Uncovering the Risk of Academic Information System Vulnerability through PTES and OWASP Method

Ferzha Putra Utama[1]* and Raden Muhamad Hilmi Nurhadi[2]
[1]Information System Department, Faculty of Engineering, University of Bengkulu
Bengkulu, Indonesia 38371
[2]Informatics Department, Faculty of Engineering, University of Bengkulu
Bengkulu, Indonesia 38371
Email: [1]fputama@unib.ac.id, [2]hilminurhadi2@gmail.com

Abstract—**The security of academic information systems needs consideration to anticipate various threats, resulting in data leakage, misuse of information, modification, and data destruction. There are 36 public and private universities that utilize the academic information system provided by the software developed by Company XYZ. Limited resources in universities contribute to the weak handling of vulnerabilities in academic information systems. The research aims to determine the vulnerability level of academic information systems developed by Company XYZ through penetration testing. The research employs a deductive approach to explore academic system vulnerabilities based on incidents related to system security issues at a university. The research utilizes a combination of two testing methods: Penetration Testing Execution Standard (PTES) and Open Web Application Security Project (OWASP), chosen for their reliability, ease of use, and support by penetration testing tools. Penetration testing follows the PTES, involving seven steps: pre-engagement interaction, information collection, threat modeling, vulnerability analysis, exploitation, post-exploitation, and reporting. The threat focus in the research aligns with the top 10 of 2021 OWASP, ranking the ten most critical security risks. Results reveal eight critical security issues based on measurements using the Common Vulnerability Scoring System (CVSS) method. There are two high-level vulnerabilities, five medium-level vulnerabilities, and one low-level vulnerability. Moreover, the three principal vulnerabilities are Structured Query Language (SQL) Injection, broken access control, and weak encryption. Universities can enhance data integrity by independently remediating vulnerabilities discovered in the research. Furthermore, universities are encouraged to raise awareness within the academic community regarding the security of academic data.**

Index Terms—**Academic Information System Vulnerability, Penetration Testing Execution Standard (PTES), Open Web Application Security Project (OWASP)**

I. Introduction

INFORMATION systems have been widely used to manage academic activities. The academic community in higher education obtains its information through an information system [1]. Additionally, most universities' information systems are managed through servers in their activities to provide academic information. It makes the academic information system an important infrastructure to ensure the continuity of academic services [2, 3]. The management of academic data in higher education institutions requires security guarantees, including universities in Indonesia. University server security is necessary to manage networks, authenticate, and prevent the theft of academic data and information [4]. The management of study, courses, and student data is centrally handled by the Indonesian government. Centralized and integrated data make it easier for each university to process data according to the business process. The gathering of academic data from all higher education institutions in Indonesia makes it significant, constituting big data. Academic big data integrates computing resources to effectively improve the provision of information and aid decision-making [5, 6].

At the college level, academic information systems are employed to manage study plans for students, handle assignments from lecturers, manage grades, and generate transcripts. These activities generate data utilized for various academic purposes, such as assessing whether a student is eligible for graduation, evaluating the adequacy of a lecturer's obligations, and addressing other essential matters [7]. Additionally, information on students' learning history proves invaluable for analyzing performance in colleges [8]. The assessment of student learning performance through academic information systems is undoubtedly crucial

Received: Dec. 28, 2022; received in revised form: May 05, 2023; accepted: May 05, 2023; available online: April 05, 2024.
*Corresponding Author

in determining the sustainability and policies of student studies. In essence, academic data play a vital role in maintaining integrity. Furthermore, the integrity of academic data serves as an indicator of the university's commitment to upholding educational standards. The preservation of a university's honor is evident through reliable academic data, safeguarded against unauthorized interventions.

Centralized education data allow universities to access it through a centrally integrated academic information system. As a result, all universities are required to provide academic information systems to facilitate such integration. Many universities in Indonesia use academic information systems developed by third parties. The software development company that has seized the opportunity is a well-known private entity. Among the most popular is XYZ, which frequently furnishes information systems for both public and private universities in Indonesia. XYZ Company develops an academic information system that offers standard modules, addressing the general needs of universities in managing academic data.

