ILKOM Jurnal Ilmiah Vol. 16, No. 2, August 2024, pp. 161-171
Accredited 2nd by RISTEKBRIN No. 200/M/KPT/2020; E-ISSN 2548-7779 | P-ISSN 2087-1716

Research Article Open Access (CC–BY-SA)

Vulnerability Assessment and Penetration Testing on Student Service Center System

Khairunnisak Nur Isnaini [a,1*]; Muhammad Hasyim Asyari [a,2]; Sigit Fathu Amrillah [a,3]; Didit Suhartono [a,4]

[a] Universitas Amikom Purwokerto, Jl. Letjend Pol. Soemarto No.127, Watumas, Purwanegara, Kec. Purwokerto Utara, Kabupaten Banyumas, Jawa Tengah 53127
[1] nisak@amikompurwokerto.ac.id; [2] pon.jdev@gmail.com; [3] sigitfathuamrillah@gmail.com; [4] didit@amikompurwokerto.ac.id
** Corresponding author*

Article history: Received March 18, 2024; Revised July 13, 2024; Accepted July 30, 2024; Available online August 20, 2024

Abstract

The number of system breaches has recently increased across various sectors, including the education sector. These breaches are carried out through various methods such as SQL Injection, XSS Attack, web defacement, malware, and others. Security vulnerabilities in the system also pose a potential threat to the Student Service Center owned by XYZ University, which stores a significant amount of confidential and sensitive data. The worst impact of all is the system is paralyzed, damaging the ongoing performance and reputation of institutions. The purpose of this research is to identify security vulnerabilities in the system using the Vulnerability Assessment and Penetration Testing (VAPT) method. The results showed that the system identified file upload functionality that poses a risk of being exploited for security attacks. Additionally, file path traversal can allow unauthorized access to directories, potentially enabling the injection of malicious code. Future research could explore the application of machine learning to enhance security measures and streamline the penetration testing process.

Keywords: Information Security; Penetration Testing; Security; Vulnerability; Vulnerability Assessment and Penetration Testing

Introduction

In the growing digital era, web applications have become an integral part of various aspects of human life, including education [1]. Universities, as centers of higher education, have also adopted this technology to provide information, services, and interaction with students or the general public online. However, the potential for significant security risks always coexist with technological developments [2]. Risk is an undesirable event that results from an occurrence or event that has a detrimental impact [3]. It occurs due to a lack of security implementation that can increase security risks [4]. In this context, security vulnerabilities in web applications have become a serious concern for stakeholders, especially due to the rise of cyber-attacks targeting various types of sites, including college sites with academic domains (ac.id). One type of attack that has emerged is gambling site injection attacks, where attackers try to exploit vulnerabilities in web applications to inject advertisements, links, or unwanted content, which in this case is related to gambling sites.

In recent times, there have been several cyberattack incidents in Indonesia, including Web Defacement incidents targeting Government and Education sites. One of the more notable incidents is the "*Slot Gacor* or Online Gambling Web Defacement," in which attackers changed the appearance of a site by replacing it with an online gambling site. In recent times, there have been several cyberattack incidents in Indonesia, including web defacement incidents targeting government and education sites. One of the more notable incidents is the "*Slot Gacor* or online gambling web defacement, in which attackers changed the appearance of a site by replacing it with an online gambling site. [5]. Works by exploiting vulnerable websites to launch malicious code that damages, deletes, or alters the content of web pages [6]. Defacement attacks can disrupt website operations, damage the owner's reputation, and potentially result in data loss. [7]. The number reached 1,650,000 higher education institution websites with the .ac.id domain, 2,090,000 government sites with the .go.id domain, and 313,000 school sites with the .sch.id domain, all of which were victims of slot gambling injection attacks [8]. The attack affects the appearance of the website, turning it into an online gambling page. The motive behind this action is to prevent the site from being blocked by the authorities, which is why educational sites are one of the main targets [5]. An agency or organization requires adequate and supportive resources, such as information that must be maintained and protected from various kinds of disasters and threats [9].

This indicates vulnerabilities in the design or implementation of web applications that allow attackers to access and manipulate data, including uploading unauthorized content. Among vulnerabilities other than those previously mentioned (malware, viruses, SQL Injection), vulnerabilities such as file path traversal and unrestricted file upload

https://doi.org/10.33096/ilkom.v1xix.xxx.x-x

are of significant concern. Path traversal allows attackers to access files and directories outside of the web root directory by exploiting the "unsafe direct object reference" vulnerability [10]. By exploiting the absolute URL pointing to a file, attackers modify a variable to access files outside the web root directory. Simply by adding the "../" character repeatedly, they can jump to another directory. For example, with the string "../../../etc/passwd", they can access the password system file on Linux [11]. While unrestricted file upload is a serious vulnerability that has a great impact on the system and its infrastructure. This vulnerability can result in a total system takeover, including the back end and database. An unrestricted file upload vulnerability could allow an attacker to upload and execute malicious scripts on a web server, potentially leading to the exploitation of the system [12], [13]. Attackers who can upload malicious files can perform drive-by download attacks, crash websites, or access file systems through a web shell. After gaining remote access to the server, they can extract data.

XYZ University is a private college that specializes in Computer Information Technology and has implemented a web-based information system. The university's website, xyz, provides a virtual representation of the campus [14]. XYZ University offers a Student Service Center system, an academic tool designed to expedite the processing of academic information on a web-based platform [15]. The Student Service Center website caters to various student needs, such as course registration (KRS), thesis submission, attendance verification, academic transcripts (KHS), and more. Additionally, the Student Service Center system features a profile menu that includes comprehensive personal and parental information, including the National Identification Number (NIK) data.

In general, possible security risks that can occur on a website include malware, viruses, and SQL Injection [8]. The risk does not rule out the possibility of occurring on the website used by XYZ University. The most common vulnerability in web applications is injection. Injection attacks utilize various flaws to send untrusted user input, which is then processed by the web application [16]. The attacks caused by SQL injection are significant and pose a serious threat to web-based applications because hackers can easily take over their systems. [17]. According to [18] the most common vulnerability currently in web applications is Cross-Site Scripting (XSS) attacks. According to [19] IDOR (Insecure direct object reference) is a security vulnerability that occurs when a web application does not validate or authorize access to direct objects, such as data or resources, adequately manner and is as vulnerable as the ID object. Attackers can easily infiltrate the system and deploy a backdoor virus to a user's machine or a main server [20]. These risks are additional elements that influence the adoption of certain technologies. When risks to information security occur, the impact can disrupt the smooth flow of work, increase the use of time and financial resources, and potentially damage the organization's reputation. It is important to identify and manage risks to ensure a better level of system security and to reduce the impact that may occur to be more controlled [21].

