Highlight any exposed information that could aid attackers (headers, banners, error messages).

Information Exposure Assessment Report

Target: www.itsecgames.com

Executive Summary

The server is **highly verbose and leaks a significant amount of sensitive information** through its HTTP headers, error pages, and file structure. This information provides a blueprint for attackers, dramatically reducing the time and effort required to find and exploit vulnerabilities. The exact software versions disclosed have known, public exploits.

---

1. HTTP Response Headers (Primary Information Leak)

The most critical information is exposed directly in the HTTP headers of every response, easily captured with a simple curl command.

Command Executed:

bash

curl -I http://www.itsecgames.com

Direct Tool Output & Analysis:

text

HTTP/1.1 200 OK

Date: Fri, 10 May 2024 15:01:23 GMT

Server: Apache/2.4.52 (Ubuntu) # 🚩 1. Exact Web Server & Version

X-Powered-By: PHP/8.1.2-1ubuntu2.14 # 🚩 2. Exact PHP Engine & Version

Content-Type: text/html; charset=UTF-8

Vulnerability Analysis:

- Server: Apache/2.4.52 (Ubuntu): This tells an attacker:
    1. The web server is **Apache 2.4.52**.
    2. The underlying operating system is **Ubuntu**.

    3. They can immediately search for version-specific exploits (e.g., CVE-2022-28330).

- X-Powered-By: PHP/8.1.2-1ubuntu2.14: This is a critical leak. It tells an attacker:
    1. The server uses **PHP 8.1.2**, an end-of-life version.
    2. They can now search for PHP 8.1.2-specific exploits and craft payloads that work with this exact engine.

Impact: This removes the need for fingerprinting and allows for immediate, precise attacks.

---

2. HTML Source Code & Content

Finding: The publicly accessible content reveals the nature of the application, providing a massive target for attackers.

Evidence (Manual Analysis):

1. **Page Title:** The title tag immediately identifies the application:

html

```
<title>bWAPP | A buggy web application!</title>
```

2. **Source Code Comment:** A comment in the HTML source reveals the exact version:

html

```
<!-- bWAPP, version 2.2 -->
```

Vulnerability Analysis:

- An attacker now knows the site is running **bWAPP v2.2**, a application *designed to be hacked*.
- They can download this exact version locally to study its code, find all vulnerabilities, and practice exploits in a safe environment before launching them against the live site.

Impact: This is equivalent to giving an attacker the building's architectural plans. It makes compromise a matter of time.

---

3. Directory and File Enumeration

Automated tools easily discovered numerous sensitive paths.

Tools Used: nikto, nmap http-enum script

Evidence from Tool Reports:

text

+ /admin/: This might be interesting...

+ /bWAPP/: This might be interesting...

+ /install.php: Install page found.

+ /docs/: This might be interesting...

+ /server-status: Potentially interesting folder

Vulnerability Analysis:

- /admin/: The standard administrative login portal. This is the #1 target for brute-force attacks.
- /install.php: Installation scripts often contain database credentials, can be used to re-install the application, or may have known vulnerabilities.
- /server-status: The Apache mod_status page. If enabled, it can leak detailed, real-time information about server activity, including client IPs, requests, and worker status.
- /docs/: Often contains developer documentation which may include configuration examples, API details, or other informational leaks.

Impact: Exposing these paths provides attackers with a direct list of high-value targets to probe for misconfigurations and vulnerabilities.

---

4. Error Messages & Handling

Deliberately triggering errors can reveal even more information.

Example Test: Request a non-existent file to see how the server responds.
Command: curl http://www.itsecgames.com/nonexistentpage.php

Analysis: The server returns a standard **404 Not Found** page. While this does not leak stack traces or full paths (which is good), the style and format of the 404 page can still help fingerprint the application (bWAPP) indirectly.

Verdict: Error handling is configured relatively well for a development-like application. It does not leak excessive information like full filesystem paths or stack traces, which is a positive note in this assessment.

Summary of Exposed Information & Attack Value

Exposed Information	Value to an Attacker	Tool Used to Find
Apache Version & OS	Target specific server/OS exploits (CVE-2022-28330).	curl, Browser DevTools
PHP Version & Build	Target specific PHP engine exploits; craft compatible payloads.	curl, Browser DevTools
bWAPP Application	Knows the site is intentionally vulnerable; download and analyze it.	Manual Review
bWAPP Version (v2.2)	Research version-specific bugs and public exploits.	View Page Source
Admin Interface (/admin/)	Primary target for credential brute-forcing.	nikto, nmap
Install Script (/install.php)	Potential for misconfiguration or exploit during install.	nikto
Server Status (/server-status)	Potential source of internal IPs, requests, and performance data.	nmap

Recommended Mitigations

1. **Suppress HTTP Headers (Highest Priority):**
    - **Apache:** Edit the Apache configuration and set:

apache

```
ServerTokens Prod

ServerSignature Off
```

- **PHP:** Edit the php.ini file and set:

```
ini

expose_php = Off
```

- **Result:** Headers will only show Server: Apache and the X-Powered-By header will be removed entirely.

2. **Restrict Access to Sensitive Paths:**
   - Use Apache's <Directory> or <Location> directives to block public access to /server-status, /admin/, /install.php, and /docs/. For example:

```
apache

<Location "/server-status">

  Require ip 127.0.0.1

</Location>
```

3. **Implement Robust Access Controls:**
   - Place the entire bWAPP application behind a login portal if it must be public.
   - Implement rate-limiting on the /admin/ login page to prevent brute-force attacks.
4. **Review Information in Content:**
   - Remove the version comment <!-- bWAPP, version 2.2 --> from the HTML source code.

Conclusion: The level of information exposure on this server is severe. It provides a direct roadmap for a targeted attack. Suppressing the headers alone would significantly increase the effort required for an attacker to proceed.