Many universities do not independently develop academic information systems due to limited resources. This resource shortage is one factor leading universities in Indonesia to opt for information systems developed by third parties. Moreover, the government's demand for accelerated integration of academic data necessitates more time for universities to develop their academic information systems independently. Utilizing products from software development companies is considered a more effective and efficient approach for universities. Typically, universities serve as end-users of academic systems developed by these software development companies. Academic information system managers in universities are responsible for configuring the system according to their specific academic needs. Currently, there are 36 universities utilizing the academic system produced by the XYZ Company. Despite its widespread use, some risks still need to be uncovered for the university users of the system developed by XYZ Company. In other words, a thorough analysis of the security level of the XYZ company's academic information system is required.

Based on the annual report of Indonesia's National Computer Security Incident Response Team (CSIRT), in 2021, there was a surge in cyber attacks across various sectors. Among the affected sectors, the academic sector, comprising universities, was hit the hardest with 2,217 cases, surpassing the number of attacks that other government and non-government sectors experienced. The private sector was also severely impacted, with 1,483 cases, while the local and central governments suffered 1,097 and 477 cases, respectively. Other sectors experienced attacks of less than 250 cases during the period (see Fig. 1) [9]. Although these incidents are confined to web defacement, such cases can result from inadequate monitoring and security measures for data on the system. The insufficient security of the academic system in Indonesia creates a vulnerability, allowing hackers to easily modify or even steal educational data [10]. This issue poses a severe threat to higher education, necessitating an assessment of the security level in the academic system and the implementation of appropriate solutions [11].

No prior research has investigated the vulnerability of academic information systems developed by XYZ Company. Vulnerability testing involves a security evaluation of information systems connected to the Internet, applications, and their components, aiming to detect vulnerabilities and enhance security management, particularly in universities [12]. The research is crucial to preemptively address vulnerabilities in academic information systems and ensure system integrity. Additionally, the findings from the research can offer insights into how universities address vulnerability issues within XYZ's academic system, providing valuable recommendations. These recommendations are poised to aid the internal system development team at the college in efficiently addressing the discovered vulnerabilities [13].

So far, limited research has been conducted on the safety of the academic system in higher education. Previous research indicates that Indonesia's educational system requires enhanced security measures [14, 15]. It is crucial to perform penetration testing to assess the vulnerability of the academic system, particularly in light of recent events and threats to higher education institutions in Indonesia [16]. System security also aims to mitigate the risk of user errors. In addition to identifying system weaknesses, vulnerability tests can uncover gaps that may be exploited due to user errors. The system is generally designed to anticipate and minimize mistakes caused by users. Many errors have the potential to arise from user actions, such as neglecting to change passwords regularly, using the same password for multiple accounts, and sharing sensitive financial account information [4, 17–19]. System vulnerabilities not only result in personal losses for users but also jeopardize the integrity of system security, particularly within higher education institutions.

The research seeks to evaluate the vulnerability level of academic information systems provided by XYZ Company through penetration testing. The research presents a quantitative assessment of vulnerability in XYZ's academic information system through penetration testing [20]. Penetration testing is conducted using