Security risks that arise on websites can be mitigated by implementing information security techniques or methods. Organizations, governments, and other and other individual interests can mitigate security vulnerability risks with a variety of measures [22]. On the other hand, information security plays a crucial role in safeguarding information within a company, organization, or educational institution. When information security at an educational institution is not properly managed, it can pose a risk to the continued functionality of the information system. [3]. All organizations emphasize the importance of information security in protecting data and information as valuable assets [21]. As website users, it is essential to ensure that the technology used is secure and protected from potential security attacks [23]. Security in web applications is a critical aspect that should be owned.

In general, security risks on the web can be addressed by improving information security. One effective approach is through vulnerability assessment techniques, Vulnerability Assessment and Penetration Testing (VAPT) is a method used to secure web applications and systems by auditing and exploiting vulnerabilities [24]. It encompasses a variety of security assessment services aimed at identifying and mitigating risks within an organization's IT infrastructure [17]. Therefore, it is crucial to conduct a vulnerability assessment to explore vulnerabilities, weaknesses, and deficiencies in a system. Vulnerability assessment is an internal and external security evaluation activity that involves identifying systems, network computers, applications, and other potential targets of illegal attacks [25]. Implementing a set of management and control mechanisms to enhance security is essential [26]. By conducting a vulnerability assessment, institutions, organizations, and individuals can proactively address security issues before they are discovered by hackers and exploited for financial gain and other malicious purposes [18].

Based on previous research [27] VAPT on government websites, identifying various vulnerabilities and their risk levels. In another study, VAPT was also used to analyze vulnerabilities in the use of cloud model services, specifically Software as a Service (SaaS), Infrastructure as a Service (IaaS), and Platform as a Service (PaaS) from the perspective of penetration testers [28]. Research conducted by [29] has attempted to address the security weaknesses of computer network systems by integrating CVSS and FMEA methods to determine the priority of handling based on the potential impact of vulnerabilities. Previous research emphasizes the proactive role of VAPT as a cybersecurity defense technology. [21] describes the VAPT lifecycle, vulnerability assessment techniques, and tools used to enhance cyber defense. Research conducted by [22] has emphasized the significance of VAPT in analyzing modern cybersecurity, as well as its advantages in safeguarding IT infrastructure and data from cyber threats. A previous study [7] highlighted an increase in hacking attacks and data leaks in Indonesia, with a focus on backdoor slot attacks on various domains, particularly in government and educational institutions.

This study highlights the need for rapid response and mitigation measures in the face of continued cyberattacks. Potential future research could include further analysis of affected domains, attack sources, threat modeling, and mitigation and recovery strategies. This research examines potential attacks that utilize path traversal and unrestricted files. Specifically, it aims to identify security vulnerabilities on the website of XYZ University that could be exploited for further illegal attacks.

Method

The author chose the VAPT method because of its orderly sequence of steps, and it is easy to understand because it uses the penetration testing stages that are often used during assessments. The strength of this method lies in its initiation with the scanning of security gaps in the studied application, followed by the delivery of appropriate suggestions to fix the identified vulnerabilities [30]. Depicted in **Figure 1**, the VAPT Lifecycle, is a thorough nine-step process to assess and potentially exploit the security of a system or device. The process begins with defining the scope of the assessment, either from a black, gray, or white box perspective and gathering critical target information during reconnaissance. Vulnerabilities are identified through various techniques, leading to a penetration testing plan, where access and privileges in the system are pursued. At the result analysis stage, findings are examined, and recommendations are documented for management action. During this process, the victim system and its programs may be changed, but the cleanup phase ensures the return of the system to its pre-assessment state. This comprehensive approach realizes the VAPT lifecycle, protecting systems and devices from potential threats.

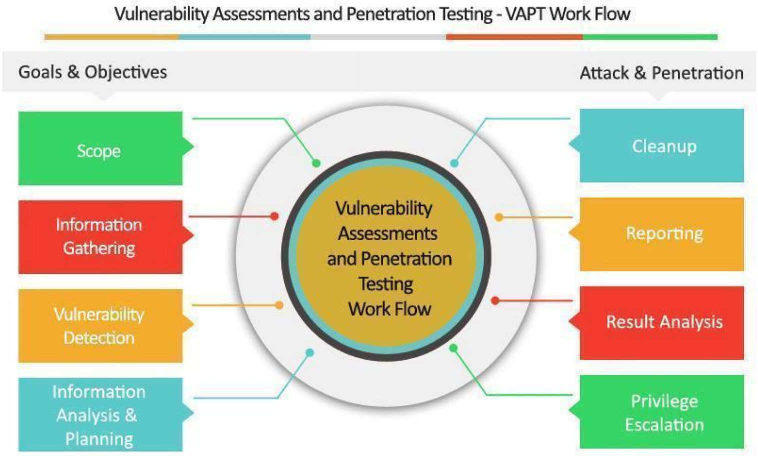

Vulnerability Assessments and Penetration Testing - VAPT Work Flow
Goals & Objectives
Attack & Penetration
Scope
Information Gathering
Vulnerability Detection
Information Analysis & Planning
Vulnerability Assessments and Penetration Testing Work Flow
Cleanup
Reporting
Result Analysis
Privilege Escalation

Figure 1. The life cycle of VAPT (Source: https://firewall.firm.in/vapt/)

A. Scope

Within the Assessment and Testing phase, it is crucial to define the scope of the assignment precisely. There are three potential scopes available: white box, black box, and grey box [30]. In this study, a grey box approach is utilized due to the authenticated portal system under examination. Grey box testing is used to gather information limited to the system or network under study, while also being able to identify and address internal and external security issues posed by attackers [31]. Grey box testing was chosen for this study because it includes integration testing, validation testing, system testing based on program code, and specification and design requirements [32]. Grey box testing is a hybrid technique that combines elements of both black box and white box testing and is conducted on either domestic or external networks [33].

