VULNERABILITY ASSESSMENT AND PENETRATION TESTING (VAPT) REPORT

1. Executive Summary

This report documents a security assessment conducted on a vulnerable target system using open-source tools and a structured VAPT methodology. The objective of the assessment was to identify security weaknesses, evaluate their risk, and provide remediation recommendations without the use of paid tools.

During the assessment, multiple exposed services were identified, and a critical web application vulnerability related to insecure file upload functionality was discovered. Although basic security controls were present, the application relied on weak validation mechanisms that could potentially be bypassed, leading to serious security risks such as remote code execution.

2. Scope and Objectives

2.1 Scope

- Target System: TryHackMe vulnerable machine
- Network Scope: Single target IP address
- Services Tested:
  - HTTP (Port 80)
  - SSH (Port 22)

2.2 Objectives

- Identify exposed services and applications
- Enumerate web application directories
- Detect and validate security vulnerabilities
- Assess risk using CVSS and a risk matrix
- Provide remediation recommendations

3. Methodology

The assessment followed a structured **Vulnerability Assessment and Penetration Testing (VAPT)** methodology aligned with OWASP and NIST guidelines.

Phases Followed

1. Planning & Reconnaissance
2. Discovery & Enumeration
3. Vulnerability Identification
4. Validation
5. Risk Assessment
6. Reporting

4. Tools Used

Tool	Purpose
Kali Linux	Attack platform
Nmap	Network & service enumeration
Gobuster	Web directory enumeration
Metasploit Framework	Exploit research
Web Browser	Manual validation
CVSS Calculator	Risk scoring

5. Reconnaissance and Enumeration

5.1 Network Scanning (Nmap)

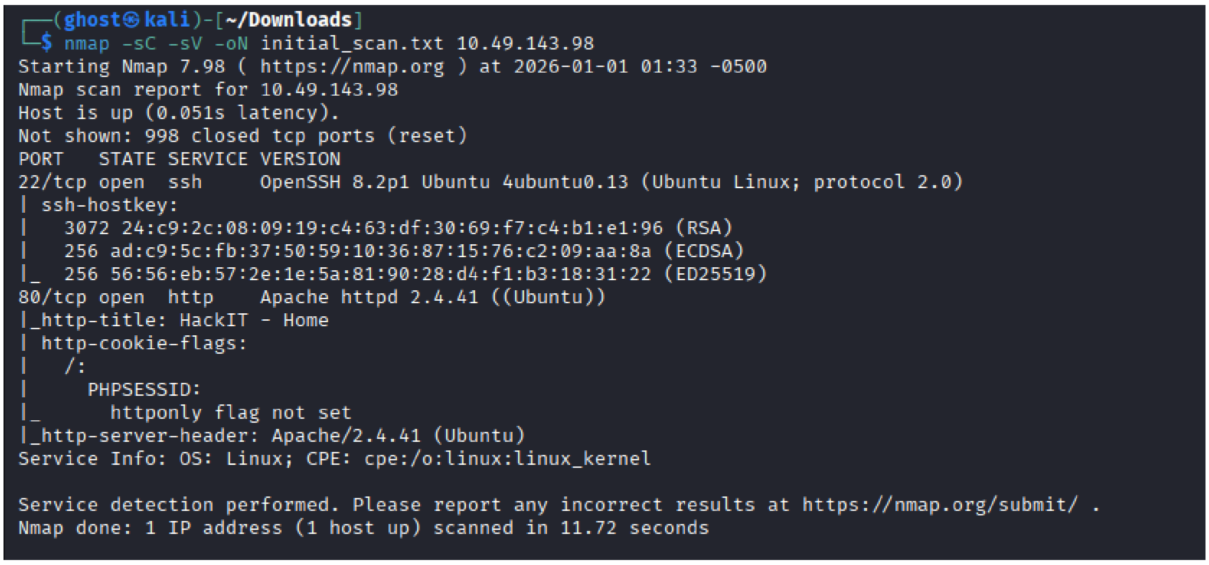
Nmap was used to identify open ports and running services on the target system.