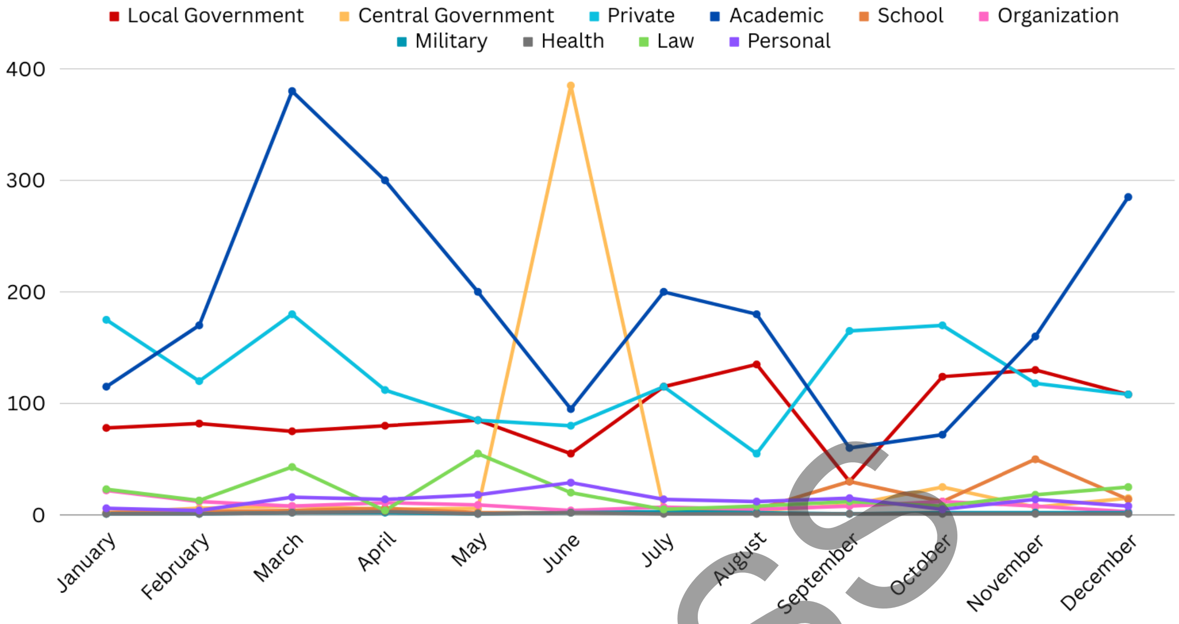


Fig. 1. Sectors affected by website attacks [9].

the Penetration Testing Execution Standard (PTES) method [4, 21], which is based on threats outlined in documents, such as the top 10 of Open Web Application Security Project (OWASP) in 2021 [17, 18, 22–25].

II. Research Method

The research applies a quantitative method to assess the vulnerability of XYZ's academic information system and calculate its vulnerability level. The quantitative representation can serve as a reference for higher education institutions in prioritizing the handling of system vulnerabilities [11, 26–28]. The research focuses on an academic information system provided by the XYZ Company, which 36 universities in Indonesia utilize. Data are gathered through Focus Group Discussions (FGDs) with system managers, observation of server and system specifications, documentation, and literature studies based on the applied framework. FGDs are utilized as part of ethical hacking to enhance institutional performance by addressing system vulnerabilities [12, 18, 29, 30]. The Information System Development Center grants permission for penetration testing at one university.

The Information System Security Assessment Framework (ISSAF), the Open-Source Security Testing Methodology Manual (OSSTMM), OWASP, the Penetration Testing Framework 0.59 (PTF), and PTES are just a few techniques for conducting penetration tests [31, 32]. In the research, PTES and OWASP are chosen as the preferred approaches due to their trustworthiness, ease of use, and support by penetration testing tools. Table I provides an explanation of why these specific techniques are considered superior to alternative methods.

Penetration testing using the PTES method involves seven steps, commencing with the initial phase of pre-engagement interaction, followed by information collection, threat modeling, vulnerability analysis, exploitation, post-exploitation, and concluded by the reporting stage (see Fig. 2). The pre-engagement interaction stage is defined by the utilization of testing tools. The research employs five software tools: Whatweb, Dirsearch, Hashes, Burp Suite, and Sqlmap.

Such tools are employed to assist in the subsequent stage. Data are gathered during the intelligence gathering stage, which involves a set of processes to acquire information about system vulnerabilities through footprint, tool assistance, and pen-tester intelligence. Information concerning XYZ's academic information system is collected using the Whatweb and Dirsearch tools. Whatweb facilitates understanding the details of a website by scanning ports and checking the web interface corresponding [33]. Dirsearch, on the other hand, reveals that web paths are valuable in mapping the potential paths of attacks [34]. Each student and lecturer possesses an individual account of the academic information system. Hence, everyone bears the responsibility for securing data on every account they hold. The account's identity should be

TABLE I
COMPARISON OF PENETRATION TESTING METHODS.

Penetration Testing Methods	Strengths	Weakness
Information System Security Assessment Framework (ISSAF)	ISSAF can be used to conduct tests for online application security, switch and router security, Virtual Private Network (VPN) security, password security, and firewall security.	Since 2006, ISSAF has not received any maintenance or updates.
Open-Source Security Testing Methodology Manual (OSSTMM)	OSSTMM includes trust measures that facilitate risk evaluation for additional and untestable factors.	OSSTMM requires a paid membership.
Open Web Application Security Project (OWASP)	OWASP not only offers a testing methodology but also provides numerous tools and guidance for cybersecurity. It is also an open-source platform. Additionally, OWASP offers a guide for scoring security tests.	Pen-testers should have appropriate knowledge and know how to use the tools when testing in the field.
Penetration Testing Framework 0.59 (PTF)	PTF encompasses various security tests, including password cracking, wireless penetration testing, network footprinting, and physical security.	PTF consists of many stages, with at least 15 sections. However, PTF lacks structure and no longer receives updates.
Penetration Testing Execution Standard (PTES)	PTES leverages other resources, allowing the integration of other frameworks, such as OWASP, which is recommended for testing web applications.	The last update to the PTES document was in April 2012.