B. Information Gathering

The information-gathering process involves a series of actions aimed at obtaining extensive internal data related to the IT environment, including details such as IP addresses, network configurations, operating system versions, and more. This step applies to all three types of scope discussed earlier. This stage focuses on identifying the menus within the Student Service Center system that offer file download and upload features. Additionally, it aims to gather information about the framework, programming languages, and operating systems used in the system [33].

C. Vulnerability Detection

In this stage, among various vulnerability assessment techniques, such as Static Analysis, Manual Testing, Automated Testing, and Fuzz Testing, the authors have adopted two techniques: manual and automated. In the manual approach, researchers rely on their own knowledge and experience, without depending on any tools or software, to identify vulnerabilities. For the automated method, the authors only use Burp Suite to detect menus that contain file upload and download functions [33].

D. Information Analysis and Planning

This phase is employed to assess vulnerabilities that have been identified concerning the information processed within the IT environment, and to devise a strategy for infiltrating the system and network. This means that the results of vulnerability detection are analyzed whether there are false positives or vice versa, which then from the results is used for the penetration testing / exploitation process in an effort to penetrate the system [33].

E. Penetration Testing

This phase is used to assess vulnerabilities that have been identified in the information processed within the IT environment, aiming to devise a strategy for penetrating the system and network [34]. This involves analyzing the results of vulnerability detection to determine the presence of false positives or false negatives. Subsequently, these results are utilized in the penetration testing/exploitation process to penetrate the system [33].

F. Privilege Escalation

After successfully hacking into the system, this technique is used to recognize and obtain higher access rights, such as root privileges and administrative access to the system [33].

G. Results Analysis

At this stage, root causes are carefully identified, pathways to exploitation are detected, and recommendations are designed to address the risk or weakness. This is the final stage in the lifecycle that ensures a thorough explanation of the problem or risk, methods to address the problem, and improve the security of the system or device [33].

H. Reporting

Reports are created and well-documented, including details about the areas that have been accessed, the methods and tools used in the vulnerability assessment, detailed information regarding the data found and lost, changes made to the system or device, total time spent, and so on [33].

I. Cleanup

In the final stage, a cleanup process is also performed to delete temporary files and restore the system to its original state [33].

Results and Discussion

The author has defined the scope for penetration testing in the Student Service Center system using the Grey Box method, incorporating both manual and automated techniques. The prepared resources include student account credentials, the Burp Suite software, and a web browser. Subsequently, the comprehensive test results are outlined in each stage of the VAPT lifecycle, spanning from Information Gathering to Cleanup.

A. Information Gathering

The author utilized the available student credentials to authenticate to the Student Service Center and access all menus in search of file download and upload features. BurpSuite also automatically recorded all requests on the URLs accessed by the author, allowing it to identify which endpoints provided file download and upload functionality. The results of this information collection are presented in **Table 1** below.

Table 1. The menu contains upload and download features

Menu with download and upload features	Details		
	Host	***Path***	***Features***
Detail Pengumuman Informasi Akademik	https://student.xyz.ac.id/	/main/detail_pengumuman/{id_pengumuman}	*Download*
(MBKM) Fakultas Ilmu Komputer	https://student.xyz.ac.id/	/mbkm_fakultas	*Download & upload*
Kampus Merdeka (MSIB)	https://student.xyz.ac.id/	/mbkm_kemendikbud	*Download & upload*
Sertifikasi Kompetensi	https://student.xyz.ac.id/	/sertifikasi_kompetensi	*Upload*
Organisasi Mahasiswa	https://student.xyz.ac.id/	/organisasi_mahasiswa	*Upload*
Prestasi Mahasiswa	https://student.xyz.ac.id/	/prestasi	*Upload*
Seminar Atau Workshop	https://student.xyz.ac.id/	/seminar-workshop	*Upload*
Perubahan Biodata	https://student.xyz.ac.id/	/biodata?act=mahasiswa	*Upload*

From **Table 1**, it can be seen that the update and download menus on the student.xyz.ac.id host can be detected using the BurpSuite tools. This includes identifying the path and each feature, which can involve both download and upload or only upload.

In addition to the information about the menu mentioned earlier, the author also managed to collect information related to the server, including the framework, programming language, storage, JavaScript library, as well as the user interface framework used. This kind of information provides significant support in speeding up and simplifying the penetration testing process. To obtain such information, the author used a tool called Wappalyzer installed as an extension on the Chrome web browser.

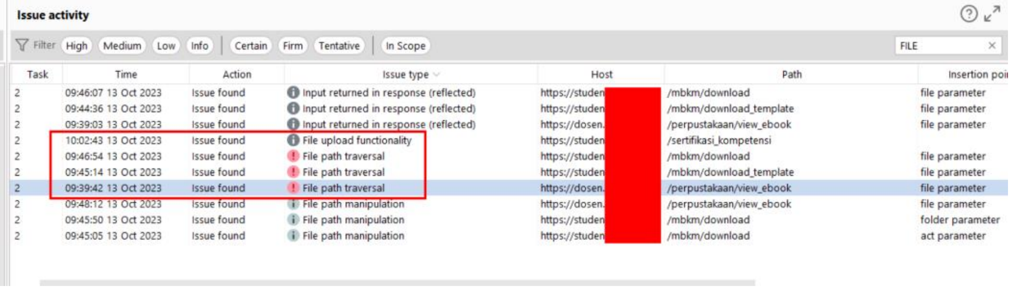
Table 2. Server Information

Server Information	
Technologies	***Version***
Framework Codeigniter	Undetected
Apache HTTP Server	2.4.56
PHP	Undetected
Ekstensi Web Server OpenSSL	1.0.2k
Operation System : UNIX	Undetected
Control Web Panel (CWP)	Undetected

Based on **Table 2**, the author gathers information about the server, the technology used, and the software version using the Wappalyzer and Burpsuite extensions. This information is identified through the request and response folders. Among the technologies identified are Codeigniter Framework, PHP, and the operating system, as well as Control Web Panel Information, some of which have unknown versions. The information obtained is highly valuable for the following stage.