Command Used:

```
nmap -sC -sV -oN initial_scan.txt <TARGET_IP>
```

Results Identified:

- Port 22/tcp – OpenSSH 8.2p1 (Ubuntu)
- Port 80/tcp – Apache HTTP Server 2.4.41 (Ubuntu)



```
┌──(ghost㉿kali)-[~/Downloads]
└─$ nmap -sC -sV -oN initial_scan.txt 10.49.143.98
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-01 01:33 -0500
Nmap scan report for 10.49.143.98
Host is up (0.051s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 24:c9:2c:08:09:19:c4:63:df:30:69:f7:c4:b1:e1:96 (RSA)
|   256 ad:c9:5c:fb:37:50:59:10:36:87:15:76:c2:09:aa:8a (ECDSA)
|_  256 56:56:eb:57:2e:1e:5a:81:90:28:d4:f1:b3:18:31:22 (ED25519)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-title: HackIT - Home
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
|_http-server-header: Apache/2.4.41 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.72 seconds
```

Figure 1: Nmap scan results showing open ports and service versions.

5.2 Web Directory Enumeration (Gobuster)

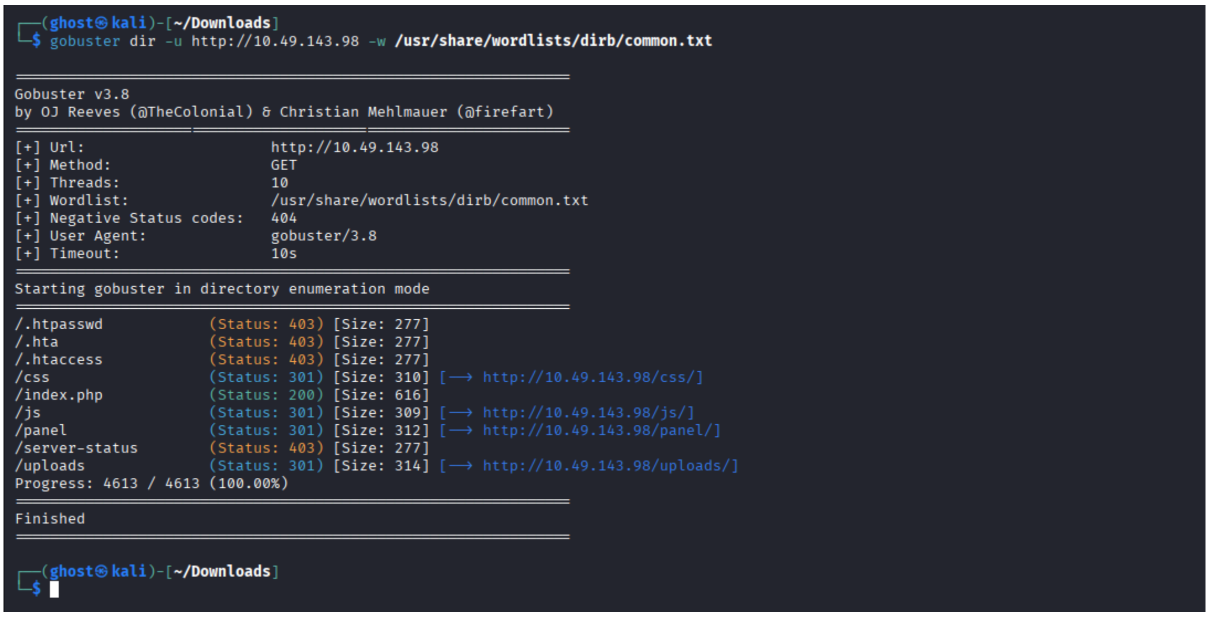
Gobuster was used to enumerate hidden directories on the web server.

Command Used:

gobuster dir -u 10.49.143.98 -w /usr/share/wordlists/dirb/common.txt

Discovered Directories:

- /panel
- /uploads
- /index.php



```
┌──(ghost㉿kali)-[~/Downloads]
└─$ gobuster dir -u http://10.49.143.98 -w /usr/share/wordlists/dirb/common.txt

===============================================================
Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.49.143.98
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.8
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.htpasswd            (Status: 403) [Size: 277]
/.hta                 (Status: 403) [Size: 277]
/.htaccess            (Status: 403) [Size: 277]
/css                  (Status: 301) [Size: 310] [--> http://10.49.143.98/css/]
/index.php            (Status: 200) [Size: 616]
/js                   (Status: 301) [Size: 309] [--> http://10.49.143.98/js/]
/panel                (Status: 301) [Size: 312] [--> http://10.49.143.98/panel/]
/server-status        (Status: 403) [Size: 277]
/uploads              (Status: 301) [Size: 314] [--> http://10.49.143.98/uploads/]
Progress: 4613 / 4613 (100.00%)
===============================================================
Finished
===============================================================

┌──(ghost㉿kali)-[~/Downloads]
└─$ 
```

Figure 2: Gobuster output showing discovered web directories.

