VULNERABILITY ASSESSMENT AND PENETRATION TESTING (VAPT) REPORT

Executive Summary

This assessment was conducted on an authorized TryHackMe target to identify security weaknesses using open-source tools. The evaluation identified multiple low- and high-risk vulnerabilities, including a Remote Code Execution (RCE) issue in Concrete5 CMS. The vulnerability allows an attacker to execute system commands on the server, resulting in full compromise.

Scope and Authorization

- **Target Type:** Authorized TryHackMe vulnerable machine
- **Target IP:** 10.49.187.138
- **Testing Type:** Black-box assessment
- **Access Method:** OpenVPN
- **Authorization:** Explicitly permitted for educational purposes under TryHackMe

All activities were conducted within a controlled lab environment.

Planning and Documentation

Planning was performed prior to execution to define scope boundaries, permitted tools, and testing methodology. The assessment followed industry-recognized practices such as NIST SP 800-115 and the OWASP Web Security Testing Guide. Documentation of findings was maintained in a structured format similar to professional reporting platforms such as Dradis CE, enabling systematic tracking of vulnerabilities, evidence, and remediation details throughout the engagement.

Environment Setup

Attacker System

- Operating System: Kali Linux
- Network Access: OpenVPN (TryHackMe VPN)

Target System

- Platform: Linux-based server
- Application: Concrete5 CMS
- Web Service: Apache HTTP Server

Tools Used

Tool	Purpose
Nmap	Network discovery and service enumeration
Nikto	Web server vulnerability and misconfiguration scanning
Nuclei	Template-based vulnerability detection
Metasploit Framework	Validation of known vulnerabilities (exploitability check)
Netcat	Listener for command execution validation
Web Browser	Manual interaction and verification
OpenVPN	Secure access to authorized lab environment

Methodology Followed

The assessment followed a standard VAPT lifecycle:

1. Reconnaissance
2. Service Enumeration
3. Vulnerability Scanning
4. Manual Validation
5. Risk Classification
6. Documentation and Reporting

Reconnaissance and Enumeration

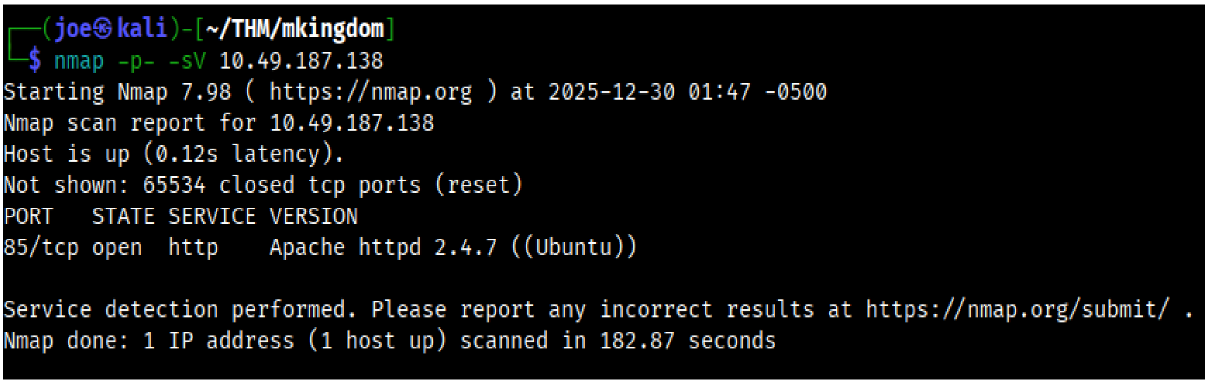
Nmap Scan

Command used:

```
nmap -sV -p- <target-ip>
```

Observations:

- Host was reachable
- Multiple open ports detected
- Web server identified as Apache
- Additional exposed services observed



```
┌──(joe㉿kali)-[~/THM/mkingdom]
└─$ nmap -p- -sV 10.49.187.138
Starting Nmap 7.98 ( https://nmap.org ) at 2025-12-30 01:47 -0500
Nmap scan report for 10.49.187.138
Host is up (0.12s latency).
Not shown: 65534 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
85/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 182.87 seconds
```

Figure 1: Nmap output showing open ports and service versions

Web Vulnerability Assessment (Nikto)