B. Vulnerability Detection

Based on the information obtained in the previous stage, URLs that provide download and upload features are automatically detected by Burp Suite, while also displaying the vulnerabilities present in these URLs.



Task	Time	Action	Issue type	Host	Path	Insertion poi
2	09:46:07 13 Oct 2023	Issue found	Input returned in response (reflected)	https://studen[redacted]	/mbkm/download	file parameter
2	09:44:36 13 Oct 2023	Issue found	Input returned in response (reflected)	https://studen[redacted]	/mbkm/download_template	file parameter
2	09:39:03 13 Oct 2023	Issue found	Input returned in response (reflected)	https://dosen.[redacted]	/perpustakaan/view_ebook	file parameter
2	10:02:43 13 Oct 2023	Issue found	File upload functionality	https://studen[redacted]	/sertifikasi_kompetensi	
2	09:46:54 13 Oct 2023	Issue found	File path traversal	https://studen[redacted]	/mbkm/download	file parameter
2	09:45:14 13 Oct 2023	Issue found	File path traversal	https://studen[redacted]	/mbkm/download_template	file parameter
2	09:39:42 13 Oct 2023	Issue found	File path traversal	https://dosen.[redacted]	/perpustakaan/view_ebook	file parameter
2	09:48:12 13 Oct 2023	Issue found	File path manipulation	https://dosen.[redacted]	/perpustakaan/view_ebook	file parameter
2	09:45:50 13 Oct 2023	Issue found	File path manipulation	https://studen[redacted]	/mbkm/download	folder parameter
2	09:45:05 13 Oct 2023	Issue found	File path manipulation	https://studen[redacted]	/mbkm/download	act parameter

Figure 2. Vulnerability detection results

As shown in **Figure 2**, the vulnerability detection results reveal the existence of endpoints with file upload functionality and a high severity file path traversal vulnerability in the download feature. Therefore, it can be inferred that the goal of identifying download and upload features has been accomplished. Nevertheless, the findings of this audit undergoes further analysis, and preparations is made for a manual penetration test at a later stage.

C. Information Analysis and Planning

The results of automatic detection from burp suite based on **Figure 2**, and **Table 1** are then mapped and planned exploitation of the upload, download and gap features detected by burp suite. The mapping is as in **Table 3**.

Table 3. Penetration testing planning

Issue detection	**Severity**	**Scenario Testing**
File upload functionality (Menu Sertifikat kompetensi, Organisasi mahasiswa, Prestasi mahasiswa, Seminar, Biodata mahasiswa dan MBKM.)	Info	Testing is done by uploading backdoor files with certain extension variations such as .php, .php2, .php3, .php4, .php5, .php6, .php7, file.php%00, and so on.
File path traversal on transcript menu	Info	Attempt to access configuration files (database, api, environment), and server logs.

Issue detection	Severity	Scenario Testing
File path traversal on MBKM (*Fakultas*) menu	Medium	Attempt to access configuration files (database, api, environment), and server logs.
File path traversal on MBKM (Kemendikbud) menu	Medium	Attempt to access configuration files (database, api, environment), and server logs.
File path traversal on detail *"pengumuman menu"*	High	Attempt to access configuration files (database, api, environment), and server logs.

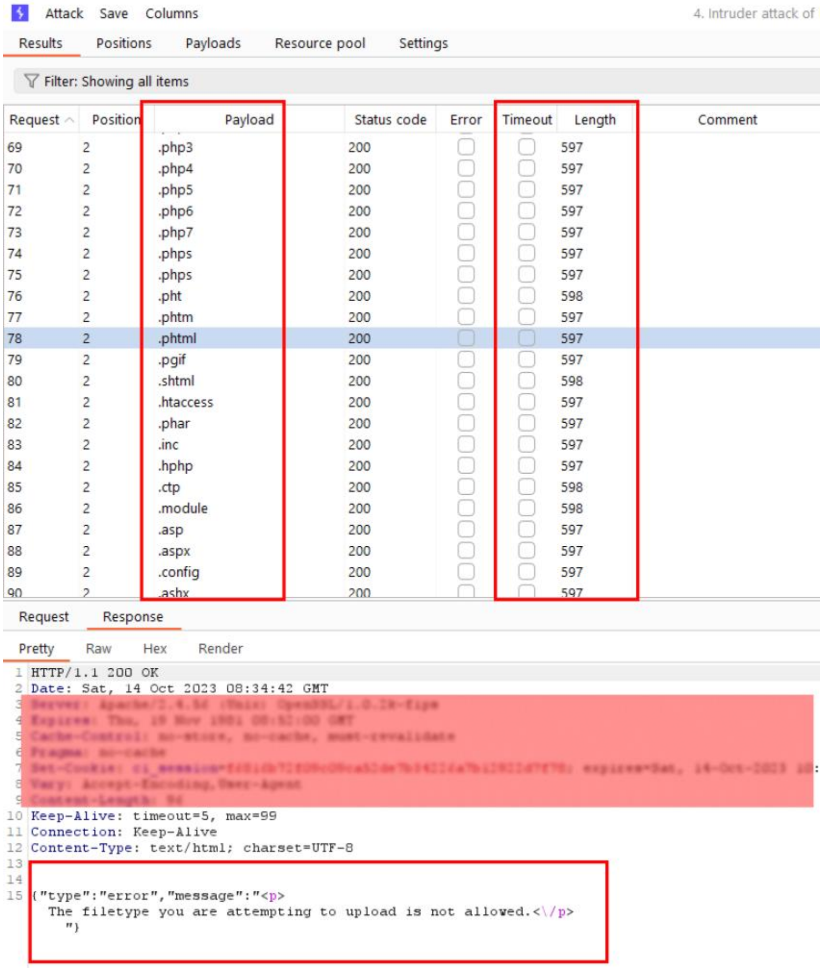
In **Table 3**, the scanning results reveal that the File upload functionality and File path traversal issues are grouped into several severity levels. The results indicate 2 issues in the information category, 2 issues at the medium level, and 1 issue at the high level. This aligns with the research findings [35] indicating that an attacker who exploits a directory traversal attack on a web server can have unrestricted access. By combining this with code injection, they can upload a shell onto the web server and execute website defacement attacks. As for the file upload functionality vulnerability, once exploited by an attacker, it can be used to launch various attacks such as installing a web shell, contaminating web applications, spreading malware, and phishing [12].