6. Vulnerability Identification

6.1 Identified Vulnerability

Vulnerability Name: Insecure File Upload
CWE: CWE-434 – Unrestricted File Upload
Affected Endpoint: /panel

The /panel directory contained a file upload feature that allowed users to upload files to the server. Uploaded files were stored in a publicly accessible directory (/uploads).

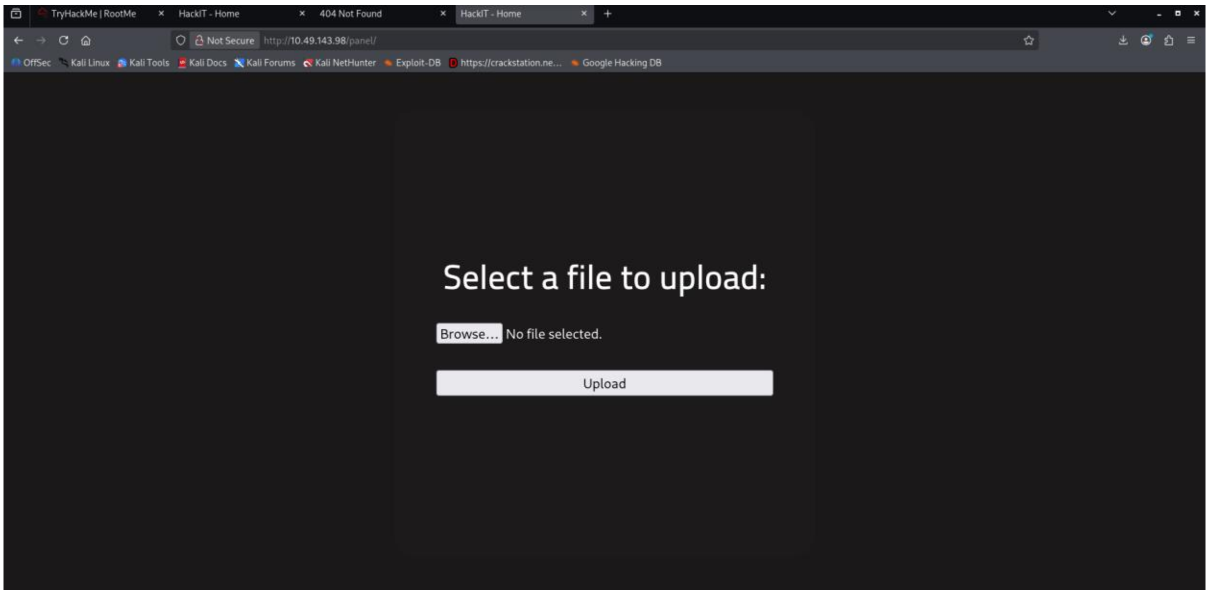
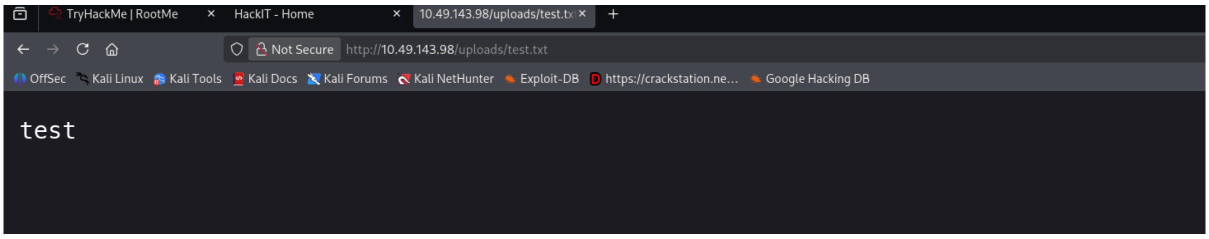


Figure 3: File upload interface discovered at /panel.

7. Vulnerability Validation

7.1 File Upload Testing

The uploaded file was accessible directly via the browser.



```
test
```

Figure 4: Accessing uploaded file from /uploads/test.txt.

7.2 Security Control Testing

A PHP file upload test was conducted to assess server-side validation.

