Web Application Pen-Testing

AY 2022/2023

Week 2.1 Practical

OWASP Top 10 - 2021

A07:2021-Identification and Authentication Failures Part 1

Brute Forcing WordPress Password

Document Version: 3 | Last Updated: 27th Oct 2022

Tools covered:

* Scanning Tools: nmap, whatweb
* Enumeration Tools: dirb, wpscan, ZAP
* Exploitation Tool: ZAP

Major changes to this version v3 compared to the previous version v2 of this document

* Cover page
* Overall changes to heading titles
* Start and Login to Kali Linux VM
* Install OWASP ZAP in Kali Linux

Contents

[OWASP Top 10 – 2021 3](#_Toc117743022)

[A07:2021-Identification and Authentication Failures - Description 3](#_Toc117743023)

[Setup 4](#_Toc117743024)

[Start OWASP Broken Web Apps (owaspbwa) VM 4](#_Toc117743025)

[Start and Login to Kali Linux VM 5](#_Toc117743026)

[Install OWASP ZAP in Kali Linux 6](#_Toc117743027)

[Scanning 7](#_Toc117743028)

[Make sure the Virtual Machine Settings 🡪 Network Adapter 🡪 Host-only 7](#_Toc117743029)

[Find out the Ports 7](#_Toc117743030)

[View Nmap’s help summary page 7](#_Toc117743031)

[Identify the list of services running on the owaspbwa VM using nmap 8](#_Toc117743032)

[Perform further information gathering on the open ports using nmap 9](#_Toc117743033)

[Browse all the websites found using Kali Linux Web Browser 10](#_Toc117743034)

[Obtain the URL of the Wordpress website running on OWASPBWAP VM 11](#_Toc117743035)

[Making sure WordPress is indeed running on http://OWASPBWAP\_IP/wordpress/ 12](#_Toc117743036)

[View WhatWeb’s help summary page 12](#_Toc117743037)

[Perform further information gathering on the WordPress Content Management System 13](#_Toc117743038)

[Enumeration 14](#_Toc117743039)

[Finding Directories and Files using dirb 14](#_Toc117743040)

[View dirb’s help summary page 14](#_Toc117743041)

[Enumerating http://OWASPBWAP\_IP/wordpress/ Directories, Files, and Pages 15](#_Toc117743042)

[Enumerating Directory with Specific Extension List 16](#_Toc117743043)

[Enumerate Usernames for http://OWASPBWAP\_IP/wordpress using wpscan 17](#_Toc117743044)

[View wpscan’s help summary page 17](#_Toc117743045)

[Enumerating users using wpscan 18](#_Toc117743046)

[Spidering 19](#_Toc117743047)

[Start OWASP ZAP inside Kali Linux 19](#_Toc117743048)

[Click “Automated Scan” 19](#_Toc117743049)

[Discovery 21](#_Toc117743050)

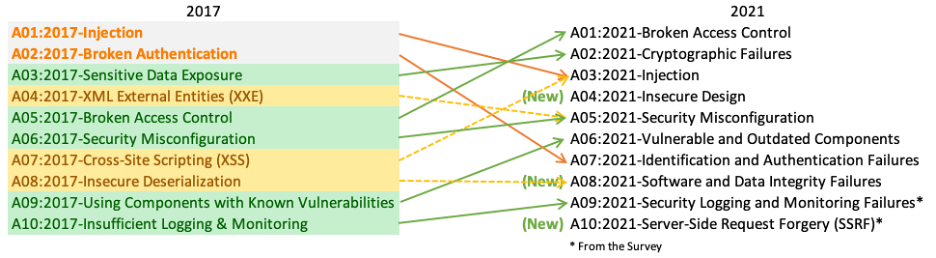
[Open URL http://OWASPBWAP\_IP/wordpress/wp-login.php in System Browser 21](#_Toc117743051)

[Exploitation 23](#_Toc117743052)

[Using OWASP ZAP Fuzz Feature 23](#_Toc117743053)

A07:2021-Identification and Authentication Failures

# OWASP Top 10 – 2021



A07:2021-Identification and Authentication Failures was previously Broken Authentication and is sliding down from the second position, and now includes Common Weakness Enumeration (CWEs) that are more related to identification failures. This category is still an integral part of the Top 10, but the increased availability of standardized frameworks seems to be helping.