D. Penetration Testing

After analyzing and planning, the testers focused on two vulnerabilities: file upload and file path traversal. Although the detection results from Burp Suite showed other vulnerabilities, the discussion below focused on the test results of these two vulnerabilities.

1. Unrestricted file upload

In this vulnerability test, backdoor uploads are performed on all menus that provide file upload features.



Request	Position	Payload	Status code	Error	Timeout	Length	Comment
69	2	.php3	200	□	□	597	
70	2	.php4	200	□	□	597	
71	2	.php5	200	□	□	597	
72	2	.php6	200	□	□	597	
73	2	.php7	200	□	□	597	
74	2	.phps	200	□	□	597	
75	2	.phps	200	□	□	597	
76	2	.pht	200	□	□	598	
77	2	.phtm	200	□	□	597	
78	2	.phtml	200	□	□	597	
79	2	.pgif	200	□	□	597	
80	2	.shtml	200	□	□	598	
81	2	.htaccess	200	□	□	597	
82	2	.phar	200	□	□	597	
83	2	.inc	200	□	□	597	
84	2	.hphp	200	□	□	597	
85	2	.ctp	200	□	□	598	
86	2	.module	200	□	□	598	
87	2	.asp	200	□	□	597	
88	2	.aspx	200	□	□	597	
89	2	.config	200	□	□	597	
90	2	.ashx	200	□	□	597	

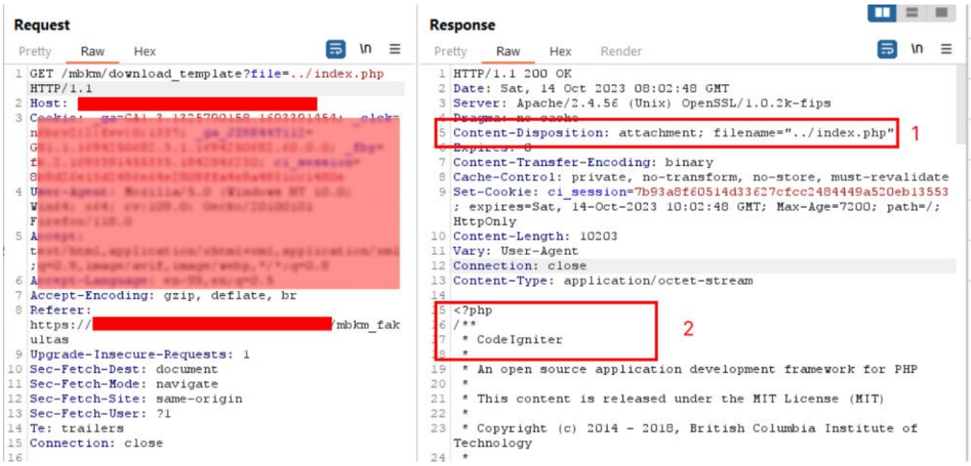
```
HTTP/1.1 200 OK
Date: Sat, 14 Oct 2023 08:34:42 GMT
[redacted]
Keep-Alive: timeout=5, max=99
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8


{"type":"error","message":"<p>
  The filetype you are attempting to upload is not allowed.<\/p>
    "}
```

Figure 3. Test with backdoor file upload

As illustrated in Figure 3, using Intruder in Burp suite, a brute force test was performed on file extensions that can be used as a backdoor, such as .php, .php2, .php3, .php4, .php5, .php6, .php7, file.php%00, and so on. However, the test results show that none of the backdoors were successfully uploaded with these extensions. This is evidenced in the response in Figure 3, where the server returns the message "error: The filetype you are attempting to upload is not allowed.", and with the content length remaining the same, at 597. Note that the red block on each image is an attempt to mask information that is not possible for the public to know.

2. *File Path Traversal*

In testing this vulnerability, the main stage to ensure the vulnerability is not false positive, is done by downloading the index.php file backwards in the previous directory by means of "../index.php". This test is carried out on the download template feature in the mbkm menu.

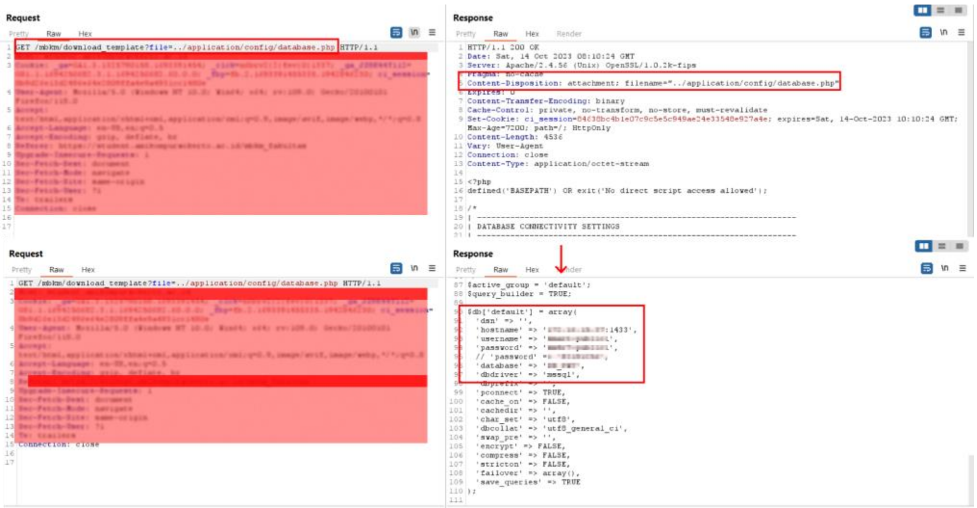


```
GET /mbkm/download_template?file=../index.php
HTTP/1.1
Host: [redacted]
[redacted]
Accept-Encoding: gzip, deflate, br
Referer:
https://[redacted]/mbkm_fak
ultas
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Te: trailers
Connection: close
```

```
HTTP/1.1 200 OK
Date: Sat, 14 Oct 2023 08:02:48 GMT
Server: Apache/2.4.56 (Unix) OpenSSL/1.0.2k-fips
Pragma: no-cache
Content-Disposition: attachment; filename="../index.php"
Expires: 0
Content-Transfer-Encoding: binary
Cache-Control: private, no-transform, no-store, must-revalidate
Set-Cookie: ci_session=7b93a8f60514d33627cfcc2484449a520eb13553
; expires=Sat, 14-Oct-2023 10:02:48 GMT; Max-Age=7200; path=/;
HttpOnly
Content-Length: 10203
Vary: User-Agent
Connection: close
Content-Type: application/octet-stream

<?php
/**
 * CodeIgniter
 *
 * An open source application development framework for PHP
 *
 * This content is released under the MIT License (MIT)
 *
 * Copyright (c) 2014 - 2018, British Columbia Institute of
Technology
 *
```