- Uploading .php files was blocked
- Error message indicated that PHP files are not permitted

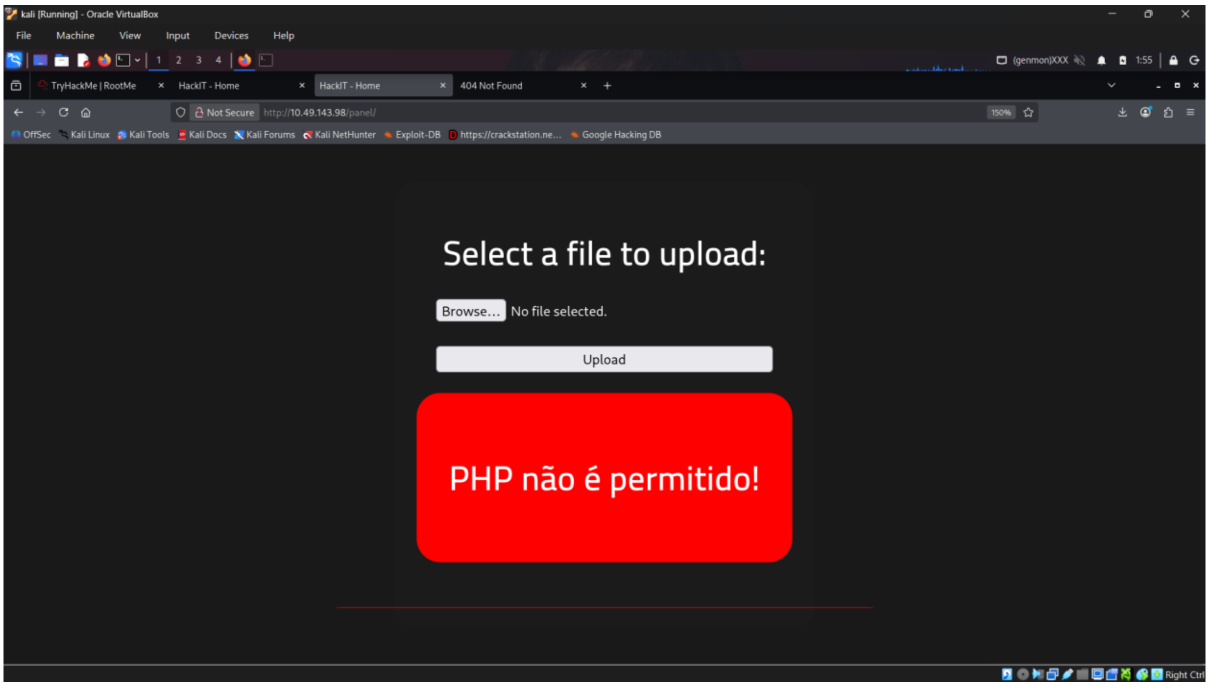


Figure 5: PHP file upload blocked by application validation.

This confirms the presence of **extension-based filtering**, which is a weak security control that can potentially be bypassed.

