

Step 1: Asset Identification**

```
admin login
```

Type-of-database

Notes	Description	Asset
Located at <code>/admin</code> – accessible over HTTP	The backend login interface for managing the OpenCart platform	Admin Login Page
Admin Login Page	<code>admin</code> (default)	Admin Username
Needs improvement	No brute-force protection or CAPTCHA detected	Login Protection

Open MySQL and check

```
sudo mysql -u root -p
```

```
SHOW DATABASES;
```

there are:

```
opencart
```

show the that using opencart:

```
SELECT user, host FROM mysql.user;
```

usually

```
opencartuser@localhost
```

Asset	Description	Notes
OpenCart Database	MySQL database holding users, orders, product data	Named <code>opencart</code>
Database User	MySQL user with access to OpenCart DB	<code>opencartuser@localhost</code>
DB Credentials Location	Stored in OpenCart config file	Check <code>/var/www/html/config.php</code> & <code>/var/www/html/admin/config.php</code>

Never expose config.php publicly

```
root@DESKTOP-IT2CGTD:/var/www/html/opencart# cat config.php
<?php
// APPLICATION
define('APPLICATION', 'Catalog');

// HTTP
define('HTTP_SERVER', 'http://localhost/opencart/');

// DIR
define('DIR_OPENCART', '/var/www/html/opencart/');
define('DIR_APPLICATION', DIR_OPENCART . 'catalog/');
define('DIR_EXTENSION', DIR_OPENCART . 'extension/');
define('DIR_IMAGE', DIR_OPENCART . 'image/');
define('DIR_SYSTEM', DIR_OPENCART . 'system/');
define('DIR_STORAGE', DIR_SYSTEM . 'storage/');
define('DIR_LANGUAGE', DIR_APPLICATION . 'language/');
define('DIR_TEMPLATE', DIR_APPLICATION . 'view/template/');
define('DIR_CONFIG', DIR_SYSTEM . 'config/');
define('DIR_CACHE', DIR_STORAGE . 'cache/');
define('DIR_DOWNLOAD', DIR_STORAGE . 'download/');
define('DIR_LOGS', DIR_STORAGE . 'logs/');
define('DIR_SESSION', DIR_STORAGE . 'session/');
define('DIR_UPLOAD', DIR_STORAGE . 'upload/');

// DB
define('DB_DRIVER', 'mysqli');
define('DB_HOSTNAME', 'localhost');
define('DB_USERNAME', 'opencartuser');
define('DB_PASSWORD', 'StrongPassword123');
define('DB_DATABASE', 'opencart');
define('DB_PORT', '3306');
define('DB_PREFIX', 'oc_');
```

```
config-dist.php  config.php  config-test1  index.php  language  model  view
root@DESKTOP-IT2CGTD:/var/www/html/opencart/admin# cat config.php
```

```
<?php
// APPLICATION
define('APPLICATION', 'Admin');

// HTTP
define('HTTP_SERVER', 'http://localhost/opencart/admin/');
define('HTTP_CATALOG', 'http://localhost/opencart/');

// DIR
define('DIR_OPENCART', '/var/www/html/opencart/');
define('DIR_APPLICATION', DIR_OPENCART . 'admin/');
define('DIR_EXTENSION', DIR_OPENCART . 'extension/');
define('DIR_IMAGE', DIR_OPENCART . 'image/');
define('DIR_SYSTEM', DIR_OPENCART . 'system/');
define('DIR_CATALOG', DIR_OPENCART . 'catalog/');
define('DIR_STORAGE', DIR_SYSTEM . 'storage/');
define('DIR_LANGUAGE', DIR_APPLICATION . 'language/');
define('DIR_TEMPLATE', DIR_APPLICATION . 'view/template/');
define('DIR_CONFIG', DIR_SYSTEM . 'config/');
define('DIR_CACHE', DIR_STORAGE . 'cache/');
define('DIR_DOWNLOAD', DIR_STORAGE . 'download/');
define('DIR_LOGS', DIR_STORAGE . 'logs/');
define('DIR_SESSION', DIR_STORAGE . 'session/');
define('DIR_UPLOAD', DIR_STORAGE . 'upload/');

// DB
define('DB_DRIVER', 'mysqli');
define('DB_HOSTNAME', 'localhost');
define('DB_USERNAME', 'opencartuser');
define('DB_PASSWORD', 'StrongPassword123');
define('DB_DATABASE', 'opencart');
define('DB_PORT', '3306');
define('DB_PREFIX', 'oc_');

// OpenCart API
define('OPENCART_SERVER', 'https://www.opencart.com/');
root@DESKTOP-IT2CGTD:/var/www/html/opencart/admin#
```