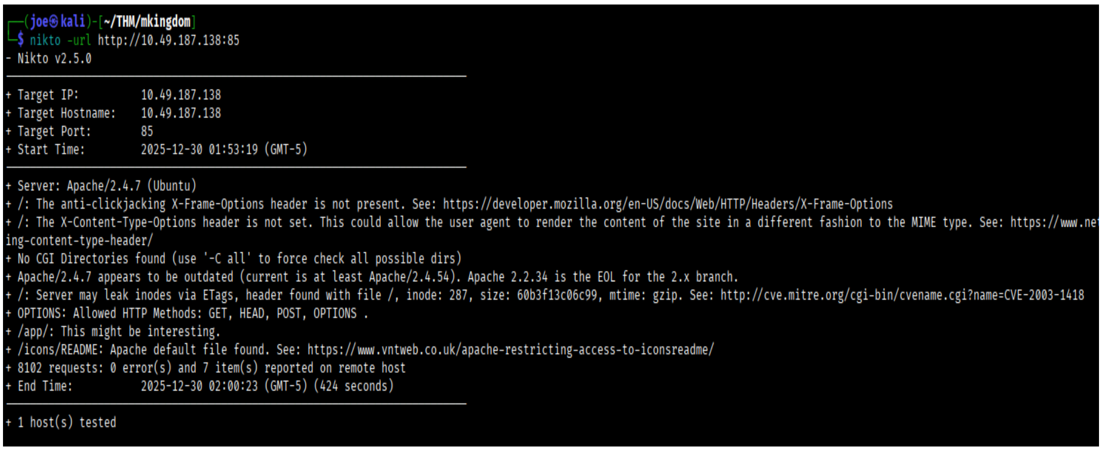
Nikto was used to identify common web server misconfigurations and insecure settings.

Command used:

nikto -h http://<target-ip>

Findings:

- Missing X-Frame-Options header
- Missing X-Content-Type-Options header
- Apache version disclosure
- Allowed HTTP methods (GET, POST, OPTIONS, HEAD)
- Potential information leakage through headers



```
┌──(joe㉿kali)-[~/THM/mkingdom]
└─$ nikto -url http://10.49.187.138:85
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          10.49.187.138
+ Target Hostname:    10.49.187.138
+ Target Port:        85
+ Start Time:         2025-12-30 01:53:19 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.7 (Ubuntu)
+ /: 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: https://www.ne
ing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ 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.
+ /: Server may leak inodes via ETags, header found with file /, inode: 287, size: 60b3f13c06c99, mtime: gzip. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ OPTIONS: Allowed HTTP Methods: GET, HEAD, POST, OPTIONS .
+ /app/: This might be interesting.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ 8102 requests: 0 error(s) and 7 item(s) reported on remote host
+ End Time:           2025-12-30 02:00:23 (GMT-5) (424 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Figure 2: Nikto scan results showing missing security headers

Automated Vulnerability Scanning (Nuclei)

Nuclei was executed to detect known CVEs and misconfigurations using updated templates.

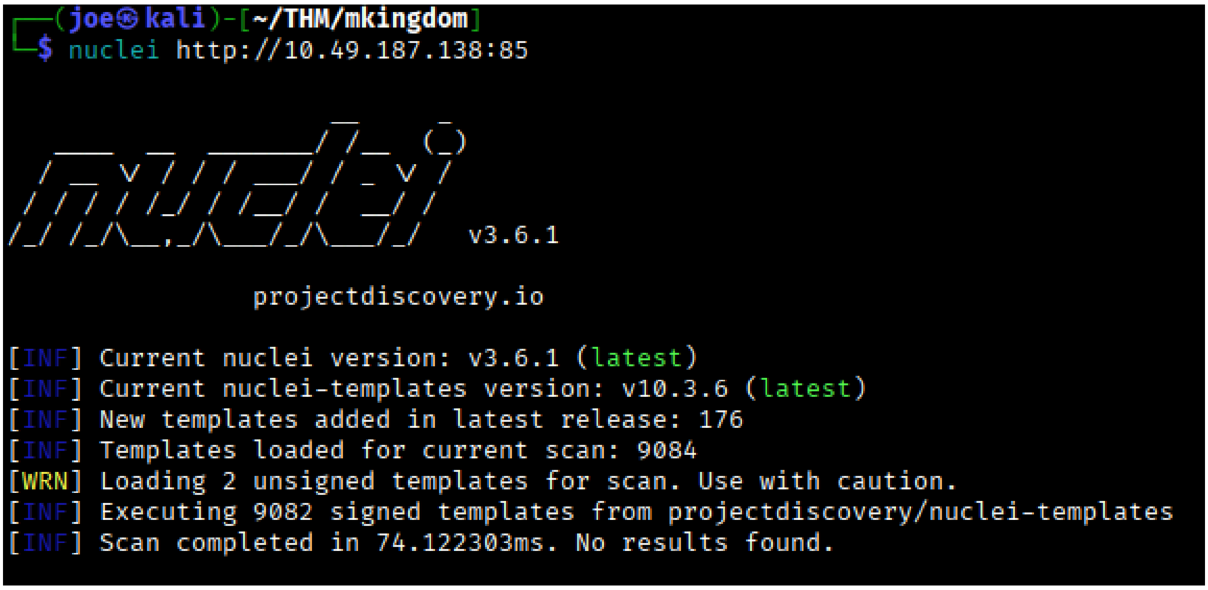
Command used:

nuclei -u http://<target-ip>

Result:

No matching vulnerabilities were detected.

This indicates that the target does not expose known issues detectable through standard Nuclei templates.



