*Web Application Pen-Testing*

*AY 2022/2023*

*Total Recall Challenge*

*(A challenge to recall all the skills you learnt thus far)*

*Submission Deadline: Sunday 4th Dec 2023 23:59*

*Pen-Test WackoPicko Website*

*Table of Contents*

[*1.* *WackoPicko* 3](#_Toc120093110)

[*1.1.* *Valid Logins for reference* 3](#_Toc120093111)

[*2.* *Reconnaissance Phase – Scanning* 3](#_Toc120093112)

[*2.1.* *Output from nmap tool – ports, services, server OS, etc. (Ref. Week 2.1)* 3](#_Toc120093114)

[*2.2.* *Output from whatweb tool – webserver OS, etc. (Ref. Week 2.1)* 4](#_Toc120093115)

[*2.3.* *Output from curl tool – HTTP methods supported by webserver (Ref. Week 1.2)* 4](#_Toc120093116)

[*2.4.* *Any other Scanning tools used (Optional)* 4](#_Toc120093117)

[*3.* *Reconnaissance Phase – Enumeration* 5](#_Toc120093118)

[*3.1.* *Output from dirb tool (Ref. Week 2.1)* 5](#_Toc120093120)

[*3.2.* *Output from ZAP Spidering tool (Ref. Week 2.1)* 5](#_Toc120093121)

[*3.3.* *Any other Enumeration tools used (Optional)* 5](#_Toc120093122)

[*3.4.* *Manually browse and understand the WackoPicko website as if you are a customer of this website and note down what interesting aspects you observed* 5](#_Toc120093123)

[*4.* *Discovery Phase – Automated Vulnerability Scan* 6](#_Toc120093124)

[*4.1.* *Output from Authenticated Scan using ZAP (Ref. Week 4.1)* 6](#_Toc120093129)

[*4.2.* *Any other Automated Vulnerability Scanners used (Optional)* 7](#_Toc120093130)

[*5.* *Exploitation & Post Exploitation Phase* 7](#_Toc120093131)

[*5.1.* *XSS Related Vulnerabilities (Ref. Week 6.1)* 7](#_Toc120093133)

[*5.2.* *Session ID Vulnerability (Ref. Week 2.2)* 7](#_Toc120093134)

[*5.3.* *SQL Injection Vulnerability (Ref. Week 4.1)* 8](#_Toc120093135)

[*5.4.* *Command-Line Injection Vulnerability (Ref. Week 4.2)* 8](#_Toc120093136)

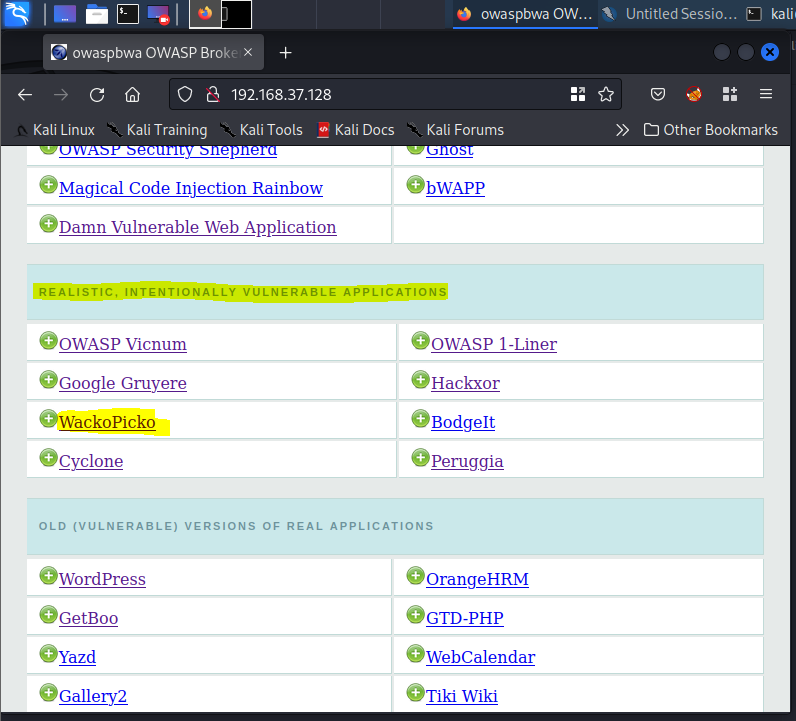
[*5.5.* *Forceful Browsing, File Inclusion, Parameter Manipulation (Ref. Week 3)* 8](#_Toc120093137)

[*5.6.* *Weak Username / password (Ref. Week 2.1 & 2.2)* 9](#_Toc120093138)

[*5.7.* *Logic Flaw (Try it Out)* 9](#_Toc120093139)

# *WackoPicko*

WackoPicko is a website that contains known vulnerabilities. It is available inside the OWASPBWA VM as shown below.



Please fill up this document with ONLY important screenshots mainly indicating the tool used and the outputs observed. Submit this filled up document via POLITEMall by Sunday 4th Dec 2023 23:59.