and the bug called "- > "Sensitive file disclosure – database credentials leaked in confige.php""

Step 2: Identify Threats & Vulnerabilities

Asset	Threat	Description
Customer Database	SQL Injection	Exploit input fields (e.g., login, forms) to extract names, emails, payments.
Admin Panel Login	Brute Force / Credential Stuffing	Repeated login attempts may lead to admin access.
Payment Page	Man-in-the-Middle / Formjacking	Capture card data through injected scripts or weak HTTPS.
Apache Web Server	DoS / Directory Listing / RCE	Attack server to cause downtime or extract files from unsecured directories.
Configuration Files	Information Disclosure / LFI	May expose DB credentials or secrets if accessed via misconfigured paths.
Uploads Folder	File Upload / RCE	Uploading malicious files (e.g., shells) to execute arbitrary code on server.
Session Management System	Session Hijacking	Stealing or predicting session IDs to impersonate users.
Email / Notification System	Phishing / Spoofing	Sending fake or malicious emails to customers if system is compromised.
Product Management & Inventory	Tampering	Changing prices, removing products, or manipulating inventory stock.
Logs and Audit Trails	Log Deletion / Tampering	Erasing or editing logs to hide attack traces or investigation evidence.

Using Wappalyzer

Component	Technology Used	Version
E-commerce Platform	OpenCart	—
Web Server	Apache HTTP Server	2.4.52
Programming Language	PHP	—
Operating System	Ubuntu (Linux)	—
Database	MySQL	—
UI Framework	Bootstrap	—
JavaScript Libraries	jQuery	3.7.1

Component	Technology Used	Version
Time Handling Lib	Moment.js	2.29.1
Icons & Fonts	Font Awesome	—






Some CVE's for Apache HTTP Server(2.4.52)





CVE	Type	Impact	The level
CVE-2022-22720	Request Smuggling	Manipulation of requests; may lead to security policy violations	● Critical (CVSS 9.8) (rapid7.com , cvedetails.com)
CVE-2022-22719	mod_lua buffer overflow	Potential for DOS or execution of unauthorized code	● Critical (CVSS 9.1)
CVE-2022-22721	XML request body overflow	Memory limits bypassed, potentially causing code execution	● Critical (CVSS 9.1)
CVE-2022-23943	mod_sed memory overwrite	Memory attack that can be exploited for DOS or RCE	● Critical (CVSS 9.8)

Recommendations for the Risk Report:

- **Current Risk:** Using Apache 2.4.52 exposes the website to severe attacks such as HTTP Request Smuggling and Buffer Overflows.
- **Impact:** High likelihood of security breaches including Remote Code Execution (RCE) or Denial of Service (DoS) attacks.
- **Recommendation:**
 - **Immediately upgrade** to the latest stable version (2.4.60 or higher).
 - Alternatively, apply security patches for the relevant CVEs if upgrading is not possible.
 - Review and **harden configurations** for sensitive modules like `mod_lua` and `mod_sed` to minimize attack surface.