# A07:2021-Identification and Authentication Failures - Description

Confirmation of the user's identity, authentication, and session management is critical to protect against authentication-related attacks. There may be authentication weaknesses if the application:

* Permits automated attacks such as credential stuffing, where the attacker has a list of valid usernames and passwords.
* Permits brute force or other automated attacks.
* Permits default, weak, or well-known passwords, such as "Password1" or "admin/admin".
* Uses weak or ineffective credential recovery and forgot-password processes, such as "knowledge-based answers," which cannot be made safe.
* Uses plain text, encrypted, or weakly hashed passwords data stores (see [A02:2021-Cryptographic Failures](https://owasp.org/Top10/A02_2021-Cryptographic_Failures/)).
* Has missing or ineffective multi-factor authentication.
* Exposes session identifier in the URL.
* Reuse session identifier after successful login.
* Does not correctly invalidate Session IDs. User sessions or authentication tokens (mainly single sign-on (SSO) tokens) aren't properly invalidated during logout or a period of inactivity.

[Source: <https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/>]

# Scanning

## Make sure the Virtual Machine Settings 🡪 Network Adapter 🡪 Host-only

## ****Find out the Ports****

Websites usually run-on Port 80 and 443 and other common ports are 8000, 8080, 8443. But they can run on any port. Use Nmap to find out:

It is a good habit to know your tool’s various options, by using sudo TOOL\_NAME -h:

### View Nmap’s help summary page

Type the following command into the Kali Linux’s Terminal Emulator and press Enter:

sudo nmap -h

Text

Description automatically generated

### Identify the list of services running on the owaspbwa VM using nmap

Type the following command into the Kali Linux’s Terminal Emulator and press Enter:

sudo nmap -sS -Pn -T4 -p- 192.168.184.137

* -sS: TCP SYN scans
* -Pn: Treat all hosts as online -- skip host discovery
* -T<0-5>: Set timing template (higher is faster)
* -p <port ranges>: Only scan specified ports
* OWASPBWA\_IP: owaspbwa VM IP Address

In the output below we can identify ports 80, and 443 are open and are running the services http, and https respectively.

Text

Description automatically generated

### Perform further information gathering on the open ports using nmap

Type the following command into the Kali Linux’s Terminal Emulator and press Enter:

sudo nmap -O -A -Pn -T4 -p80,443 192.168.184.137

* -O: Enable OS detection
* -A: Enable OS detection, version detection, script scanning, and traceroute
* -Pn: Treat all hosts as online -- skip host discovery
* -T<0-5>: Set timing template (higher is faster)
* -p <port ranges>: Only scan specified ports
* OWASPBWA\_IP: owaspbwa VM IP Address

In the output below we can identify that port 80 is open and running the http service using the Apache webserver version 2.2.14. Same is the case with port 443 as well. The landing page’s http-title is also indicated. Finally, the Server OS details: Linux 2.6.17 - 2.6.36 can also be seen.

Text

Description automatically generated

### Browse all the websites found using Kali Linux Web Browser

This will make sure that we don’t miss websites running on non-standard ports.

Type the following into the address bar of the Kali Linux Web Browser and hit enter:

http://OWASPBWA\_IP:80

In the output below we notice that we are redirected to <http://OWASPBWA_IP>. We see a list of several websites. However, in this practical exercise we will **focus on “WordPress” website**.

Graphical user interface

Description automatically generated with medium confidence

Type the following into the address bar of the Kali Linux Web Browser and hit enter:

https://OWASPBWA\_IP:443

We will notice an error: “Secure Connection Failed”. We will not proceed further with this, instead we will use the webpage that was displayed using the above http://OWASPBWA\_IP:80

### Obtain the URL of the Wordpress website running on OWASPBWAP VM

Type the following into the address bar of the Kali Linux Web Browser and hit enter:

http://OWASPBWA\_IP

We see a list of several websites. However, in this practical exercise we will **focus on “WordPress” website**. Click on the link “WordPress”.

Graphical user interface

Description automatically generated with medium confidence

Take note and copy the URL displayed in the address bar of the Kali Linux Web Browser, it should indicate: http://OWASPBWAP\_IP/wordpress/

A screenshot of a computer

Description automatically generated

## Making sure WordPress is indeed running on http://OWASPBWAP\_IP/wordpress/

Even though the hyperlink inside http://OWASPBWAP\_IP is listed as “WordPress” we need to make sure that WordPress content management system (CMS) is indeed running on this webserver (Apache 2.2.14). We would also want to know what version of WordPress system is running. WordPress is one of the most popular CMS solutions in use, WordPress is used by 42.8% of the top 10 million websites as of October 2021. [Source: <https://en.wikipedia.org/wiki/WordPress>].

