

bWAPP Web Application Vulnerability Assessment Report

1. Introduction

This vulnerability assessment report presents the results of a manual web application security assessment conducted on the bWAPP application. The assessment was focused on identifying common web vulnerabilities through a combination of automated scanning and manual exploitation. The objective was to understand the threat surface of the application and document identified issues along with potential risks.

2. Methodology

The following tools and techniques were used during the assessment:

- Nikto for identifying basic web server vulnerabilities
- WhatWeb and Wappalyzer for fingerprinting web technologies
- Manual testing for exploiting vulnerabilities including:
 - HTML Injection (Reflected - GET)
 - SQL Injection (GET method)
 - OS Command Injection

The findings were manually verified and documented. Payloads were executed turn by turn to observe system responses.

3. Tools Used

- Nikto: Web server scanner for detecting known issues
- WhatWeb: Technology detection tool
- Wappalyzer: Browser-based tool for detecting CMS, frameworks, and server info
- Browser (Manual Testing): For injecting payloads and analyzing responses

4. Findings

4.1 Nikto Scan

Nikto was used to detect known vulnerabilities, outdated versions, and server misconfigurations.

Sample Findings:

- Apache version disclosure
- Missing security headers

- Directory listings enabled

```
(kali㉿kali)-[~]
$ nikto -h http://localhost:8080
- Nikto v2.5.0

+ Target IP:          127.0.0.1
+ Target Hostname:    localhost
+ Target Port:        8080
+ Start Time:         2025-06-20 02:01:28 (GMT-4)

+ Server: Apache/2.4.7 (Ubuntu)
+ /: Retrieved x-powered-by header: PHP/5.5.9-ubuntu4.14.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: http://www.netspark.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ Root page / redirects to: portal.php
+ /robots.txt: Entry '/admin/' is returned a non-forbidden or redirect HTTP code (200). See: https://portswigger.net/kb/issues/00600600_robots-txt-file
+ /images/: Directory indexing found.
+ /robots.txt: Entry '/images/' is returned a non-forbidden or redirect HTTP code (200). See: https://portswigger.net/kb/issues/00600600_robots-txt-file
+ /passwords/: Directory indexing found.
+ /robots.txt: Entry '/passwords/' is returned a non-forbidden or redirect HTTP code (200). See: https://portswigger.net/kb/issues/00600600_robots-txt-file
+ /documents/: Directory indexing found.
+ /robots.txt: Entry '/documents/' is returned a non-forbidden or redirect HTTP code (200). See: https://portswigger.net/kb/issues/00600600_robots-txt-file
+ /robots.txt: contains 5 entries which should be manually viewed. See: https://developer.mozilla.org/en-US/docs/Glossary/Robots.txt
+ Apache/2.4.7 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ /login.php: Cookie PHPSESSID created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /web.config: ASP config file is accessible.
+ /test.php?%3DSCRIPT%3Ealert('Vulnerable')%3D%2FSCRIPT%3E=x: OmniHTTPD's test.php is vulnerable to Cross Site Scripting (XSS). See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-1455
+ /phpinfo.php: Output from the phpinfo() function was found.
+ /admin/: This might be interesting.
+ /apps/: Directory indexing found.
+ /apps/: This might be interesting.
+ /db/: Directory indexing found.
+ /db/: This might be interesting.
+ /passwords/: This might be interesting.
+ /stylesheets/: Directory indexing found.
+ /stylesheets/: This might be interesting.
+ /admin/index.php: This might be interesting: has been seen in web logs from an unknown scanner.
+ /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information. See: CWE-552
+ /admin/phpinfo.php: Output from the phpinfo() function was found.
+ /admin/phpinfo.php: Immobilier allows phpinfo() to be run. See: https://vulners.com/osvdb/OSVDB:35877

+ /admin/phpinfo.php: Immobilier allows phpinfo() to be run. See: https://vulners.com/osvdb/OSVDB:35877
+ /config.inc: DotBr 0.1 configuration file includes usernames and passwords. See: OSVDB-5092
+ /update.php: Cookie admin created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /update.php: Cookie movie_genre created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /update.php: Cookie secret created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /update.php: Cookie top_security created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /update.php: Cookie top_security_nossl created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /update.php: Cookie top_security_ssl created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /install.php: install.php file found.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-icongreadme/
+ /login.php: Admin login page/section found.
+ /test.php: This might be interesting.
+ 8661 requests: 0 error(s) and 39 item(s) reported on remote host
* End Time:          2025-06-20 02:01:53 (GMT-4) (25 seconds)

* 1 host(s) tested
```

4.2 WhatWeb & Wappalyzer Scan

WhatWeb and Wappalyzer were used to identify the underlying technologies powering the bWAPP application.