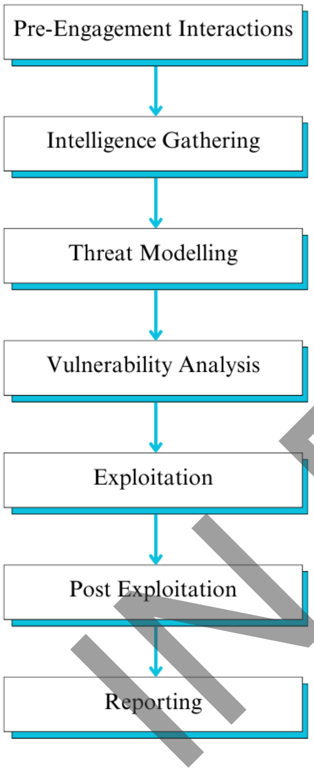


Fig. 2. Penetration Testing Execution Standard (PTES) stages.

TABLE II
TOP 10 OF OPEN WEB APPLICATION SECURITY PROJECT (OWASP) IN 2021.

OWASP Rank	OWASP Vulnerability
1	A01:2021-Broken Access Control
2	A02:2021-Cryptographic Failures
3	A03:2021-Injection
4	A04:2021-Insecure Design
5	A05:2021-Security Misconfiguration
6	A06:2021-Vulnerable and Outdated Components
7	A07:2021-Identification and Authentication Failures
8	A08:2021-Software and Data Integrity Failures
9	A09:2021-Security Logging and Monitoring Failures
10	A10:2021-Server-Side Request Forgery

kept secure and not shared with anyone. The research utilizes the Hashes tool to ascertain the username and password during the penetration testing process. The pen-tester [35] can crack password hashes with a hashes tool [17].

Penetration testing employs a set of tools within the Burp Suite. System vulnerabilities can be assessed using Burp Suite, specifically targeting vulnerabilities outlined in the OWASP’s top-10 vulnerabilities [30]. Moreover, Sqlmap is employed for automated penetration testing with Structured Query Language (SQL) Injection. It is a code injection technique targeting vulnerabilities found in system databases [36–38]. The pen-tester can gain access to the database and manipulate or delete data without authentication.

Thread modeling is conducted based on the collected information. Threat modeling offers a systematic approach to secure a software design, encompasses an understanding of the adversary’s objectives in attacking the system, and considers the assets within the system [39–41]. The thread modeling focuses on ten system vulnerabilities outlined in the OWASP in 2021, as illustrated in Table II. The OWASP’s 2021 top ten identifies the ten most critical security risks [17, 18, 22, 23]. OWASP serves as the industry standard for addressing system security risks. Penetration testing can apply the ten risks identified by OWASP to conduct vulnerability analysis on academic information systems within higher education organizations [42]. OWASP’s classification of these risks is based on four main factors: weakness prevalence, detection capability, exploitation, and technical impact factors [23].

Based on the vulnerability analysis results, a scheme for exploiting XYZ’s academic information system has

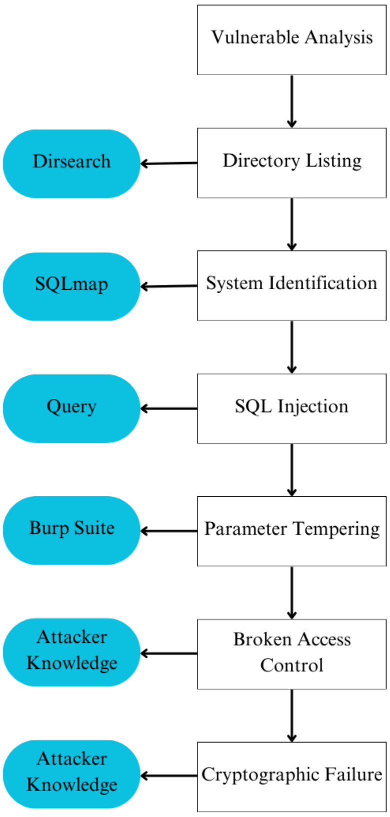
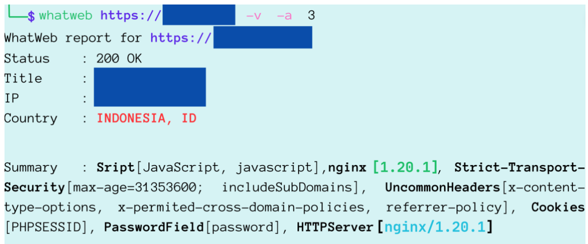