We use WhatWeb tool. WhatWeb identifies websites. It recognises web technologies including content management systems (CMS), blogging platforms, statistic/analytics packages, JavaScript libraries, web servers, and embedded devices. WhatWeb has over 900 plugins, each to recognise something different. It also identifies version numbers, email addresses, account IDs, web framework modules, SQL errors, and more. [Source: <https://www.kali.org/tools/whatweb/>]

### View WhatWeb’s help summary page

Type the following command into the Kali Linux’s Terminal Emulator and press Enter:

sudo whatweb -h

Text

Description automatically generated with low confidence

### Perform further information gathering on the WordPress Content Management System

Type the following command into the Kali Linux’s Terminal Emulator and press Enter:

sudo whatweb -a 3 http://OWASPBWA\_IP/wordpress

* -a 3: An aggressive scan of http://OWASPBWA\_IP/wordpress detects the exact version of WordPress.
  + AGGRESSION:
  + The aggression level controls the trade-off between speed/stealth and reliability.
    - --aggression, -a=LEVEL: Set the aggression level. Default: 1.
    - 1. Stealthy: Makes one HTTP request per target and also follows redirects.
    - 3. Aggressive: If a level 1 plugin is matched, additional requests will be made.
    - 4. Heavy: Makes a lot of HTTP requests per target. URLs from all plugins are attempted.

In the output below, we can notice that WordPress 2.0 is running on this webserver.

Text, chat or text message

Description automatically generated

[OPTIONAL] For more verbose details:

Type the following command into the Kali Linux’s Terminal Emulator and press Enter:

sudo whatweb -v -a 3 http://OWASPBWA\_IP/wordpress

* --verbose, -v: Verbose output includes plugin descriptions.
* -a 3: An aggressive scan of http://OWASPBWA\_IP/wordpress detects the exact version of WordPress.

# Enumeration

## Finding Directories and Files using 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 analyzing the responses. DIRB comes with a set of preconfigured attack wordlists located at /usr/share/dirb/wordlists/ for easy usage, but you can use your custom wordlists. Wordlists are generated by crawling the Internet and collecting the directory and files that are used by developers. Also, DIRB sometimes can be used as a classic CGI scanner but remember that it is a content scanner not a vulnerability scanner. [Source: <https://www.kali.org/tools/dirb/>]

To have a detailed description on the usage of the DIRB tool refer to this webpage: <https://www.hackingarticles.in/comprehensive-guide-on-dirb-tool/>wp-

### View dirb’s help summary page

Type the following command into the Kali Linux’s Terminal Emulator and press Enter:

sudo dirb

Text

Description automatically generated

### Enumerating <http://OWASPBWAP_IP/wordpress/> Directories, Files, and Pages

Type the following command into the Kali Linux’s Terminal Emulator and press Enter:

sudo dirb <http://OWASPBWAP_IP/wordpress/> -N 302

* -N <nf\_code>: Ignore responses with this HTTP code.

In the output below, among many that DIRB displayed to us, we can notice 2 interesting details, on which we can attempt password brute forcing attacks:

* DIRECTORY: http:// OWASPBWAP\_IP/wordpress/wp-admin/
* http://OWASPBWAP\_IP/wordpress/wp-login

Text

Description automatically generated

### Enumerating Directory with Specific Extension List

Type the following command into the Kali Linux’s Terminal Emulator and press Enter:

sudo dirb <http://OWASPBWAP_IP/wordpress/> -N 302 -X .php

* -N <nf\_code>: Ignore responses with this HTTP code.
* X <extensions> / -x <exts\_file>: Append each word with these extensions. The above command will extract all directory path related to php extension

In the output below, among many that DIRB displayed to us, we can notice 1 interesting detail, on which we can attempt password brute forcing attack:

http://OWASPBWAP\_IP/wordpress/wp-login.php

Text

Description automatically generated

## Enumerate Usernames for http://OWASPBWAP\_IP/wordpress using wpscan

WPScan scans remote WordPress installations to find security issues.

[Source: <https://www.kali.org/tools/wpscan/>]

### View wpscan’s help summary page

Type the following command into the Kali Linux’s Terminal Emulator and press Enter:

sudo wpscan -h

Text

Description automatically generatedText

Description automatically generated

### Enumerating users using wpscan

Type the following command into the Kali Linux’s Terminal Emulator and press Enter:

sudo wpscan --url http://OWASPBWAP\_IP/wordpress --enumerate u

* --url URL: The URL of the blog to scan
* --enumerate [OPTS]: Enumeration Process
  + u: User IDs range.