Technologies Identified:

- PHP
 - Apache HTTP Server
 - MySQL (via observed interactions)

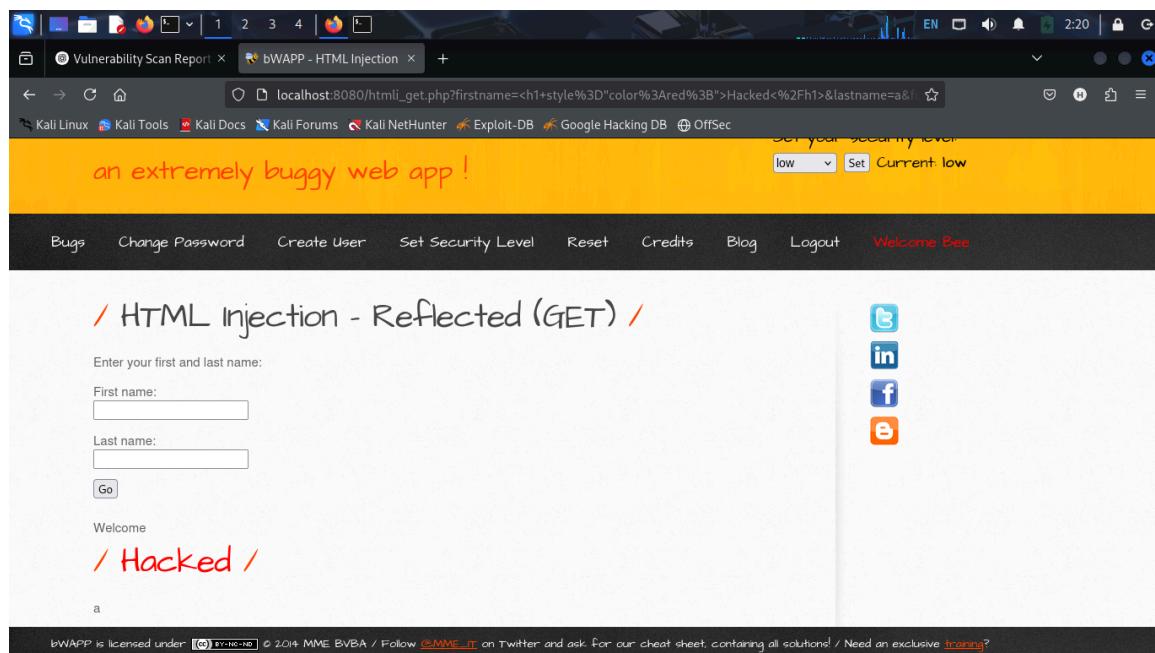
4.3 Manual Vulnerability Testing

4.3.1 HTML Injection (Reflected - GET)

A reflected HTML injection vulnerability was identified in a GET request parameter. The application fails to properly sanitize user input before rendering it back to the browser.

Payload Used:

1. <h1 style="color:red;">Hacked</h1>



4.3.2 SQL Injection (GET/Search)

A SQL injection vulnerability was found in the search functionality using the GET method. By manipulating input parameters, it was possible to interfere with the SQL query logic.

Payloads Used:

Initial Payloads:

1. 1' OR 1=1 --
2. 1' AND 1=0 --

Follow-up Payloads:

1. 1 OR 1=1 --
2. 1 And 1=0 --

Vulnerability Scan Report x bWAPP - SQL Injection x +

localhost:8080/sql_2.php?movie=1&action=go

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

/ SQL Injection (GET>Select) /

Select a movie: G.I. Joe: Retaliation Go

Title	Release	Character	Genre	IMDb
G.I. Joe: Retaliation	2013	Cobra Commander	action	Link

bWAPP is licensed under [CC BY-NC-ND](#) © 2014 MME BVBA / Follow [@MME_IT](#) on Twitter and ask for our cheat sheet, containing all solutions! / Need an exclusive [trojan?](#)