Step 3: Risk Assessment

 Asset	 Threat	 Likelihood	 Impact	 Risk Level
Customer Database	SQL Injection	High	High	Critical

 Asset	 Threat	 Likelihood	 Impact	 Risk Level
Admin Panel Login	Brute Force / Credential Stuffing	Medium	High	High
Payment Page	Man-in-the-Middle / Formjacking	Medium	High	High
Apache Web Server	Remote Code Execution / Directory Access	Medium	High	High
Configuration Files	Info Disclosure / LFI	Medium	Critical	Critical
Uploads Folder	File Upload / RCE	Medium	High	High
Session Management System	Session Hijacking	Medium	Medium	Medium
Email / Notification System	Phishing / Email Spoofing	Low	Medium	Low
Product & Inventory System	Tampering / Unauthorized Modification	Low	Medium	Low
Logs and Audit Trails	Log Deletion / Tampering	Medium	High	High

Step 4: Compliance Mapping

 Threat	Required Control	Framework
SQL Injection	Input Validation & Sanitization	PCI-DSS
Unencrypted Data	Data-at-Rest Encryption	GDPR
Unauthorized Access	Role-Based Access Control (RBAC)	ISO 27001
Session Hijacking	Secure Session Management	OWASP ASVS
Default Credentials	Credential Hardening	NIST SP 800-53
Config File Exposure	Restrict File Permissions	OWASP Top 10
File Upload Exploits	File Type & Size Validation	OWASP ASVS
Email Spoofing	SPF, DKIM, DMARC Implementation	ISO 27001
Lack of Audit Trails	Logging & Monitoring	PCI-DSS

Web Vulnerability Assessment (Nikto)

Nikto was used to scan the OpenCart installation on `http://127.0.0.1/opencart`. The following findings were identified:

- Several cookies (OCSESSID , currency) are missing the HttpOnly attribute.
- No X-Frame-Options or X-Content-Type-Options headers were detected.
- Apache version is outdated (2.4.52), exposing the server to known CVEs.
- Configuration files such as config.php and admin/config.php are publicly accessible.
- Directory listing enabled on sensitive folders (system/ , image/).
- Multiple PHP backdoor file managers were found in WordPress-like paths and others.
- Potential command execution paths (/opencart/shell , login.cgi?cli=...) detected.

Recommendations

- Harden HTTP headers (X-Frame-Options , X-Content-Type-Options , CORS policy).
- Upgrade Apache to the latest secure version (2.4.60+).
- Restrict access to config files and sensitive directories.
- Delete any suspicious or unauthorized PHP files (backdoors).
- Review and sanitize all admin file paths.

1 2 3 4	Risk / Misconfiguration	Description	Risk Level	Recommendation
1	OCSESSID and currency cookies lack HttpOnly flag	Could be accessed via JavaScript and exploited via XSS	Medium	Add the HttpOnly flag to cookies to prevent JavaScript access
2	Access-Control-Allow-Origin: *	Permits requests from any domain → Cross-Origin Risk	Medium	Restrict to trusted domains instead of using *
3	Missing X-Frame-Options header	Makes the site vulnerable to Clickjacking attacks	Medium	Add X-Frame-Options: SAMEORIGIN or DENY header
4	X-Content-Type-Options header not set	May allow MIME-type sniffing by browsers	Low	Add X-Content-Type-Options: nosniff to prevent incorrect content rendering
5	robots.txt contains 14 disallowed paths	May disclose sensitive or interesting directories	Info	Review robots.txt for potentially exposed sensitive paths
6	Apache version 2.4.52 is outdated	Known to contain multiple CVEs and	High	Upgrade to Apache version 2.4.60 or later

1 2 3 4	Risk / Misconfiguration	Description	Risk Level	Recommendation
		security flaws		
7	config.php and admin/config.php are publicly accessible	May expose database credentials and configurations	High	Secure file permissions and block access via web server
8	Directory Indexing enabled on /system/ and /image/	Allows attackers to view file listings and download files	Medium	Disable directory indexing via .htaccess or Apache config
9	PHP backdoor file managers found in multiple paths	Severe security threat—indicates potential prior compromise	Critical	Remove all suspicious files immediately and investigate system integrity
10	Suspicious files like shell, login.cgi, server.php	Allow command execution such as cat /etc/hosts	Critical	Check for Command Injection vulnerabilities and delete any backdoors

robots.txt Review:

The robots.txt file contains several Disallow rules preventing search engines from indexing dynamic filter and pagination URLs. This is a best practice for SEO, not a security risk.

However, always review robots.txt to ensure it doesn't unintentionally disclose sensitive admin or config paths.

No sensitive paths (e.g., /admin/) were found in this file.

Disallow: /admin/