Fig. 3. System exploitation scheme.



```
└─$ whatweb https://[redacted]  -v  -a  3
WhatWeb report for https://[redacted]
Status    : 200 OK
Title     : [redacted]
IP        :
Country   : INDONESIA, ID


Summary   : Sript[JavaScript, javascript],nginx [1.20.1], Strict-Transport-
Security[max-age=31353600;  includeSubDomains],  UncommonHeaders[x-content-
type-options,  x-permited-cross-domain-policies,  referrer-policy],  Cookies
[PHPSESSID], PasswordField[password], HTTPServer[nginx/1.20.1]
```

Fig. 4. Detail of academic system information in XYZ Company.

been devised, as illustrated in Fig. 3. The exploitation process initiates with a directory listing using the Dirsearch tool. Identifying the system maps through directory information guides the penetration tester in executing the exploit. Then, Sqlmap is employed to ensure the validity of the vulnerability analysis and its susceptibility to SQL Injection. The SQL Injection phase involves entering specific queries. Prior to executing SQL injection, a thorough examination of vulnerabilities in a particular input form is also conducted. Tampering with parameters is a form of attack that focuses on programming logic. The Burp Suite tool is utilized to identify parameters used as keys in various processes, such as input, delete, insert, and others [27, 30]. Then, broken access control and cryptographic failures are exploited by leveraging the knowledge of the penetration tester [43].

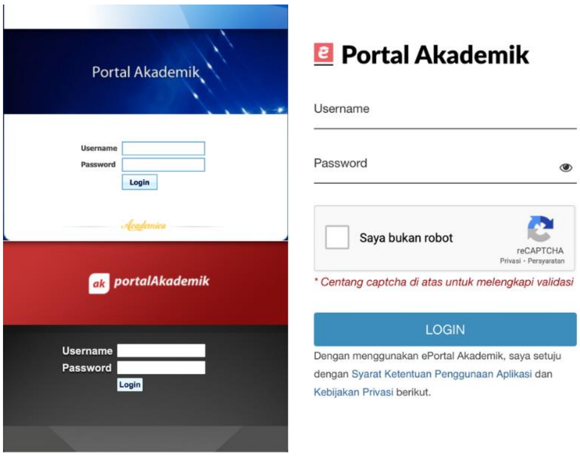
The discovered vulnerability levels are subsequently calculated using the Common Vulnerability Scoring System (CVSS). This method captures crucial characteristics of software, hardware, and firmware vulnerabilities [23]. CVSS comprises three metric groups: base, temporal, and environmental [44, 45]. The base matrix assesses the vulnerability level of a system. Base matrix is most frequently published as it remains constant over time [33].

Next, the output comprises numerical indicators reflecting the severity of vulnerabilities within the academic information system. Utilizing the values presented by CVSS, the vulnerability level is categorized into five classes: none, low, medium, high, and critical [42, 46]. Recommendations for mitigating system vulnerabilities are formulated according to risk parameters, which are instrumental in gauging the vulnerability level [47]. Quantitative estimates pertaining to vulnerability levels are derived from reports generated by CVSS. It is advisable to prioritize addressing vulnerabilities in academic information systems for higher education institutions, taking into account XYZ's corporate stakeholders. A thorough risk assessment is imperative following the completion of the penetration test to effectively manage risks [48].

III. Results and Discussion

The information collection stage is crucial for gathering the system details required to create the scenario. Scenario determination aims to restrict testing to identify vulnerabilities [49]. Information gathering is a stage that offers an overview for the tester to analyze system vulnerability gaps [4, 21]. Figure 4 presents details of XYZ's academic system information obtained through Whatweb tools. This process is a component of intelligence gathering.