```
┌──(joe㉿kali)-[~/THM/mkingdom]
└─$ nuclei http://10.49.187.138:85

                     __     _
   ____  __  _______/ /__  (_)
  / __ \/ / / / ___/ / _ \/ /
 / / / / /_/ / /__/ /  __/ /
/_/ /_/\__,_/\___/_/\___/_/   v3.6.1

                projectdiscovery.io

[INF] Current nuclei version: v3.6.1 (latest)
[INF] Current nuclei-templates version: v10.3.6 (latest)
[INF] New templates added in latest release: 176
[INF] Templates loaded for current scan: 9084
[WRN] Loading 2 unsigned templates for scan. Use with caution.
[INF] Executing 9082 signed templates from projectdiscovery/nuclei-templates
[INF] Scan completed in 74.122303ms. No results found.
```

Figure 3: Nuclei output showing no findings

Vulnerability Validation Using Metasploit

The Metasploit Framework was used to verify whether any publicly available exploit modules existed for the identified services and application version.

No applicable CVE-based exploit modules were found for the target. This indicates that the identified Remote Code Execution issue is **not associated with a publicly assigned CVE**, and therefore cannot be exploited using automated Metasploit modules.

The vulnerability observed during testing resulted from **application-level misconfiguration**, specifically improper file upload handling and weak access controls. Such vulnerabilities are typically identified through manual testing and configuration review rather than automated exploitation tools.