## *Valid Logins for reference*

* Regular users
* scanner1/scanner1
* scanner2/scanner2
* bryce/Bryce

# *Reconnaissance Phase – Scanning*



## *Output from nmap tool – ports, services, server OS, etc. (Ref. Week 2.1)*

|  |
| --- |
|  |

## *Output from whatweb tool – webserver OS, etc. (Ref. Week 2.1)*

|  |
| --- |
|  |

## *Output from curl tool – HTTP methods supported by webserver (Ref. Week 1.2)*

|  |
| --- |
|  |

## *Any other Scanning tools used (Optional)*

|  |
| --- |
|  |

# *Reconnaissance Phase – Enumeration*



## *Output from dirb tool (Ref. Week 2.1)*

|  |
| --- |
|  |

## *Output from ZAP Spidering tool (Ref. Week 2.1)*

|  |
| --- |
|  |

## *Any other Enumeration tools used (Optional)*

|  |
| --- |
|  |

## *Manually browse and understand the WackoPicko website as if you are a customer of this website and note down what interesting aspects you observed*

Understand what the website is doing, e.g., you can register an account, login using the registered account, and explore other pages that are visible after your login, etc.

|  |
| --- |
| When registering an account, there are no password requirements.  Users can upload non-jpg or png file extensions when uploading pictures.  HTML tags are not sanitized for the comments on pictures  The search bar do not sanitize its HTML tags. |

# *Discovery Phase – Automated Vulnerability Scan*

WackoPicko is known to have the following vulnerabilities. See if you can discover (scan), exploit, and post exploit these vulnerabilities (as many as possible).

|  |  |
| --- | --- |
| 1. Reflected XSS  - The query parameter is vulnerable. | 9. Command-line Injection  - The password field is vulnerable to a command line injections. |
| 2. Stored XSS  - The comment field is vulnerable. | 10. File Inclusion  - The page is vulnerable to a file inclusion vulnerability, however you have to include %00 at the end. |
| 3. SessionID vulnerability  - The session cookie value is admin\_session, which is an auto-incrementing value. | 11. Parameter Manipulation  - The userid parameter can be manipulated to see any user's page when you need to be logged in otherwise. |
| 4. Stored SQL Injection  - The first name field of the register users form contains a stored SQL injection which is then used unsanitized on the similar users page. | 12. Reflected XSS Behind JavaScript  - The name parameter is vulnerable. |
| 5. Reflected SQL Injection  - The username field is vulnerable. | 13. Logic Flaw  - A coupon can be applied multiple times reducing the price of an order to zero. The coupon in the initial data is SUPERYOU21. |
| 6. Directory Traversal  - The tag field has a directory traversal vulnerability enabling a malicious users to overwrite any file the web server uses has access to. | 14. Reflected XSS Behind a Flash Form  - The value parameter is vulnerable. |
| 7. Multi-Step Stored XSS  - The comment field is vulnerable to XSS, however must go through a preview form. | 15. Weak username/password  - There is a default username/password combination of admin/admin. |
| 8. Forceful Browsing  - The user doesn't have to purchase the picture to see the high quality version. |  |



## *Output from Authenticated Scan using ZAP (Ref. Week 4.1)*

While doing the authenticated scan please scan for individual vulnerabilities listed above rather than scan the entire website, it might take a long time and the output could be quite confusing.

|  |
| --- |
| Calender.php  Guestbook.php (POST)  Passcheck.php  Passcheck.php (POST)        These are the vulnerabilites detected from the active scan on the webpages mentioned above.  In total, there are 37 of them. |

## *Any other Automated Vulnerability Scanners used (Optional)*

#### [14 best open-source web application vulnerability scanners <https://resources.infosecinstitute.com/topic/14-popular-web-application-vulnerability-scanners/>]

|  |
| --- |
|  |

# *Exploitation & Post Exploitation Phase*



## *XSS Related Vulnerabilities (Ref. Week 6.1)*

|  |
| --- |
| Stored/Persistent XSS      Reflected XSS |

## *Session ID Vulnerability (Ref. Week 2.2)*

|  |
| --- |
| Graphical user interface, text, application, email  Description automatically generated  Graphical user interface, text, application  Description automatically generated |
| The session ID are very predictable as they are not random but rather go from 1 to 20000. |

## *SQL Injection Vulnerability (Ref. Week 4.1)*

|  |
| --- |
| Graphical user interface, text  Description automatically generated  Graphical user interface, text, application, email  Description automatically generated  Graphical user interface, application  Description automatically generated  Graphical user interface, application  Description automatically generated |

## *Command-Line Injection Vulnerability (Ref. Week 4.2)*

|  |
| --- |
| Graphical user interface, text  Description automatically generated  Text  Description automatically generated |
| Graphical user interface, text  Description automatically generated |
| Graphical user interface, application  Description automatically generated |

## *Forceful Browsing, File Inclusion, Parameter Manipulation (Ref. Week 3)*

|  |
| --- |
| * **Forceful Browsing**   **Text  Description automatically generated**  **Text  Description automatically generated** |

## *Weak Username / password (Ref. Week 2.1 & 2.2)*

|  |
| --- |
| Graphical user interface, application  Description automatically generated  Graphical user interface, application  Description automatically generated  Graphical user interface, text, application  Description automatically generated |

## *Logic Flaw (Try it Out)*

|  |
| --- |
|  |