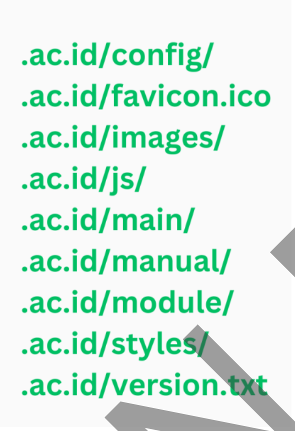
The XYZ Company develops an academic information system using a specific PHP framework they have created and consistently updated. Consequently, the XYZ information system has multiple versions, a subset of which is highlighted on the login page (see Fig. 5). Each higher education institution is required to pay for updates. However, according to the research findings, the updated version has minimal impact on system security. Consequently, the data are collected and utilized as information, and the scenario is formulated in this phase. With the created scenario, it is anticipated that the testing will stay within the problem boundary.



* Centang captcha di atas untuk melengkapi validasi

Dengan menggunakan ePortal Akademik, saya setuju dengan Syarat Ketentuan Penggunaan Aplikasi dan Kebijakan Privasi berikut.

Fig. 5. Login page in XYZ’s product. It has statements: *saya bukan robot* (I am not a robot), *centang captcha di atas untuk melengkapi validasi* (check the captcha above to complete validation), and *dengan menggunakan ePortal Akademik, saya setuju dengan syarat ketentuan penggunaan aplikasi dan kebijakan privasi berikut* (by using ePortal Akademik, I agree with the term of the application and privacy policy).



```
.ac.id/config/
.ac.id/favicon.ico
.ac.id/images/
.ac.id/js/
.ac.id/main/
.ac.id/manual/
.ac.id/module/
.ac.id/styles/
.ac.id/version.txt
```

Fig. 6. Directory structure in XYZ.

At the vulnerability stage, port-based network scanning, web vulnerability scanning, and directory listings are performed. The pen-tester conducts directory listing and brute-force attacks using the Dirsearch tool, which functions to reveal the directories on the website and expand the testing area. Based on observations of XYZ’s academic system, the directory structure reveals that through the university website, the users tend to show similarities in directory config, module, or main, and some still display file version.txt. Figure 6 displays the results obtained from web path discovery using the Dirsearch tool.

TABLE III
STRUCTURED QUERY LANGUAGE (SQL) INJECTION IN LOGIN PAGE.

SQL Injection – Login		
	Matric	Value
Common Vulnerability Scoring System (CVSS) with a base score of 7.4 (high)	Attack vector	Network
	Attack complexity	Low
	Privileges required	None
	User interaction	None
	Scope	Unchanged
	Confidentiality	High
	Integrity	None
	Availability	None
Vulnerability	Allow to get all the information that is on the database	
Path	/index.php?pModule=zdKbnKU=&pSub=zdKbnKU=&pAct=0dWjppy lusername=test&password=test	
Recommendation Apply a filter to the username input field in the Login page before querying the database: *mysqli_real_escape_string()*		

The CVSS method is an approach to measuring system risk and vulnerability. The research opts for CVSS because it is supported by a dataset containing numerous vulnerabilities [50, 51]. CVSS has also been widely used to measure penetration testing results due to its easy scoring. There are several other methods for assessing vulnerabilities, such as Computer Emergency Response Team/Coordination Center (CERT/CC) [52, 53], X-Force IBM [53, 54], and Vulnerability Rating and Scoring System (VRSS) [55]. Table A1 in Appendix shows why CVSS is considered superior to the other scoring methods.

Penetration testing reveals eight vulnerabilities in XYZ’s academic information system. The vulnerabilities and their corresponding risk levels based on CVSS are presented in Tables III through X. The first vulnerability discovered is SQL Injection on the Login page. As seen in Table III, in the path column, the highlighted code is a parameter with potential vulnerabilities for SQL Injection. The recommended mitigation is to apply filters on parameters with vulnerabilities in usernames. For example, the username parameter is protected by the code *mysqli_real_escape_string()* to anticipate any input query in the username field on the login page. The penetration tester can retrieve the login information of an account from the data found in the database. A vulnerability on the Login page results in a severity score of 7.4, categorizing it as a high priority. These vulnerabilities are a priority to be addressed by higher education institutions.