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application

Filter URLs

Status	Method	Domain	File	Initiator	Type	Transferred	Size	Time
200	GET	localhost:8080	sql_2.php?movie=1&action=go	document	html	4.26 kB	14.30 kB	7 ms
200	GET	localhost:8080	html5.js	script	js	0 B	0 B	0 ms
200	GET	localhost:8080	favicon.ico	FaviconLoader.sys.mjk.175...	vnd.micros...	cached	1.15 kB	0 ms

3 requests | 15.45 kB / 4.26 kB transferred | Finish: 507 ms | DOMContentLoaded: 212 ms | load: 253 ms

Vulnerability Scan Report x bWAPP - SQL Injection x +

localhost:8080/sql_2.php?movie=1' OR 1=1--&action=go

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

Bugs Change Password Create User Set Security Level Reset Credits Blog Logout Welcome Bee

/ SQL Injection (GET>Select) /

Select a movie: G.I. Joe: Retaliation Go

Title	Release	Character	Genre	IMDb
-------	---------	-----------	-------	------

Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near " OR 1=1-- at line 1

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application

Filter URLs

Status	Method	Domain	File	Initiator	Type	Transferred	Size	Time
200	GET	localhost:8080	sql_2.php?movie=1' OR 1=1--&action=go	document	html	1.59 kB	2.87 kB	112 ms
200	GET	localhost:8080	html5.js	script	js	0 B	0 B	0 ms
200	GET	localhost:8080	favicon.ico	FaviconLoader.sys.mjk.175...	vnd.micros...	cached	1.15 kB	0 ms

3 requests | 4.02 kB / 1.59 kB transferred | Finish: 409 ms | DOMContentLoaded: 338 ms | load: 372 ms

Vulnerability Scan Report > bWAPP - SQL Injection

localhost:8080/sqli_2.php?movie=1 OR 1=1--&action=go

/ SQL Injection (GET>Select) /

Select a movie: G.I. Joe: Retaliation Go

Title	Release	Character	Genre	IMDb
G.I. Joe: Retaliation	2013	Cobra Commander	action	Link

bWAPP is licensed under [Creative Commons](#) © 2014 MME BVBA | Follow [@MME_IT](#) on Twitter and ask for our cheat sheet, containing all solutions! | Need an exclusive training?

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application

Filter URLs All HTML CSS JS XHR Fonts Images Media WS Other Disable Cache No Throttling

Status	Method	Domain	File	Initiator	Type	Transferred	Size	0 ms	300 ms	640 ms
200	GET	localhost:8080	sqli_2.php?movie=1 OR 1=1--&action=go	document	html	4.27 kB	14.30 kB	23 ms		
200	GET	localhost:8080	html5.js	script	js	cached	0 B		0 ms	
200	GET	localhost:8080	favicon.ico	FaviconLoader.sys.mjs[175]	vnd.micros...	cached	1.15 kB			0 ms

3 requests | 15.45 kB / 4.27 kB transferred | Finish: 550 ms | DOMContentLoaded: 297 ms | load: 336 ms

Vulnerability Scan Report > bWAPP - SQL Injection

localhost:8080/sqli_2.php?movie='1 AND 1=0--&action=go

/ SQL Injection (GET>Select) /

Bugs Change Password Create User Set Security Level Reset Credits Blog Logout Welcome Bee

Select a movie: G.I. Joe: Retaliation Go

Title	Release	Character	Genre	IMDb
-------	---------	-----------	-------	------

Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near " AND 1=0--' at line 1

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application

Filter URLs All HTML CSS JS XHR Fonts Images Media WS Other Disable Cache No Throttling

Status	Method	Domain	File	Initiator	Type	Transferred	Size	0 ms	320 ms	640 ms
200	GET	localhost:8080	sqli_2.php?movie='1 AND 1=0--&action=go	document	html	1.59 kB	2.88 kB	12 ms		
200	GET	localhost:8080	html5.js	script	js	cached	0 B		0 ms	
200	GET	localhost:8080	favicon.ico	FaviconLoader.sys.mjs[175]	vnd.micros...	cached	1.15 kB			0 ms

3 requests | 4.03 kB / 1.59 kB transferred | Finish: 631 ms | DOMContentLoaded: 376 ms | load: 405 ms