This demonstrates the importance of combining automated scanners with manual validation techniques during a Vulnerability Assessment and Penetration Testing (VAPT) engagement.

High-Severity Finding: Application Misconfiguration Leading to Remote Code Execution (Concrete5 CMS)

Vulnerability Name

Remote Code Execution via File Upload Misconfiguration (Concrete5 CMS)

Affected Component

Concrete5 CMS – File Manager functionality

Severity

High

Description

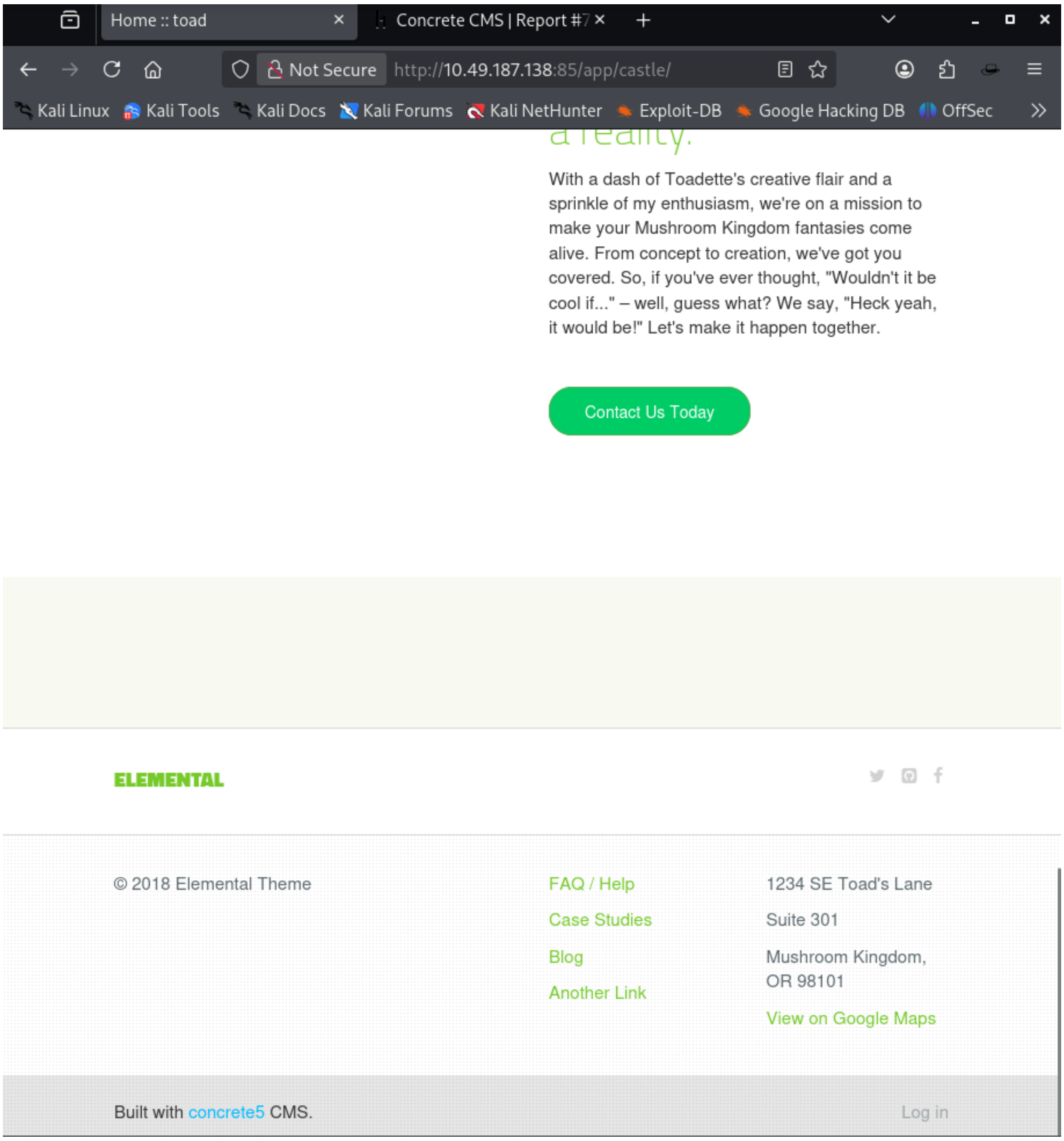
The target web application was found to have insecure configuration settings within the Concrete5 CMS administrative interface. Weak credentials allowed administrative access, after which file upload restrictions could be modified.

This misconfiguration allowed executable file types (such as PHP) to be uploaded and accessed through the web server, resulting in remote command execution.

This issue is classified as a configuration-based vulnerability rather than a software flaw with an assigned CVE. Such weaknesses are commonly identified through manual testing rather than automated vulnerability scanners.

Evidence of Successful Command Execution

The following evidence demonstrates successful command execution on the target system:



a reality.

With a dash of Toadette's creative flair and a sprinkle of my enthusiasm, we're on a mission to make your Mushroom Kingdom fantasies come alive. From concept to creation, we've got you covered. So, if you've ever thought, "Wouldn't it be cool if..." – well, guess what? We say, "Heck yeah, it would be!" Let's make it happen together.