Figure 4. Access index.php with path traversal vulnerability

It can be seen in Figure 4 that requesting "../index.php" produces a response that the system uses CodeIgniter. Red box number 2 proves that it is true that the system uses the CodeIgniter framework and with the results of information collection in Table 2.



```
GET /mbkm/download_template?file=../application/config/database.php HTTP/1.1
[redacted]
```

```
HTTP/1.1 200 OK
Date: Sat, 14 Oct 2023 08:10:24 GMT
Server: Apache/2.4.56 (Unix) OpenSSL/1.0.2k-fips
Pragma: no-cache
Content-Disposition: attachment; filename="../application/config/database.php"
Expires: 0
Content-Transfer-Encoding: binary
Cache-Control: private, no-transform, no-store, must-revalidate
Set-Cookie: ci_session=84638bc4b1e07c9c5e5c949ae24e33548e927a4e; expires=Sat, 14-Oct-2023 10:10:24 GMT;
Max-Age=7200; path=/; HttpOnly
Content-Length: 4536
Vary: User-Agent
Connection: close
Content-Type: application/octet-stream

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

/*
| -------------------------------------------------------------------
| DATABASE CONNECTIVITY SETTINGS
| -------------------------------------------------------------------
```

```
GET /mbkm/download_template?file=../application/config/database.php HTTP/1.1
[redacted]
Connection: close
```

```
$active_group = 'default';
$query_builder = TRUE;

$db['default'] = array(
  'dsn' => '',
  'hostname' => '[redacted]:1433',
  'username' => '[redacted]',
  'password' => '[redacted]',
  // 'password' => '[redacted]',
  'database' => '[redacted]',
  'dbdriver' => 'mssql',
  'dbprefix' => '',
  'pconnect' => TRUE,
  'cache_on' => FALSE,
  'cachedir' => '',
  'char_set' => 'utf8',
  'dbcollat' => 'utf8_general_ci',
  'swap_pre' => '',
  'encrypt' => FALSE,
  'compress' => FALSE,
  'stricton' => FALSE,
  'failover' => array(),
  'save_queries' => TRUE
);
```

Figure 5. Access database file

Then as shown in Figure 5, the request command to download the confirmed database can be executed, as depicted in the response in Figure 5 above, including the database, username, password, and other settings. Therefore, it can be inferred that the conducted tests align with the plan outlined in Table 3, which involves attempting to access configuration files (database, API, environment) and server logs.

E. *Privilege Escalation*

Privilege escalation is an attacker's attempt to increase access rights. For example, someone who previously had only read access can gain write and delete permissions through successful privilege escalation. The results of file traversal testing revealed that the IP used as the system database hostname is a private or local IP that cannot be accessed from external networks. The author attempted to increase access by connecting the device to the same network and then accessing the database. However, the test results did not succeed in elevating access rights. Nevertheless, this vulnerability allows access to all files on the student server (read access).

F. Results Analysis

The author has analyzed the test results conducted in the previous stage to identify cases that fall into the False Positive and False Negative categories. Then, they calculate the severity score based on the CVSs.

Table 4. Analysis result

Vulnerability Name	CVVS calculator result		
	Severity	Score	False Positive
File path traversal on Detail Pengumuman Informasi Akademik menu	NA	0	Y
File path traversal on transkip menu	NA	0	Y
File path traversal on MBKM (Fakultas) menu	Medium	4.9	N
File path traversal on MBKM (Kemendikbud) menu	Medium	4.9	N
File path traversal on detail pengumuman menu	High	7.5	N
File upload unrestricted on MBKM (Fakultas dan Kemendikbud) menu	NA	0	Y
File upload unrestricted (Sertifikat kompetensi) menu	NA	0	Y
File upload unrestricted (Organisasi mahasiswa) menu	NA	0	Y
File upload unrestricted (Prestasi mahasiswa) menu	NA	0	Y
File upload unrestricted (Seminar atau workshop) menu	NA	0	Y
File upload unrestricted (Biodata) menu	NA	0	Y

The analysis results in Table 4 indicate the presence of a Medium severity vulnerability, which has a score value of 4.9. This assessment is based on the CVSS:3.1 scheme with parameters AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:N/A:N (Attack Vector: Network, Attack Complexity: Low, Privileges Required: High, User Interaction: None, Scope: Unchanged, Confidentiality: High, Integrity: None, Availability: None). The medium severity assessment is caused by several factors, including the existence of menus that require user (student) authentication to take advantage of this vulnerability so that the privileges required are rated high, as well as the potential leakage of all information contained in the student system source code due to exploitation of these vulnerabilities, which results in a high level of confidentiality assessment and which gets a high severity level because the privileges required are none. While the false positive is due to the web application firewall and the blocking of unauthorized files on the server side in all upload menus.

G. Reporting

The following is a report on the results of penetration testing, this report contains the name of the vulnerability, the vulnerable endpoint, and suggestions for improvement.