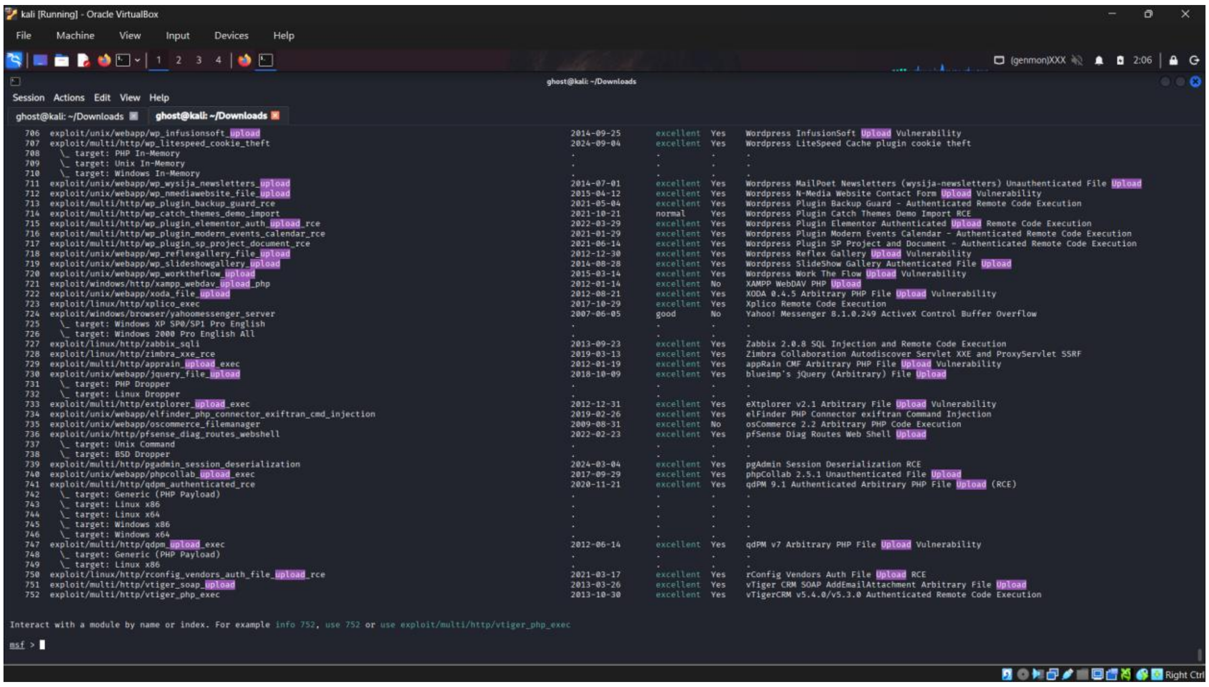
8. Metasploit Framework Analysis

The Metasploit Framework was used to search for known exploit modules related to file upload vulnerabilities.

Search Command Used:

```
search type:exploit upload
```

The search returned multiple exploit modules related to specific applications (e.g., WordPress plugins, phpMyAdmin, vBulletin). However, no automated Metasploit exploit module was applicable to the target system because the vulnerable upload functionality belonged to a **custom web application**.



```
706  exploit/unix/webapp/wp_infusionsoft_upload                        2014-09-25  excellent  Yes   Wordpress InfusionSoft Upload Vulnerability
707  exploit/multi/http/wp_litespeed_cookie_theft                      2024-09-04  excellent  Yes   Wordpress LiteSpeed Cache plugin cookie theft
708    \_ target: PHP In-Memory                                        .           .          .     .
709    \_ target: Unix In-Memory                                       .           .          .     .
710    \_ target: Windows In-Memory                                    .           .          .     .
711  exploit/unix/webapp/wp_wysija_newsletters_upload                  2014-07-01  excellent  Yes   Wordpress MailPoet Newsletters (wysija-newsletters) Unauthenticated File Upload
712  exploit/unix/webapp/wp_nmediawebsite_file_upload                  2015-04-12  excellent  Yes   Wordpress N-Media Website Contact Form Upload Vulnerability
713  exploit/multi/http/wp_plugin_backup_guard_rce                     2021-05-04  excellent  Yes   Wordpress Plugin Backup Guard - Authenticated Remote Code Execution
714  exploit/multi/http/wp_catch_themes_demo_import                    2021-10-21  normal     Yes   Wordpress Plugin Catch Themes Demo Import RCE
715  exploit/multi/http/wp_plugin_elementor_auth_upload_rce            2022-03-29  excellent  Yes   Wordpress Plugin Elementor Authenticated Upload Remote Code Execution
716  exploit/multi/http/wp_plugin_modern_events_calendar_rce           2021-01-29  excellent  Yes   Wordpress Plugin Modern Events Calendar - Authenticated Remote Code Execution
717  exploit/multi/http/wp_plugin_sp_project_document_rce              2021-06-14  excellent  Yes   Wordpress Plugin SP Project and Document - Authenticated Remote Code Execution
718  exploit/unix/webapp/wp_reflexgallery_file_upload                  2012-12-30  excellent  Yes   Wordpress Reflex Gallery Upload Vulnerability
719  exploit/unix/webapp/wp_slideshowgallery_upload                    2014-08-28  excellent  Yes   Wordpress SlideShow Gallery Authenticated File Upload
720  exploit/unix/webapp/wp_worktheflow_upload                         2015-03-14  excellent  Yes   Wordpress Work The Flow Upload Vulnerability
721  exploit/windows/http/xampp_webdav_upload_php                      2012-01-14  excellent  No    XAMPP WebDAV PHP Upload
722  exploit/unix/webapp/xoda_file_upload                              2012-08-21  excellent  Yes   XODA 0.4.5 Arbitrary PHP File Upload Vulnerability
723  exploit/linux/http/xplico_exec                                    2017-10-29  excellent  Yes   Xplico Remote Code Execution
724  exploit/windows/browser/yahoomessenger_server                     2007-06-05  good       No    Yahoo! Messenger 8.1.0.249 ActiveX Control Buffer Overflow
725    \_ target: Windows XP SP0/SP1 Pro English                       .           .          .     .
726    \_ target: Windows 2000 Pro English All                         .           .          .     .
727  exploit/linux/http/zabbix_sqli                                    2013-09-23  excellent  Yes   Zabbix 2.0.8 SQL Injection and Remote Code Execution
728  exploit/linux/http/zimbra_xxe_rce                                 2019-03-13  excellent  Yes   Zimbra Collaboration Autodiscover Servlet XXE and ProxyServlet SSRF
729  exploit/multi/http/apprain_upload_exec                            2012-01-19  excellent  Yes   appRain CMF Arbitrary PHP File Upload Vulnerability
730  exploit/unix/webapp/jquery_file_upload                            2018-10-09  excellent  Yes   blueimp's jQuery (Arbitrary) File Upload
731    \_ target: PHP Dropper                                          .           .          .     .
732    \_ target: Linux Dropper                                        .           .          .     .
733  exploit/multi/http/extplorer_upload_exec                          2012-12-31  excellent  Yes   eXtplorer v2.1 Arbitrary File Upload Vulnerability
734  exploit/unix/webapp/elfinder_php_connector_exiftran_cmd_injection 2019-02-26  excellent  Yes   elFinder PHP Connector exiftran Command Injection
735  exploit/unix/webapp/oscommerce_filemanager                        2009-08-31  excellent  No    osCommerce 2.2 Arbitrary PHP Code Execution
736  exploit/unix/http/pfsense_diag_routes_webshell                    2022-02-23  excellent  Yes   pfSense Diag Routes Web Shell Upload
737    \_ target: Unix Command                                         .           .          .     .
738    \_ target: BSD Dropper                                          .           .          .     .
739  exploit/multi/http/pgadmin_session_deserialization                2024-03-04  excellent  Yes   pgAdmin Session Deserialization RCE
740  exploit/unix/webapp/phpcollab_upload_exec                         2017-09-29  excellent  Yes   phpCollab 2.5.1 Unauthenticated File Upload
741  exploit/multi/http/qdpm_authenticated_rce                         2020-11-21  excellent  Yes   qdPM 9.1 Authenticated Arbitrary PHP File Upload (RCE)
742    \_ target: Generic (PHP Payload)                                .           .          .     .
743    \_ target: Linux x86                                            .           .          .     .
744    \_ target: Linux x64                                            .           .          .     .
745    \_ target: Windows x86                                          .           .          .     .
746    \_ target: Windows x64                                          .           .          .     .
747  exploit/multi/http/qdpm_upload_exec                               2012-06-14  excellent  Yes   qdPM v7 Arbitrary PHP File Upload Vulnerability
748    \_ target: Generic (PHP Payload)                                .           .          .     .
749    \_ target: Linux x86                                            .           .          .     .
750  exploit/linux/http/rconfig_vendors_auth_file_upload_rce           2021-03-17  excellent  Yes   rConfig Vendors Auth File Upload RCE
751  exploit/multi/http/vtiger_soap_upload                             2013-03-26  excellent  Yes   vTiger CRM SOAP AddEmailAttachment Arbitrary File Upload
752  exploit/multi/http/vtiger_php_exec                                2013-10-30  excellent  Yes   vTigerCRM v5.4.0/v5.3.0 Authenticated Remote Code Execution


Interact with a module by name or index. For example info 752, use 752 or use exploit/multi/http/vtiger_php_exec

msf > 
```