Contact Us Today

ELEMENTAL

© 2018 Elemental Theme

FAQ / Help
Case Studies
Blog
Another Link

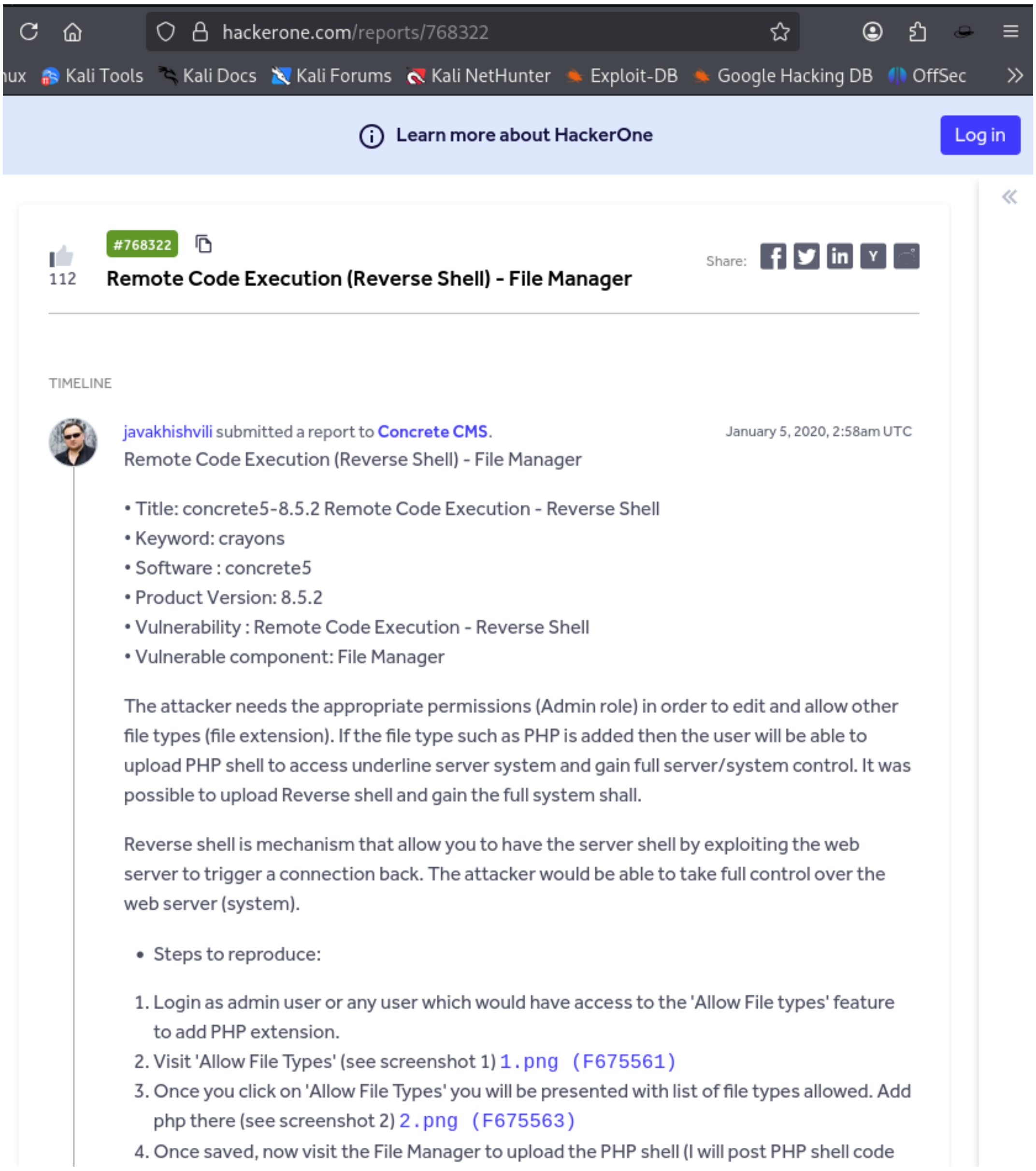
1234 SE Toad's Lane
Suite 301
Mushroom Kingdom,
OR 98101
View on Google Maps

Built with concrete5 CMS.

Log in

Figure 4: Web application homepage revealing Concrete5 CMS usage, indicating version disclosure through publicly accessible page content.

The web interface reveals that the target application is running the Concrete5 content management system. Identifying the CMS helps in understanding the application structure and possible configuration weaknesses.



hackerone.com/reports/768322
ux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Learn more about HackerOne
Log in
#768322
112
Share:

Remote Code Execution (Reverse Shell) - File Manager

TIMELINE

javakhishvili submitted a report to Concrete CMS. January 5, 2020, 2:58am UTC

Remote Code Execution (Reverse Shell) - File Manager

- Title: concrete5-8.5.2 Remote Code Execution - Reverse Shell
- Keyword: crayons
- Software : concrete5
- Product Version: 8.5.2
- Vulnerability : Remote Code Execution - Reverse Shell
- Vulnerable component: File Manager

The attacker needs the appropriate permissions (Admin role) in order to edit and allow other file types (file extension). If the file type such as PHP is added then the user will be able to upload PHP shell to access underline server system and gain full server/system control. It was possible to upload Reverse shell and gain the full system shall.