Table 5. Reporting based on testing vulnerability

Vulnerability Name	Details	
	Endpoint	Remediation
File path traversal on MBKM (Fakultas) menu	*/mbkm/download_template?file={payload}*	- Validate the user input before processing it. Ideally, compare the user input with a whitelist of permitted values. If that isn't possible, verify that the input contains only permitted content, such as alphanumeric characters only. - After validating the supplied input, append the input to the base directory and use a platform filesystem API to canonicalize the path. Verify that the
	/mbkm/download?act=kampus_merdeka&folder=file_pendaftaran&file={payload}	
File path traversal on MBKM (Kemendikbud) menu	*/mbkm/download_template?file={payload}*	
	/mbkm/download?act=kampus_merdeka&folder=file_komitmen&file={payload}	
	/mbkm/download?act=kampus_merdeka&folder=bukti_pembayaran&file={payload}	
	/mbkm/download?act=kampus_merdeka&folder=file_lolos&file={payload}	
File path traversal on detail pengumuman	*/perpustakaan/view_ebook?file={payload}*	

Vulnerability Name	Details	
	Endpoint	Remediation
		canonicalized path starts with the expected base directory. - Use Firewalls

Table 5 contains vulnerability issues, vulnerability locations and prevention recommendations that can be done at the student service center. This is to prevent the potential utilization of file path traversal vulnerabilities contained in the student center system by other parties.

H. Cleanup

This attempt was unsuccessful because in the course of this research, the author was unable to obtain super user access rights. However, it should be explained that within the scope of this research, no modifications were made to the database, or utilization of the information obtained. Therefore, this stage is not considered a necessity.

Conclusion

By applying the vulnerability assessment and penetration testing methods in this test, all stages in it can be applied properly, it's just that the test was not successful until getting superuser access rights. The penetration testing report shows that overall, the student center service system is proven safe from unrestricted file upload vulnerabilities, because the system has successfully prevented the use of unwanted file extensions, this is evidenced by the results of the file upload vulnerability analysis labeled NA (Not Applicable).

While file path traversal was successfully exploited and the penetration test report results show the download feature located on the MBKM menu and academic collection details are vulnerable to file path traversal with medium to high severity. If you look at the results of cvvs, this vulnerability has a very strong impact on confidential file information on the server at risk of leakage. Then mitigate the vulnerability of file path traversal that can be done by the xyz campus including, validating user input on the url download file before being processed by the server, or can apply waf (web application firewall).

The further research can build upon similar studies by having full access to the system. This allows for a noticeable difference in the vulnerability results obtained. In the future, vulnerability assessments with penetration testing could be automated by implementing a specific architecture or utilizing appropriate algorithms.

References

[1] A. Saeful, “Teknologi Dalam Bingkai Pendidikan,” *AL Fikr. J. Pemikir. dan Pendidik. Islam*, vol. 2, no. 1, pp. 41–54, 2022, doi : 10.51476/alfikrah.v2i1.357.

[2] A. R. M. Aditya, A. W. O. K. Putri, D. L. Musthofa, and P. Widodo, “Serangan Hacking Tools sebagai Ancaman Siber dalam Sistem Pertahanan Negara (Studi Kasus: Predator),” *Glob. Polit. Stud. J.*, vol. 6, no. 1, pp. 35–46, 2022, doi : 10.34010/gpsjournal.v6i1.6698.

[3] K. Isnaini, G. J. Nofita Sari, and A. P. Kuncoro, “Analisis Risiko Keamanan Informasi Menggunakan ISO 27005:2019 pada Aplikasi Sistem Pelayanan Desa,” *J. Eksplora Inform.*, vol. 13, no. 1, pp. 37–45, 2023, doi : 10.30864/eksplora.v13i1.696.

[4] K. N. Isnaini and D. Suhartono, “Security Analysis of Simpel Desa using Mobile Security Framework and ISO 27002:2013,” *INTENSIF J. Ilm. Penelit. dan Penerapan Teknol. Sist. Inf.*, vol. 7, no. 1, pp. 84–105, 2023, doi : 10.29407/intensif.v7i1.18742.

[5] Incident Response Team BSSN, “Paduan Penanganan Insiden Web Defacement Judi Online,” *Badan Siber dan Sandi Negara*, 2023.

[6] M. Albalawi, R. Aloufi, N. Alamrani, N. Albalawi, A. Aljaedi, and A. R. Alharbi, “Website Defacement Detection and Monitoring Methods: A Review,” *Electron.*, vol. 11, no. 21, 2022, doi: 10.3390/electronics11213573.

[7] T. H. Nguyen, X. Dau Hoang, and D. D. Nguyen, “Detecting Website Defacement Attacks using Web-page Text and Image Features,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 7, pp. 215–222, 2021, doi: 10.14569/IJACSA.2021.0120725.

[8] M. Fadli Mutaqin and D. Ferdiansyah, “Identifikasi Kerentanan Terhadap Serangan Slot Backdoor Pada Website di Indonesia Dengan Menggunakan Metode OSINT,” *J. Pas. Inform.*, vol. 1, no. 2, pp. 2986–5360, 2022.

[9] D. Suhartono and K. N. Isnaini, “Strategi Recovery Plan Teknologi Informasi di Perguruan Tinggi Menggunakan Framework NIST SP 800-34,” *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 20, no. 2, pp. 261–272, 2021, doi: 10.30812/matrik.v20i2.1097.

[10] A. A. Almutairi, S. Mishra, and M. AlShehri, "Web Security: Emerging Threats and Defense," *Comput. Syst. Sci. Eng.*, vol. 40, no. 3, pp. 1233–1248, 2021.

[11] H. Kapodistria, S. Mitropoulos, and C. Douligeris, "An advanced web attack detection and prevention tool," *Inf. Manag. Comput. Secur.*, vol. 19, no. 5, pp. 280–299, 2011, doi: 10.1108/09685221111188584.

[12] J. Huang, Y. Li, J. Zhang, and R. Dai, "UChecker: Automatically Detecting PHP-Based Unrestricted File Upload Vulnerabilities," in *49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2019*, 2019, no. June 2019, pp. 581–592, doi: 10.1109/DSN.2019.00064.

[13] J. Huang, J. Zhang, J. Liu, C. Li, and R. Dai, "UFuzzer: Lightweight Detection of PHP-based unrestricted file upload vulnerabilities via static-fuzzing co-analysis," in *RAID '21: Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses*, 2021, pp. 78–90, doi: 10.1145/3471621.3471859.

[14] A. D. Riyanto, M. Pinilih, A. Oktaviani, and E. Riyani, "Evaluasi Website Universitas Amikom Purwokerto," *JATISI*, vol. 9, no. 2, pp. 1–5, 2022.