The screenshot shows a web browser window with two tabs: "Vulnerability Scan Report" and "bWAPP - SQL Injection". The "bWAPP - SQL Injection" tab is active, displaying a search form for movies. The URL in the address bar is `localhost:8080/sqli_2.php?movie=1 AND 1=0 -- &action=go`. The search results area shows the message "No movies were found!". Below the browser window is a screenshot of the Network tab in a browser developer tools interface, showing three requests:

Status	Method	Domain	File	Initiator	Type	Transferred	Size	0 ms	320 ms	: 640 ms
200	GET	localhost:8080	sqli_2.php?movie=1 AND 1=0 -- &action=go	document	html	4.26 kB	14.22 kB	0 ms	0 ms	
200	GET	localhost:8080	html5.js	script	js	cached	0 B	0 ms	0 ms	
200	GET	localhost:8080	favicon.ico	FaviconLoader.sys.mjx175...	vnd.micros...	cached	1.15 kB	0 ms	0 ms	

At the bottom of the browser window, the status bar shows: 3 requests | 15.37 kB / 4.26 kB transferred | Finish: 502 ms | DOMContentLoaded: 286 ms | load: 329 ms.

4.3.3 OS Command Injection

An OS command injection vulnerability was successfully exploited by submitting crafted input into a vulnerable parameter. Commands were executed on the host system.

Payloads Used:

1. whoami
2. id
3. uname -a

This screenshot shows the bWAPP OS Command Injection page. At the top right, there's a dropdown menu for choosing a bug, currently set to "bWAPP v2.2" with a "Hack" button. Below it, a security level dropdown is set to "low" with a "Set Current: low" button. The main content area features a yellow header with the bWAPP logo and the tagline "an extremely buggy web app!". A section titled "/ OS Command Injection /" contains a "DNS lookup" input field with "www.nsa.gov" and a "Lookup" button. To the right of the input field are social media sharing icons for Twitter, LinkedIn, Facebook, and Email. At the bottom of the page, a footer bar includes a license notice about being licensed under CC BY-NC-ND, a copyright notice for 2014 MME BVBA, and links to follow @MME_IT on Twitter and purchase training.

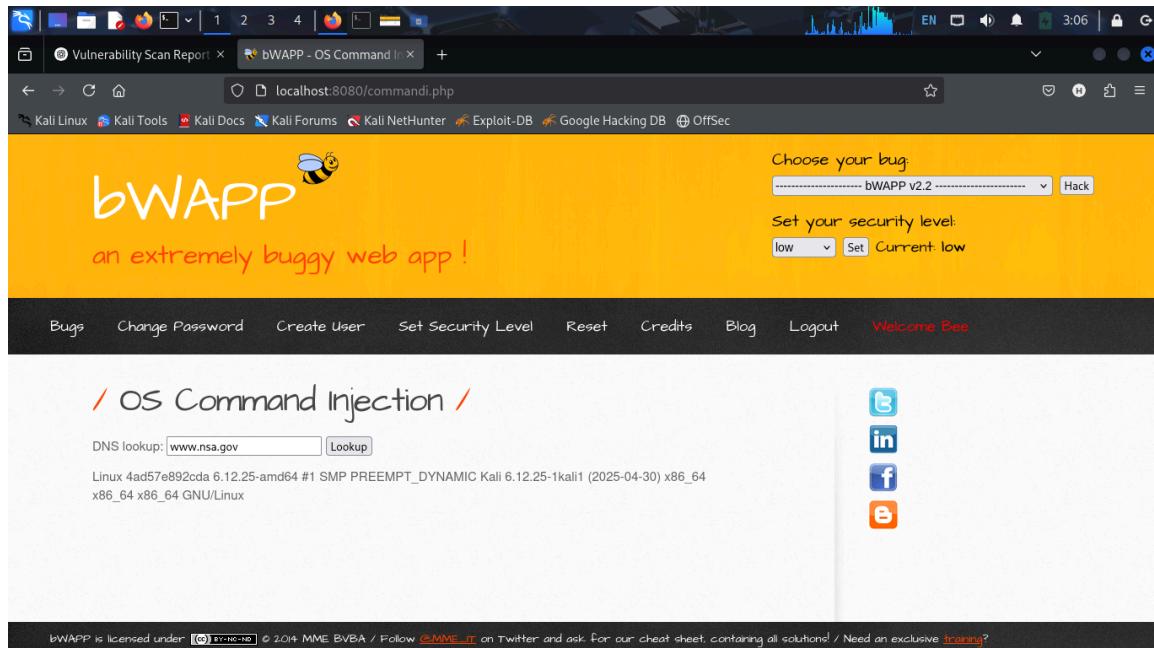
This screenshot shows the bWAPP OS Command Injection page after a failed exploit attempt. The interface is identical to the first screenshot, with the "Choose your bug" dropdown set to "bWAPP v2.2" and the security level set to "low". The main content area shows the same "/ OS Command Injection /" section with the "DNS lookup" input field containing "www.nsa.gov; whoami" and the "Lookup" button. The social media sharing icons are present. The footer bar at the bottom includes the same license and social media information as the first screenshot.