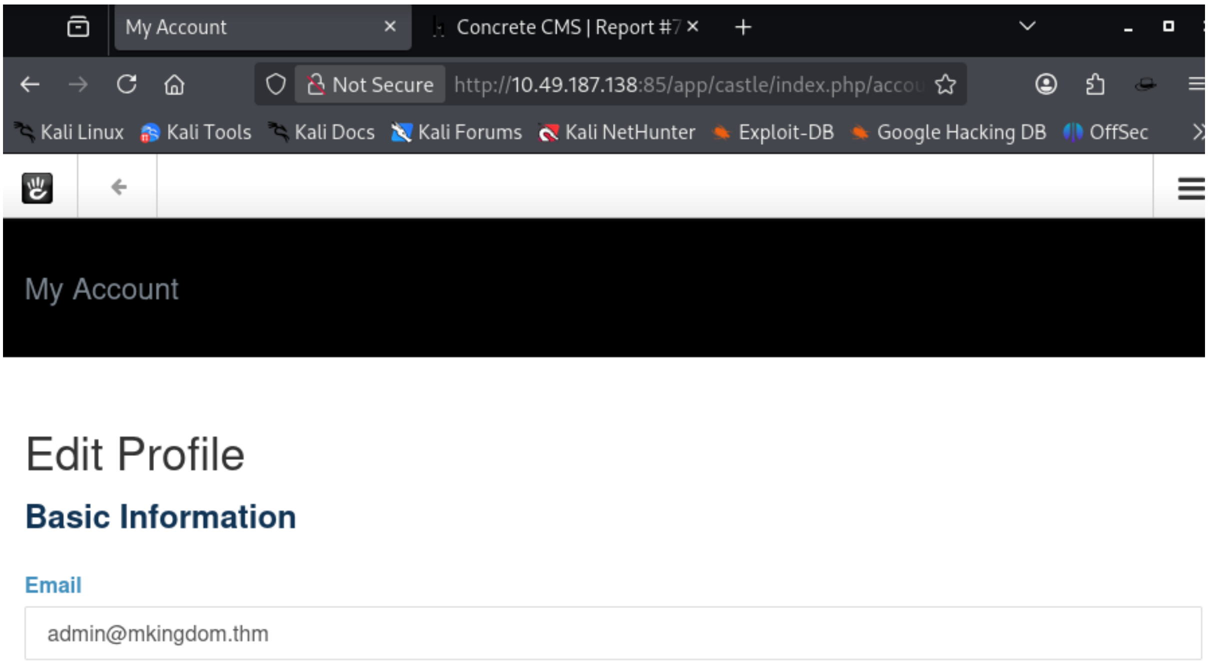
Reverse shell is mechanism that allow you to have the server shell by exploiting the web server to trigger a connection back. The attacker would be able to take full control over the web server (system).

- Steps to reproduce:

1. Login as admin user or any user which would have access to the 'Allow File types' feature to add PHP extension.
2. Visit 'Allow File Types' (see screenshot 1) `1.png (F675561)`
3. Once you click on 'Allow File Types' you will be presented with list of file types allowed. Add php there (see screenshot 2) `2.png (F675563)`
4. Once saved, now visit the File Manager to upload the PHP shell (I will post PHP shell code

Figure 5: Publicly available vulnerability disclosure describing a Remote Code Execution scenario in Concrete CMS due to improper file handling and configuration weaknesses.

This public reference was reviewed to understand how insecure file upload configurations in Concrete CMS may lead to remote code execution. The reference was used only for understanding the attack pattern and not as a direct exploit source.



http://10.49.187.138:85/app/castle/index.php/accou

My Account

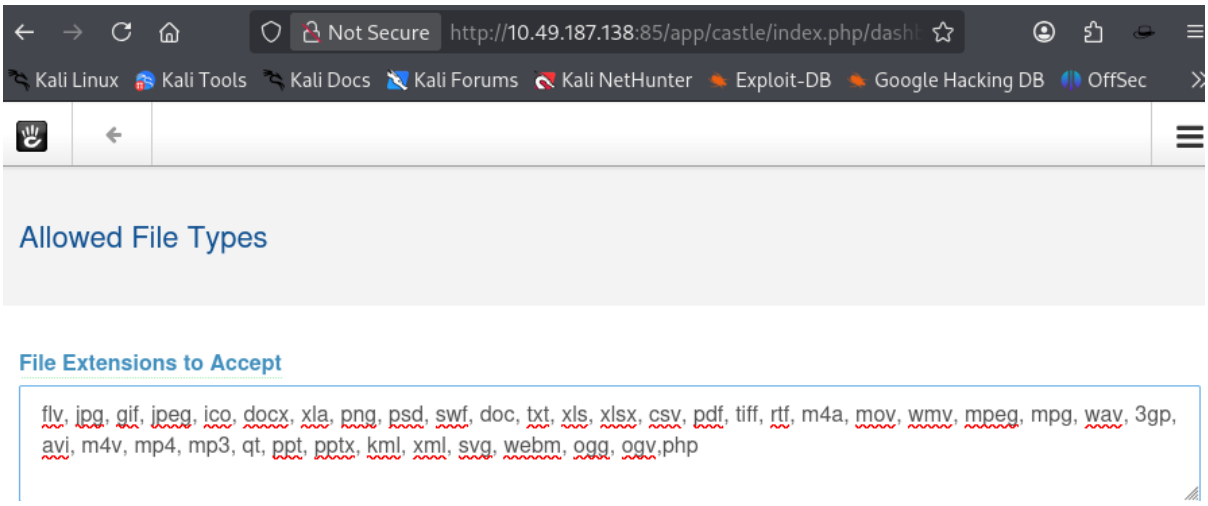
Edit Profile

Basic Information

Email: admin@mkingdom.thm

