Updating via apt package management (Expect weekly-ish updates): exploitdb

[sudo] password for kali:
Get:1 http://kali.download/kali kali-rolling InRelease [41.2 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [19.5 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [45.4 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [115 kB]
Get:5 http://kali.download/kali kali-rolling/non-free amd64 Packages [223 kB]
Fetched 65.3 MB in 16s (4157 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1777 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages will be upgraded:
  exploitdb
```



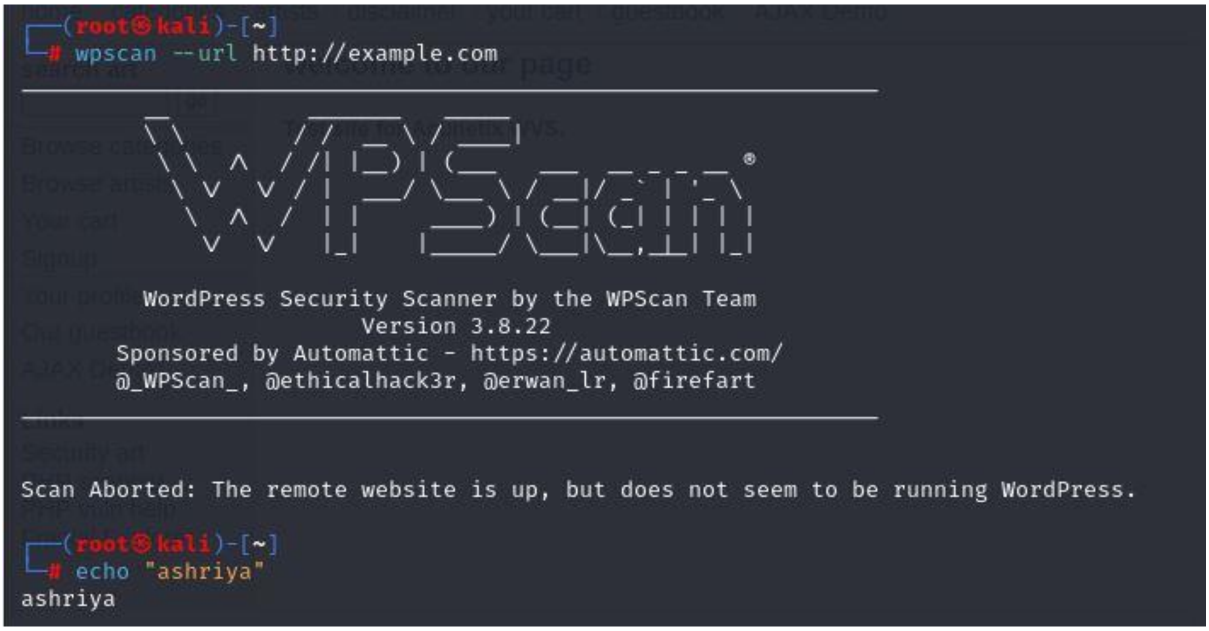
```
Hit:1 http://kali.download/kali kali-rolling InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1776 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  exploitdb-papers
0 upgraded, 1 newly installed, 0 to remove and 1776 not upgraded.
Need to get 2561 MB of archives.
After this operation, 2952 MB of additional disk space will be used.
Get:1 http://kali.download/kali kali-rolling/main amd64 exploitdb-papers all 20221122-0kali1 [2561 MB]
13% [1 exploitdb-papers 404 MB/2561 MB 16%]
13% [1 exploitdb-papers 414 MB/2561 MB 16%]
Z
zsh: suspended  searchsploit -u

┌──(root㉿kali)-[~]
└─# echo "ashriya"
ashriya
```

3.Wpscan

WPScan is a security scanner designed for testing the security of websites built using WordPress. WPScan was developed using the Ruby programming language and then released in the first version in 2019. The WPScan security scanner is primarily intended to be used by WordPress administrators and security teams to assess the security status of their WordPress installations. It is used to scan WordPress websites for known vulnerabilities both in WordPress and commonly used WordPress plugins and themes. The code base for WPScan is licensed under GPLv3.

Command:

$ wpscan –url http://example.com



```
┌──(root㉿kali)-[~]
└─# wpscan --url http://example.com
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.22
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

Scan Aborted: The remote website is up, but does not seem to be running WordPress.