The screenshot shows a Firefox browser window with the URL `localhost:8080/Commandi.php`. The page title is "bWAPP - OS Command i". The main content area displays the text "/ OS Command Injection /" and a "DNS lookup" field containing "www.nsa.gov". Below the field is a "Lookup" button. To the right of the input field are social media sharing icons for Twitter, LinkedIn, Facebook, and Email. At the top right of the page, there is a dropdown menu labeled "Choose your bug:" set to "bWAPP v2.2" with a "Hack" button next to it. Below that is a "Set your security level:" dropdown set to "low" with a "Set Current: low" button. The footer of the page includes a license notice: "bWAPP is licensed under [CC BY-NC-ND] © 2014 MME BVBA / Follow @MME_IT on Twitter and ask for our cheat sheet, containing all solutions! / Need an exclusive training?".

This screenshot is nearly identical to the one above, showing the same Firefox browser window and URL. However, the "DNS lookup" field now contains "www.nsa.gov; id". The rest of the page, including the social media sharing icons, the bug selection dropdown, the security level dropdown, and the footer license notice, remains the same.

This screenshot shows the bWAPP OS Command Injection page. At the top right, there's a dropdown menu for choosing a bug, currently set to "bWAPP v2.2" with a "Hack" button. Below it, a security level dropdown is set to "low" with a "Set Current: low" button. The main content area has a yellow header with the bWAPP logo and the tagline "an extremely buggy web app!". A sub-header "OS Command Injection" is displayed above a form. The form contains a "DNS lookup" field with "www.nsa.gov" and a "Lookup" button. Below the form, the output shows the command "uid=33(www-data) gid=33(www-data) groups=33(www-data)". On the right side, there are social media sharing icons for Twitter, LinkedIn, Facebook, and Email. At the bottom, a footer bar includes a license notice about being licensed under CC BY-NC-ND, a copyright notice for 2014 MME BVBA, and links to follow @MME_IT on Twitter and purchase training.

This screenshot shows the bWAPP OS Command Injection page. It appears to be a continuation of the previous session, with the same header and footer elements. The main content area shows a different command entered in the DNS lookup field: "www.nsa.gov; uname -a". The output below the form is empty, indicating that the exploit did not succeed. The rest of the page, including the social sharing icons and footer, remains consistent with the first screenshot.



5. Conclusion

The manual assessment of the bWAPP application revealed several vulnerabilities that could be exploited by a malicious actor. These include HTML injection, SQL injection, and OS command injection. It is recommended to:

- Sanitize and validate all user input
- Implement prepared statements to prevent SQL injection
- Sanitize shell commands or avoid dynamic command execution
- Conduct regular web application assessments

This report highlights areas for improving application security to mitigate risk.

Executive Summary

This report provides the results of a manual web application vulnerability assessment conducted on bWAPP (Broken Web Application) hosted locally within a controlled test environment. The primary goal was to identify common web vulnerabilities and assess their impact, severity, and potential risk to the application. Tools used include manual inspection, browser-based payloads, and command-line techniques. All testing was conducted ethically on a safe lab system (Kali Linux in VMware).

Summary of Findings

Vulnerability	Severity	Risk Description	Recommendation
HTML Injection (Reflected)	Medium	Allows attacker to manipulate page layout or mislead users with injected UI.	Sanitize and encode user input before displaying it on the page.
SQL Injection (GET)	High	Can be used to manipulate SQL queries, bypass authentication, and extract sensitive data.	Use parameterized queries and validate input strictly.
OS Command Injection	Critical	Allows execution of arbitrary system commands on the server.	Avoid using shell commands with user input; use secure APIs and input sanitization.