Figure 6: Administrative interface accessed using weak/default credentials, demonstrating improper authentication controls.

Successful access to the administrative dashboard using weak credentials allows unauthorized users to modify application configurations, increasing the risk of further exploitation.



http://10.49.187.138:85/app/castle/index.php/dash

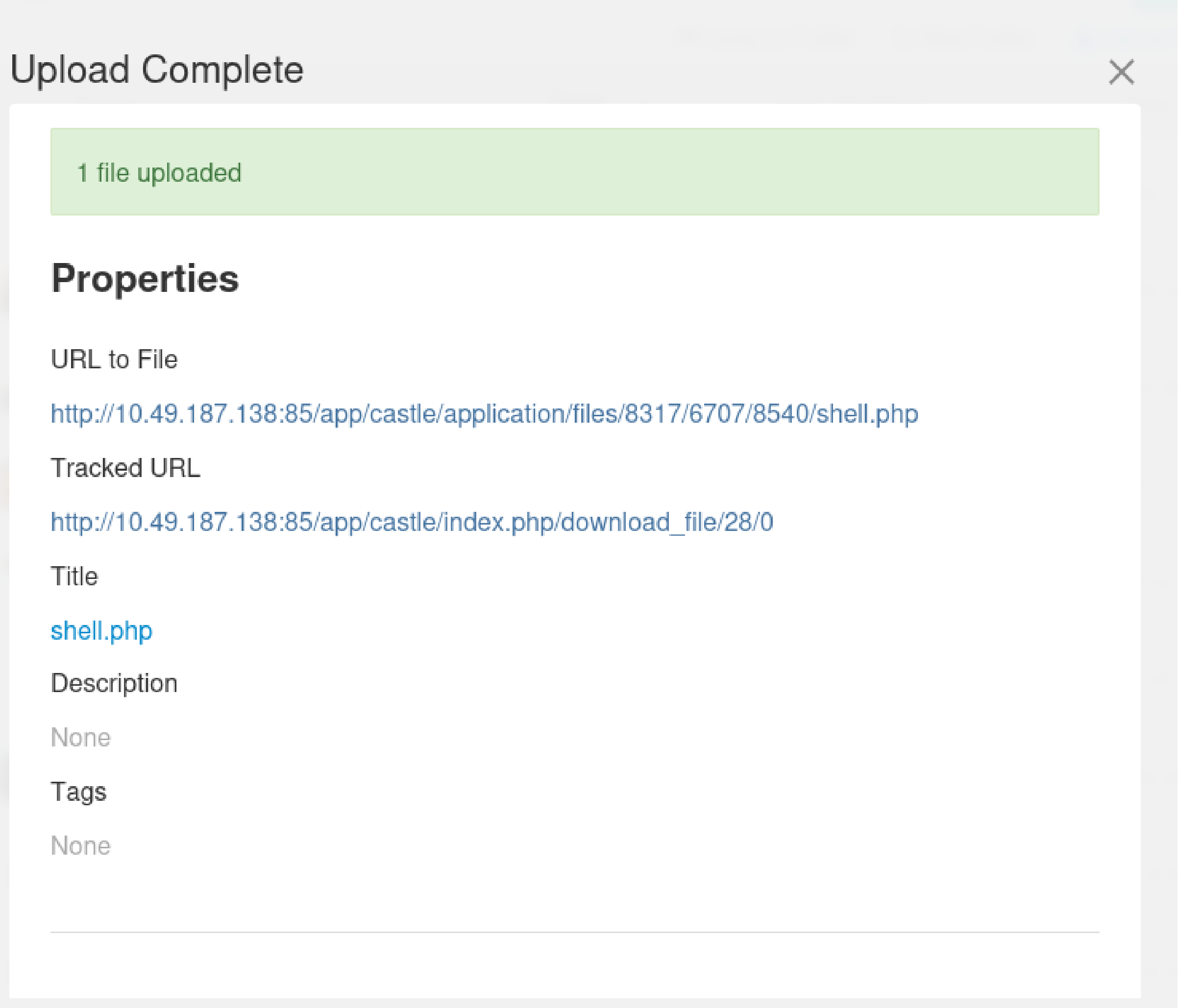
Allowed File Types

File Extensions to Accept

flv, jpg, gif, jpeg, ico, docx, xla, png, psd, swf, doc, txt, xls, xlsx, csv, pdf, tiff, rtf, m4a, mov, wmv, mpeg, mpg, wav, 3gp, avi, m4v, mp4, mp3, qt, ppt, pptx, kml, xml, svg, webm, ogg, ogv,php

Figure 7: File type configuration panel showing PHP extension enabled, allowing executable file uploads.

The file upload configuration allows executable file extensions such as PHP. Allowing such extensions can lead to remote code execution if uploaded files are processed by the server.



Upload Complete

1 file uploaded

Properties

URL to File

http://10.49.187.138:85/app/castle/application/files/8317/6707/8540/shell.php

Tracked URL

http://10.49.187.138:85/app/castle/index.php/download_file/28/0

Title

shell.php

Description

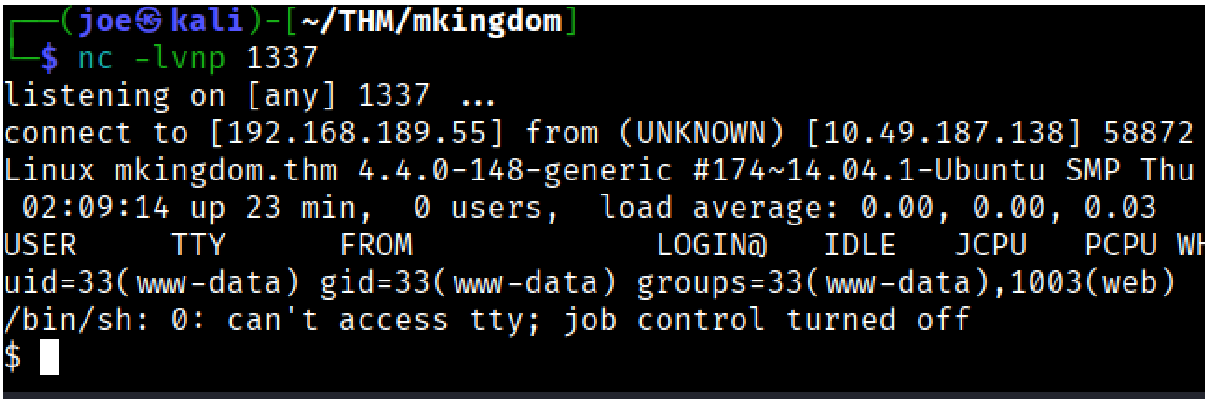
None

Tags

None

Figure 8: PHP file successfully uploaded to the server through the file manager functionality.

The uploaded file was stored in a web-accessible directory, confirming insufficient validation of uploaded content and improper restriction of executable files.