In the output below, among many that wpscan displayed to us, we can notice towards the end 1 interesting detail, under the User(s) Identified:, “admin” is listed. Therefore, we now know that “admin” is a valid username used for this wordpress website. wpscan tool has many other uses which we will visit in subsequent exercises.

[NOTE: If wpscan throws an error stating it needs to update database, change the Kali Linux’s network adapter setting to NAT and then execute the wpscan command. This allows wpscan to connect to the Internet to update its database. Once there is no more such an error, you can return the Kali Linux, network adapter setting to Host-only]

Text

Description automatically generatedText

Description automatically generated

## Spidering

In web application pen-testing, a crawler or spider is a tool that automatically goes through a website following all links in it and sometimes filling in and sending forms; this allows us to get a complete map of all the referenced pages within the site and record the requests made to get them and their responses.

### Start OWASP ZAP inside Kali Linux

Graphical user interface, text, application

Description automatically generated

### Click “Automated Scan”

Graphical user interface, application, Word

Description automatically generated

Type in <http://OWASPBWAP_IP/wordpress> in the “URL to attack” text field. Check “Use traditional spider” and click “Attack”. Note: Error Pop Up may appear, click “OK” and click “Attack” again.

Graphical user interface, text, application, email

Description automatically generated

Click on “Spider” tab and once the progress bar reaches 100%, you can click on “Stop”. On the left pane expand “Sites”, “http: OWASPBWAP\_IP”, and “wordpress” folder.

Graphical user interface, text

Description automatically generated

In the output below, among many pages that ZAP crawled and displayed to us, we can notice 1 interesting detail, on which we can attempt password brute forcing attack:

POST:wp-login.php()(log,pwd,redirect\_to,rememberme,submit)

In the right pane, you can explore the “Request” and “Response” tabs detailing the Request sent by ZAP, and the Response received by ZAP from http://OWASPBWAP\_IP/wordpress.

Graphical user interface, text, application

Description automatically generated

# Discovery

## Open URL http://OWASPBWAP\_IP/wordpress/wp-login.php in System Browser

In the left pane, right click on

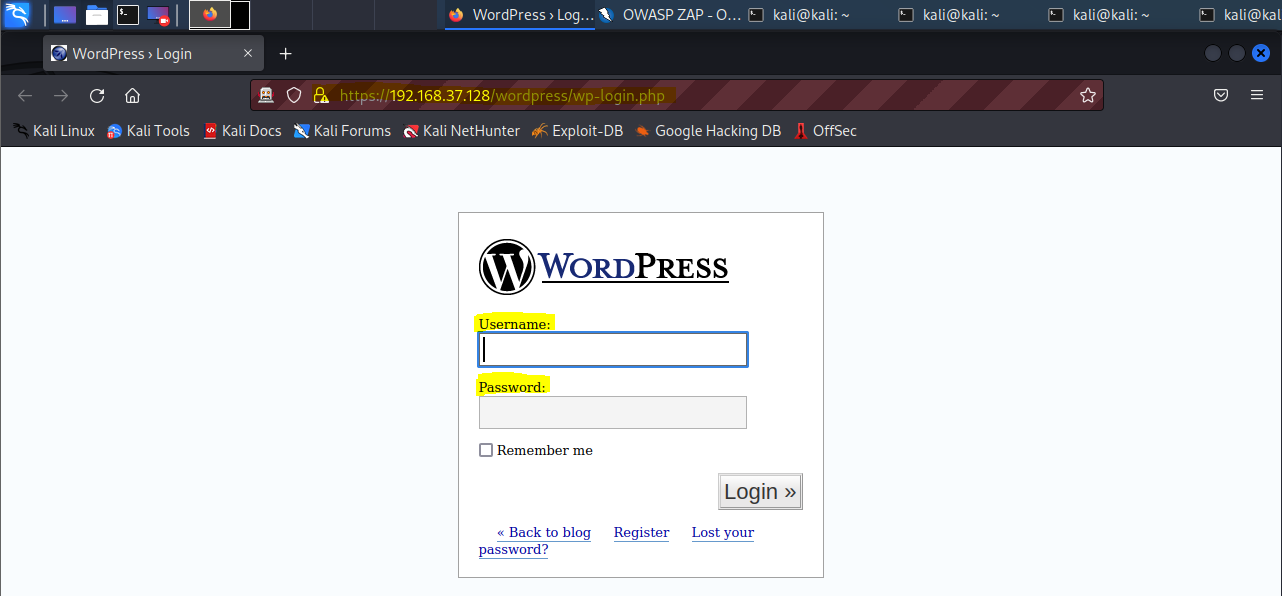
“POST:wp-login.php()(log,pwd,redirect\_to,rememberme,submit)”

Select “Open URL in Browser” and select “Firefox”.