┌──(root㉿kali)-[~]
└─# echo "ashriya"
ashriya
```

4.johntheripper

John the Ripper is a popular open source password cracking tool that combines several different cracking programs and runs in both brute force and dictionary attack modes.

John the Ripper is often used in the enterprise to detect weak passwords that could put network security at risk, as well as other administrative purposes. The software can run a wide variety of password-cracking techniques against the various user accounts on each operating system and can be scripted to run locally or remotely.

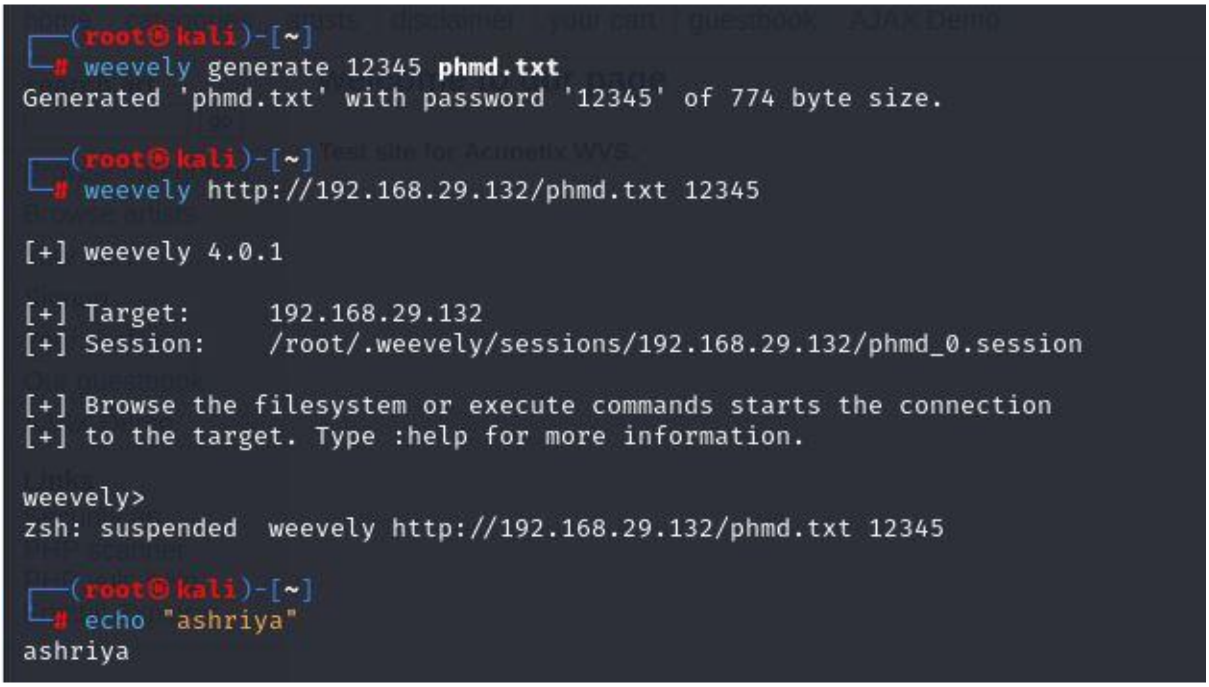
5.Weevely

Weevely is a web shell designed for post-exploitation purposes that can be extended over the network at runtime. Upload weevely PHP agent to a target web server to get remote shell access to it

Command:

$ weevely generate 12345 phmd.txt

$ weevely http://192.168.29.132/phmd.txt 12345



```
┌──(root㉿kali)-[~]
└─# weevely generate 12345 phmd.txt
Generated 'phmd.txt' with password '12345' of 774 byte size.

┌──(root㉿kali)-[~]
└─# weevely http://192.168.29.132/phmd.txt 12345

[+] weevely 4.0.1

[+] Target:     192.168.29.132
[+] Session:    /root/.weevely/sessions/192.168.29.132/phmd_0.session

[+] Browse the filesystem or execute commands starts the connection
[+] to the target. Type :help for more information.

weevely>
zsh: suspended  weevely http://192.168.29.132/phmd.txt 12345

┌──(root㉿kali)-[~]
└─# echo "ashriya"
ashriya
```