```
┌──(joe㉿kali)-[~/THM/mkingdom]
└─$ nc -lvnp 1337
listening on [any] 1337 ...
connect to [192.168.189.55] from (UNKNOWN) [10.49.187.138] 58872
Linux mkingdom.thm 4.4.0-148-generic #174~14.04.1-Ubuntu SMP Thu
 02:09:14 up 23 min,  0 users,  load average: 0.00, 0.00, 0.03
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WH
uid=33(www-data) gid=33(www-data) groups=33(www-data),1003(web)
/bin/sh: 0: can't access tty; job control turned off
$ 
```

Figure 9: Reverse shell connection received on the attacker system after accessing the uploaded file.

Accessing the uploaded PHP file triggered server-side execution, resulting in a reverse shell connection. This confirms successful command execution and demonstrates the impact of the vulnerability under controlled conditions

Risk Assessment (CVSS-Based)

Since no official CVE exists for this issue, the CVSS score was calculated manually based on observed exploitability and impact using CVSS v3.1 metrics.

Metric	Value
Attack Vector	Network
Attack Complexity	Low
Privileges Required	Low
User Interaction	None
Confidentiality Impact	High
Integrity Impact	High

Estimated CVSS Score: 8.8 (High)

The CVSS score is an estimated value based on observed behavior, as no official CVE exists for this issue.

Summary of Findings

ID	Vulnerability	Severity	Tool Used	Status
V-01	Apache version disclosure	Low	Nmap / Nikto	Identified
V-02	Missing security headers	Low	Nikto	Identified
V-03	Open HTTP methods	Low	Nikto	Identified
V-04	No public CVE available; manual validation performed	Informational	Metasploit	Verified (no applicable exploit found)
V-05	Concrete5 Remote Code Execution (file upload misconfiguration)	High	Manual / Netcat	Confirmed

Recommendations

1. Upgrade Concrete5 CMS to the latest secure version
2. Restrict executable file uploads
3. Disable execution permissions in upload directories
4. Enforce strict file-type validation
5. Add HTTP security headers
6. Harden Apache configuration
7. Limit exposed services
8. Perform periodic vulnerability scans

Conclusion

This assessment successfully identified multiple configuration weaknesses and a high-severity remote code execution vulnerability within the target system. While automated tools such as Nmap, Nikto, and Nuclei identified basic issues, the most critical finding was discovered through manual testing.

The ability to upload and execute server-side scripts due to improper access control and file validation demonstrates the importance of secure configuration practices. This assessment highlights the necessity of combining automated scanning with manual analysis to effectively identify real-world security risks. Implementing the recommended mitigations will significantly improve the overall security posture of the application.

References

- OWASP Top 10
- NIST SP 800-115
- Exploit-DB
- HackerOne Public Reports
- Nuclei Documentation
- Nikto Documentation
- TryHackMe Platform