[15] E. Marwati and D. Krisbiantoro, "Analisis Tingkat Kepuasan Pengguna Web Students Universitas Amikom Purwokerto Menggunakan Metode Eucs," *J. Inf. Syst. Manag.*, vol. 4, no. 2, pp. 67–72, 2023, doi: 10.24076/joism.2023v4i2.902.

[16] W. B. Demilie and F. G. Deriba, "Detection and prevention of SQLI attacks and developing compressive framework using machine learning and hybrid techniques," *J. Big Data*, vol. 9, no. 1, 2022.

[17] A. Alanda, D. Satria, M. Isthofa Ardhana, A. A. Dahlan, and A. Mooduto, "Web Application Penetration Testing Using SQL Injection Attack," *JOIV Int. J. Informatics Vis.*, vol. 5, no. September, pp. 320–326, 2021, doi : 10.30630/joiv.5.3.470.

[18] A. Alotaibi, L. Alghufaili, and D. M. Ibrahim, "Cross Site Scripting Attack Review," *ISeCure*, vol. 13, no. 3 Special Issue, pp. 21–30, 2021.

[19] R. Ananda Putra, I. Alnaurus Kautsar, HIndarto, and Sumarno, "Detection and Prevention of Insecure Direct Object References (IDOR) in Website-Based Applications Deteksi dan Pencegahan Insecure Direct Object References (IDOR) Pada Aplikasi Berbasis Website," in *Seminar Nasional & Call Paper Fakultas Sains dan Teknologi*, 2023, vol. 4, no. June, doi: 10.1186/s40537-022-00678-0.

[20] A. Fadlil, I. Riadi, and F. Fachri, "Mitigation Web Server for Cross-Site Scripting Attack Using Penetration Testing Method," *Int. J. Saf. Secur. Eng.*, vol. 12, no. 2, pp. 201–208, 2022.

[21] B. Fachriandi and T. Dirgahayu, "Kepedulian Keamanan Informasi di Pemerintahan: Praktik Manajemen dan Dampaknya," *J. Manaj. Inform.*, vol. 11, no. 1, pp. 72–87, 2021, doi: 10.34010/jamika.v11i1.4584.

[22] H. M. Adam and G. D. Putra, "A Review of Penetration Testing Frameworks , Tools , and Application Areas," in *2023 IEEE 7th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, 2024, no. November 2023, pp. 319–324, doi: 10.1109/ICITISEE58992.2023.10404397.

[23] Clintswood, D. G. Lie, L. Kuswandana, Nadia, S. Achmad, and D. Suhartono, "The Usage of Machine Learning on Penetration Testing Automation," in *International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS)*, 2023, no. February 2024, doi: 10.1109/ICE3IS59323.2023.10335188.

[24] A. Hasan and D. Meva, "Web Application Safety by Penetration Testing," in *International Conference on Cyber Security (ICCS)*, 2018, no. March, pp. 159–163.

[25] M. F. Safitra, M. Lubis, and A. Widjajarto, "Security Vulnerability Analysis using Penetration Testing Execution Standard (PTES): Case Study of Government's Website," in *Proceedings of the 2023 6th International Conference on Electronics, Communications and Control Engineering*, 2023, no. August, pp. 139–145, doi: 10.1145/3592307.3592329.

[26] L. Wang, R. Abbas, F. M. Almansour, G. S. Gaba, R. Alroobaea, and M. Masud, "An empirical study on vulnerability assessment and penetration detection for highly sensitive networks," *J. Intell. Syst.*, vol. 30, no. 1, pp. 592–603, 2021.

[27] A. Almaarif and M. Lubis, "Vulnerability Assessment and Penetration Testing (VAPT) Framework: Case Study of Government's Website," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 10, no. 5, pp. 1874–1880, 2020, doi: 10.18517/ijaseit.10.5.8862.

[28] N. E. A. Ismail, N. H. Ali, M. A. Jalil, F. Yunus, and A. D. Jarno, "A Proposed Framework of Vulnerability Assessment and Penetration Testing (VAPT) in Cloud Computing Environments from Penetration Tester Perspective," *J. Adv. Res. Appl. Sci. Eng. Technol.*, vol. 39, no. 1, pp. 1–14, 2024, doi: 10.37934/araset.39.1.114.

[29] B. W. Retna Mulya and A. Tarigan, "Pemeringkatan Risiko Keamanan Sistem Jaringan Komputer Politeknik

Kota Malang Menggunakan Cvss Dan Fmea," *Ilk. J. Ilm.*, vol. 10, no. 2, pp. 190–200, 2018.

[30] A. M. Ibrahim, T. Defisa, and H. B. Seta, "Analisis Keamanan Sistem pada Website Perusahaan CV. Kazar Teknologi Indonesia dengan Metode Vulnerability Assesment and Penetration Testing (VAPT)," in *Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)*, 2022, no. April, pp. 312–325.

[31] M. Alhamed and M. M. H. Rahman, "A Systematic Literature Review on Penetration Testing in Networks: Future Research Directions," *MDPI J. Appl. Sci.*, vol. 13, no. 12, 2023, doi: 10.3390/app13126986.

[32] M. TanLi, Y. Zhang, Y. Wang, and Y. Jiang, "Grey-box technique of software integration testing based on message," *J. Phys. Conf. Ser.*, vol. 2025, no. 1, 2021, doi: 10.1088/1742-6596/2025/1/012096.

[33] Y. Khera, D. Kumar, Sujay, and N. Garg, "Analysis and Impact of Vulnerability Assessment and Penetration Testing. 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 2019, pp. 525–530, doi: 10.1109/COMITCon.2019.8862224.

[34] B. A. Chandrakant and J. P. Prakash, "Vulnerability Assessment and Penetration Testing As Cyber Defence," *Int. J. Eng. Appl. Sci. Technol.*, vol. 4, no. 2, pp. 72–76, 2019, doi: 10.1016/j.procs.2015.07.458.

[35] M. Chawda, D. P. Sharma, and M. J. Patel, "Deep Dive into Directory Traversal and File Inclusion Attacks leads to Privilege Escalation," *Int. J. Sci. Res. Sci. Eng. Technol.*, vol. 4099, pp. 115–120, 2021.