Figure 6: Metasploit search results for upload-based exploits.

Conclusion:

Manual exploitation techniques would be required to exploit this vulnerability, as no suitable automated Metasploit module exists.

9. Risk Assessment

9.1 CVSS Scoring

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

CVSS v3.1 Score: High (≈ 8.0)

9.2 Risk Matrix

Likelihood	Impact	Risk Level
High	High	**High**

The vulnerability poses a high risk due to the possibility of arbitrary file upload and potential remote code execution.

10. Remediation Recommendations

The following remediation steps are recommended:

1. Implement strict server-side file type whitelisting
2. Validate MIME types in addition to file extensions
3. Rename uploaded files and prevent direct execution
4. Disable script execution in upload directories
5. Store uploaded files outside the web root
6. Implement logging and monitoring for file uploads

11. Conclusion

This assessment successfully identified and validated a critical web application vulnerability related to insecure file upload functionality. Although basic security controls were present, reliance on extension-based filtering alone is insufficient. If exploited, this vulnerability could lead to severe security consequences, including remote code execution.

Proper remediation and secure coding practices are essential to mitigate this risk and strengthen the overall security posture of the application.

12. References

- OWASP Web Security Testing Guide
- OWASP Top 10
- NIST SP 800-115
- Metasploit Framework Documentation
- Nmap Documentation

- CVSS v3.1 Specification