Graphical user interface, text, application

Description automatically generated

The wp-login.php page will be displayed as shown below



Enter “admin” as Username (note that we used wpscan tool previously to identify this valid username). Enter a random Password and observe that an Error: Invalid Password is displayed. You may try to repeatedly enter random passwords and you will realize that there is no account lockout feature on this website. This is just right for a Brute Force Attacking of the password.

Graphical user interface, text, application, email, website

Description automatically generated

# Exploitation

## Using OWASP ZAP Fuzz Feature

In the left pane, click on

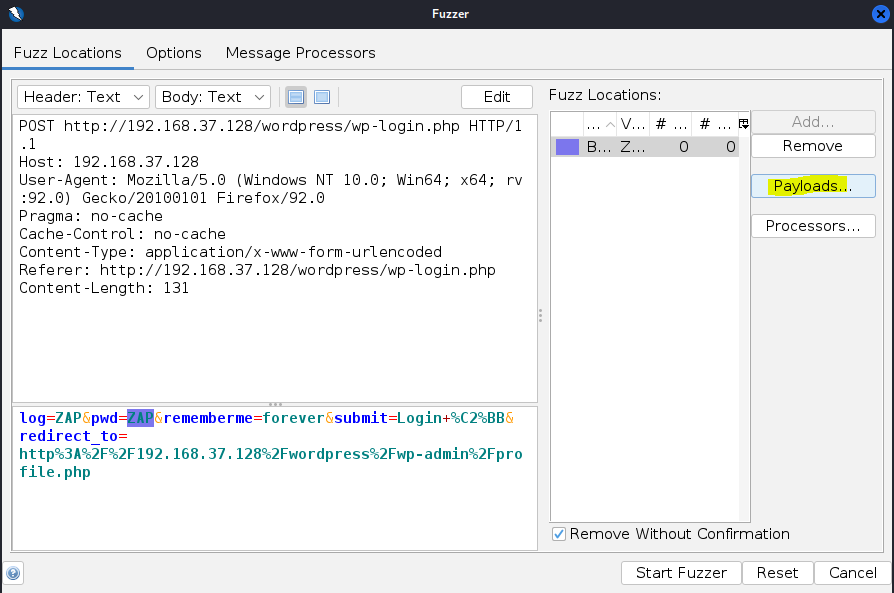
“POST:wp-login.php()(log,pwd,redirect\_to,rememberme,submit)”

On the right pane click on “Request” tab and in the right bottom pane use your mouse to highlight the only the “ZAP” in the pwd=”ZAP” and then right click and select “Fuzz”.

Graphical user interface, text, application

Description automatically generated

Click on “Payloads…”



Click on “Add…”

Graphical user interface, application

Description automatically generated

Using the drop-down menu to change Type: File Fuzzers 🡪 fuzzdb 🡪 wordlists-user-passwd 🡪 generic-listpairs 🡪 http\_default\_pass.txt

Graphical user interface, text, application

Description automatically generated

Click “OK”

Graphical user interface, application

Description automatically generated

In the bottom pane, use your mouse to highlight the only the “ZAP” in the log=”ZAP” and then click “Add…”.

Graphical user interface, text, application

Description automatically generated

Follow the steps below:

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated

After setting up both the positions for fuzzing it should look like this:

Graphical user interface, text, application

Description automatically generated

Setup Message Processors to easily identify failed attempts. Click “Add…”, then in the Add Message Processor drop-down menu select “Tag Creator” and click “Add”.

Graphical user interface, application

Description automatically generated

Enter “Error: Incorrect password.” in Regex and “Failed” in Tag fields.

Graphical user interface, application

Description automatically generated

Click the “Top” button on the right to make sure that our Tag Creator goes to the top of the list as shown below and click “Start Fuzzer”.

Graphical user interface, text, email

Description automatically generated

Successfully detected the Username and Password combination: admin, admin

A picture containing graphical user interface

Description automatically generated