Wapiti

Introduction

In Kali Linux, Wapiti is a free and open-source tool. Wapiti allows you to audit the security of your websites or web applications. It performs "black-box" scans (it does not study the source code) of the web application by crawling the webpages of the deployed web app, looking for scripts and forms where it can inject data. Once it gets the list of URLs, forms, and their inputs, Wapiti acts like a fuzzer, injecting payloads to see if a script is vulnerable. Wapiti supports both GET and POST HTTP methods for attacks. It also supports multipart forms and can inject payloads in filenames (upload). Warnings are raised when an anomaly is found (for example 500 errors and timeouts) Wapiti is able to make the difference between permanent and reflected XSS vulnerabilities.

Features

The following are the features of the Wapiti:

- Generates vulnerability reports in various formats (HTML, XML, JSON, TXT, CSV)
- Can suspend and resume a scan or an attack (session mechanism using sqlite3 databases)
- Can give you colors in the terminal to highlight vulnerabilities
- Different levels of verbosity
- Fast and easy way to activate/deactivate attack modules
- Adding a payload can be as easy as adding a line to a text file
- Configurable number of concurrent tasks to perform HTTP requests

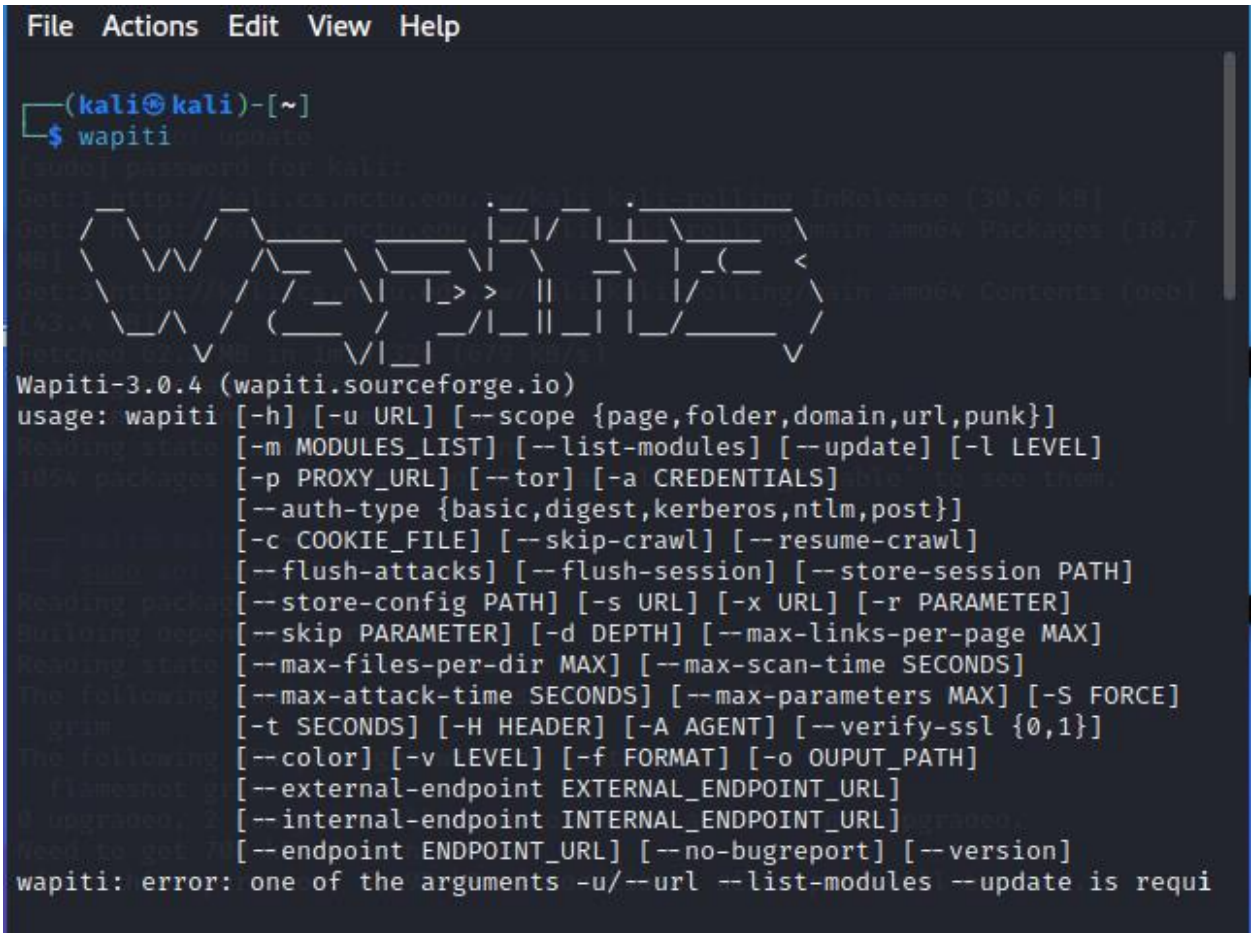
Installation

Wapiti tools are installed in Kali Linux by default.

Usage

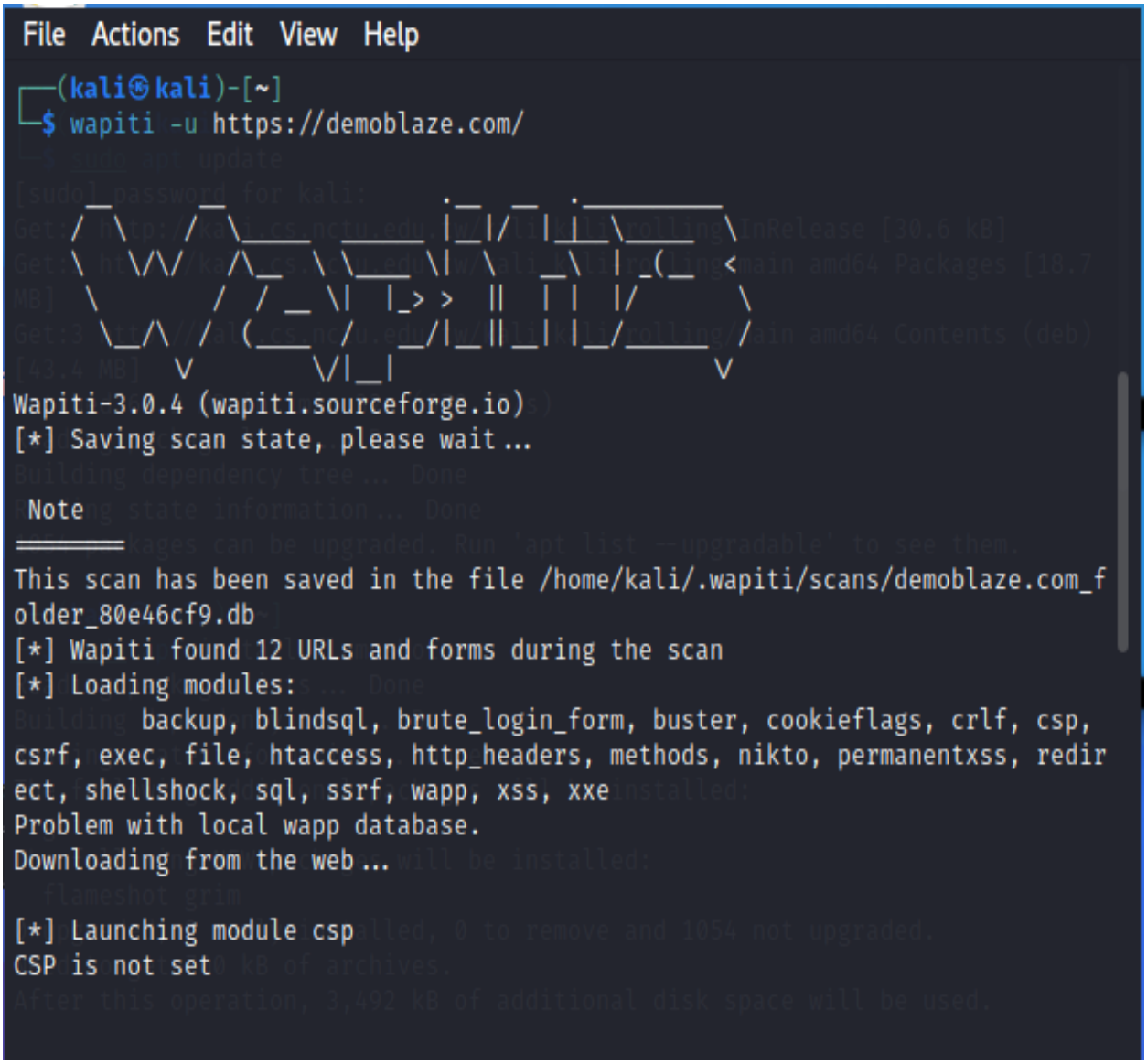
To know about Wapiti tool we can execute the following command:

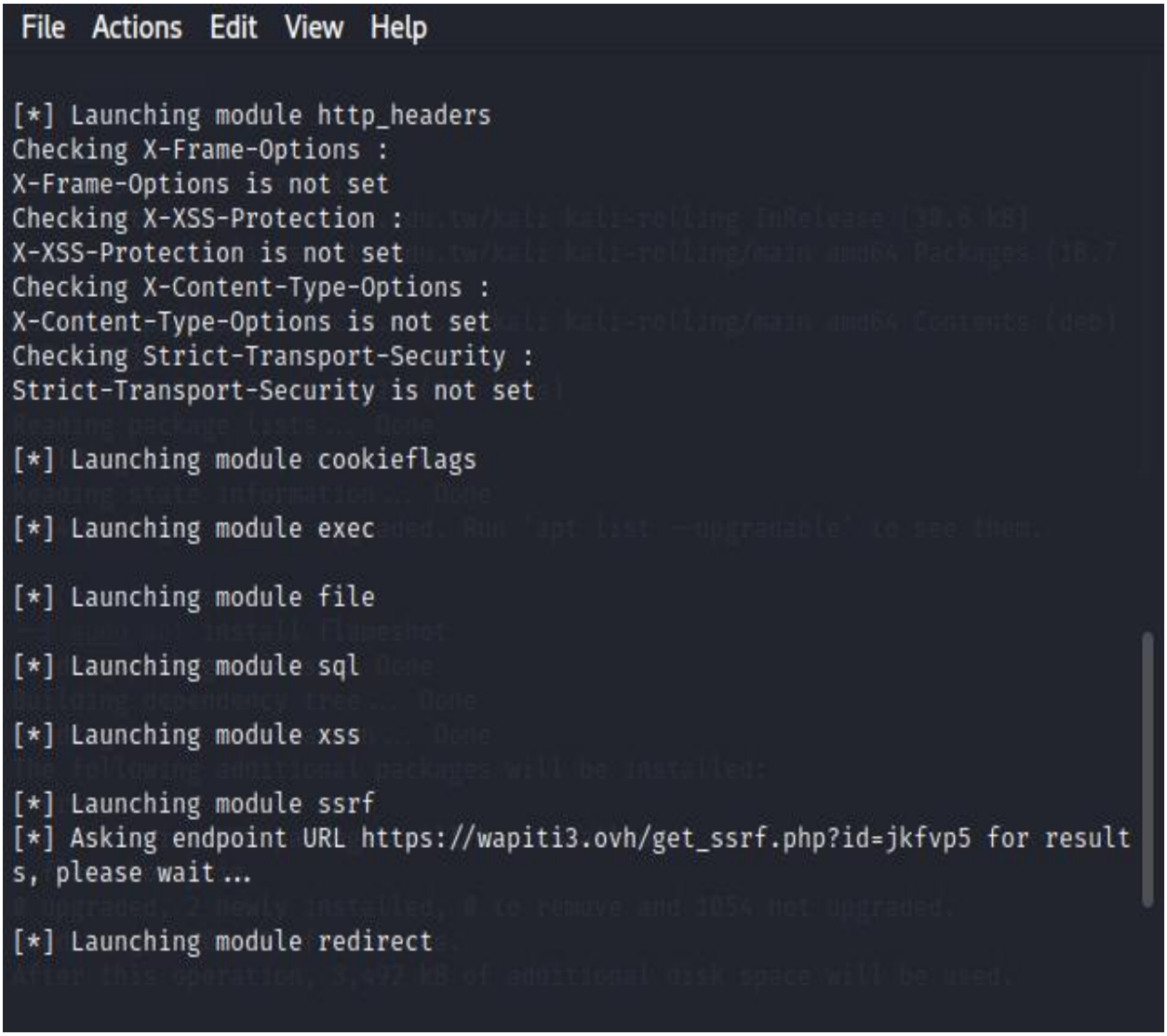
```
Wapiti
```

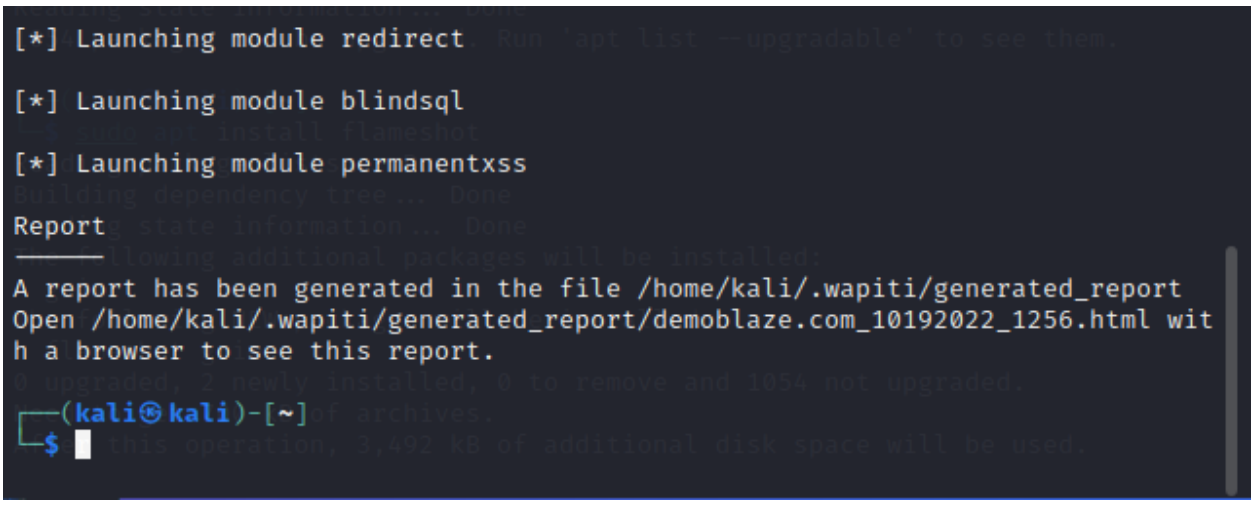


To check or scan the “demolaze.com” website HTML report using the Wapiti tool we can execute the following command:

wapiti –u https://demolaze.com/


File Actions Edit View Help
┌──(kali㉿kali)-[~]
└─$ wapiti -u https://demoblaze.com/
Wapiti-3.0.4 (wapiti.sourceforge.io)
[*] Saving scan state, please wait ...
Note
This scan has been saved in the file /home/kali/.wapiti/scans/demoblaze.com_folder_80e46cf9.db
[*] Wapiti found 12 URLs and forms during the scan
[*] Loading modules:
backup, blindsql, brute_login_form, buster, cookieflags, crlf, csp, csrf, exec, file, htaccess, http_headers, methods, nikto, permanentxss, redirect, shellshock, sql, ssrf, wapp, xss, xxe
Problem with local wapp database.
Downloading from the web ...
[*] Launching module csp
CSP is not set


File Actions Edit View Help
[*] Launching module http_headers
Checking X-Frame-Options :
X-Frame-Options is not set
Checking X-XSS-Protection :
X-XSS-Protection is not set
Checking X-Content-Type-Options :
X-Content-Type-Options is not set
Checking Strict-Transport-Security :
Strict-Transport-Security is not set
[*] Launching module cookieflags
[*] Launching module exec
[*] Launching module file
[*] Launching module sql
[*] Launching module xss
[*] Launching module ssrf
[*] Asking endpoint URL https://wapiti3.ovh/get_ssrf.php?id=jkfvp5 for results, please wait ...
[*] Launching module redirect


[*] Launching module redirect
[*] Launching module blindsql
[*] Launching module permanentxss
Report
A report has been generated in the file /home/kali/.wapiti/generated_report
Open /home/kali/.wapiti/generated_report/demoblaze.com_10192022_1256.html with a browser to see this report.
(kali㉿kali)-[~]
$

References

1. https://wapiti-scanner.github.io/