Web Application Security

Dr. Digvijaysinh Rathod
Professor
School of Cyber Security and Digital Forensics
National Forensic Sciences University

digvijay.rathod@nfsu.ac.in

Content Management System

(CMS) Security

Dr. Digvijaysinh Rathod

Introduction to CMS Security

- ✓ A Content Management System (CMS) is a platform that allows users to create, manage, and publish website content without needing deep technical knowledge. **Examples include:**

WordPress (most widely used)

Joomla

Drupal

Magento (E-commerce CMS)

Ghost, Shopify, Wix (hosted platforms)

Introduction to CMS Security

- ✓ CMS platforms are frequent targets for attackers because:
  - They are widely used
  - Many sites run outdated versions
  - Third-party plugins and themes introduce risks
  - Users often misconfigure security settings

a) Popularity = Attack Surface

Attackers release automated bots that continuously scan for:

- WordPress login pages
- Joomla admin panels
- Outdated plugin versions

b) Plugin Ecosystem

- Most CMS websites use dozens of plugins.
- Any vulnerable plugin → full website compromise.

c) Weak Admin Authentication

Defaults like:

username: admin

password: admin123

d) Incorrect Permissions

Writable directories allow attackers to upload shells or malicious PHP scripts.

Common CMS Components

1. Core CMS files
2. Themes/Templates
3. Plugins/Extensions/Modules
4. Database (MySQL / PostgreSQL)
5. Media uploads directory
6. Admin dashboard

Every component introduces potential vulnerabilities.

Common CMS Vulnerabilities

1. Weak Authentication

- Default credentials
- No rate limiting
- No MFA/2FA
- Weak password policies

2. Vulnerable Plugins & Themes

- SQL Injection
- XSS Remote Code Execution (RCE)
- File Upload vulnerabilities

3. Outdated Core Files

Older versions contain known CVEs.

Attackers automate scanning for:

/wp-includes/version.php

/administrator/manifests/...

/CHANGELOG.txt

4. Misconfigured File Permissions

Example in WordPress:

wp-content/uploads/ is writable

3.Attackers can upload:

- Backdoor shells
- Malware
- Defacement pages

5. Unvalidated File Upload

Malicious PHP uploads → full server compromise.

6. Database Exposures

- Public backups (backup.sql, db.zip)
- phpMyAdmin exposed
- DB credentials in config files

7. Cross-Site Scripting (XSS)

- Through:
- Comment boxes
- Search boxes
- Plugin forms

8. PHP Object Injection (POI)

- Especially in older CMS systems.

A. Enumeration

- Attackers identify:
- CMS type
- Version
- Installed plugins
- Admin URL

Tools:

- WPScan (WordPress)
- Droopescan (Drupal/Joomla)
- WhatWeb
- Wappalyzer

B. Bruteforce Login

- Targeting /wp-login.php or /administrator/.

C. Plugin Exploitation

- Using public exploits:
- Metasploit modules
- GitHub PoCs
- CVE database

D. Uploading Web Shells

If file upload is unrestricted:

- shell.php
- cmd.php

E. SQL Injection

- Extracting admin credentials from database.

F. Privilege Escalation

- Exploiting weak role configurations.

CMS Hardening Techniques

1. Keep Everything Updated
- CMS core updates
- Plugin and theme updates
- Remove unused plugins

2. Strong Authentication
- Enforce MFA
- Disable default users like "admin"
- Limit login attempts
- Use CAPTCHA

3. Secure File Permissions

Recommended WordPress example:

```
wp-config.php → 400
.htaccess → 444
wp-content/uploads → 755 (not 777!)
```

4. Disable Unnecessary Features
- XML-RPC
- Directory listing
- File editing from dashboard

5. Web Application Firewall (WAF)

Tools:
- Cloudflare
- ModSecurity (OWASP Core Rule Set)

6. Input Validation & Output Encoding
- Prevent XSS and Injection.

7. Database Security
- Strong DB passwords
- Restrict DB user privileges
- Use table prefix changes (e.g., `wp9a_` instead of `wp_`)

8. Backup and Monitoring
- Automatic daily backups
- Integrity checks
- File change monitoring tools

Mobile Phone Security

Dr. Digvijaysinh Rathod
Professor
School of Cyber Security and Digital Forensics
National Forensic Sciences University

digvijay.rathod@nfsu.ac.in