Table IV illustrates SQL Injection vulnerabilities identified at various paths on the Search Contacts page. The parameters susceptible to this vulnerability are *searchNiuKey* and *searchNamaKey*. Enhancing the security of the Search Contacts column by implementing the *mysqli_real_escape_string()* code to address this issue is recommended. Similar to the Login page, the

TABLE IV
STRUCTURED QUERY LANGUAGE (SQL) INJECTION ON THE SEARCH CONTACT PAGE.

SQL Injection – Search Contacts		
	Matric	Value
Common Vulnerability Scoring System (CVSS) with a base score of 7.1 (high)	Attack vector	Network
	Attack complexity	Low
	Privileges required	Low
	User interaction	None
	Scope	Unchanged
	Confidentiality	High
	Integrity	Low
	Availability	None
Vulnerability	Allow to get all the information that is on the database	
Path	/index.php?pModule=zsinppiZmQ==&pSub=wseYpZylp8WTpdSk&pAct=wseYchk[niu]=niu&searchNiuKey=test&chk[nama]=nama&searchNamaKey=test&prodi=all&searchBtn=Cari	
Recommendation		
Before querying the database, apply a filter to the Contact Search input field using *mysqli_real_escape_string()*		

TABLE V
CRYPTOGRAPHIC ISSUE.

Cryptographic Issues		
	Matric	Value
Common Vulnerability Scoring System (CVSS) with base score of 5.3 (medium)	Attack vector	Network
	Attack complexity	Low
	Privileges required	None
	User interaction	None
	Scope	Unchanged
	Confidentiality	Low
	Integrity	None
	Availability	None
Vulnerability	Allow to uncover encrypted text information	
Path	/index.php?pModule=zdKbnKU=&pSub=yMSblKORoNWYn9M=&pAct=18yZqg==&errMsg=ttaZpVeAldOWVsmaoFSBxaysp6Okm4HWoMeZ0VSok6TKxw==	
Recommendation		
Enhance the encryption method. Strongly discourage the use of the *errMsg* parameter for sending error messages. Capture and temporarily store response error messages using the session.		

Search Contacts page exhibits a high vulnerability rate of 7.1.

In Table V, the vulnerabilities identified in the highlighted path can be exploited by the penetration tester to acquire encrypted data information. The presence of cryptographic issues in XYZ’s academic system stems from the relatively weak data encryption employed. The penetration tester can deduce the encryption pattern by analyzing the Nomor Induk Mahasiswa (NIM or student ID) utilized at specific universities. Exploiting vulnerabilities within this encryption aspect allows the penetration tester to unveil additional weaknesses and exploit them further [56, 57]. The CVSS value denotes a moderate level of vulnerability in cryptographic issues, registering at 5.3 on the scale.

TABLE VI
BROKEN ACCESS CONTROL IN THE ACADEMIC ADVISOR’S STUDENT PAGE.

Broken Access Control–Students’ Information		
	Matric	Value
Common Vulnerability Scoring System (CVSS) with base score of 4.4 (medium)	Attack vector	Network
	Attack complexity	High
	Privileges required	High
	User interaction	None
	Scope	Unchanged
	Confidentiality	High
	Integrity	None
	Availability	None
Vulnerability	Allow to disclose students’ information (Profile, forms of study plan, course report, grade history, and transcripts)	
Path	/index.php?pModule=xdKnmKU=&pSub=zsiip6akmcqQqdmulpmf2Kw=&pAct=18yZqg== /index.php?pModule=1taZpQ==&pSub=0dWjmaCemQ==&pAct=18yZqg==&niu=qJR1Y2hqZJlp&view=xdKnmKU=&sia=xxx	
Recommendation		
Apply role-checking to each menu. Highly use a stronger encryption method in the *niu* parameter.		

TABLE VII
TAMPERING PARAMETERS ON MESSAGE DELIVERY.

Tampering Parameter–Message Delivery		
	Matric	Value
Common Vulnerability Scoring System (CVSS) with a base score of 4.3 (medium)	Attack vector	Network
	Attack complexity	Low
	Privileges required	Low
	User interaction	None
	Scope	Unchanged
	Confidentiality	Low
	Integrity	None
	Availability	None
Vulnerability	Manipulate the sender and recipient of the message.	
Path	/index.php?pModule=zsinppiZmQ==&pSub=zsinppiZmQ==&pAct=0dWjlpylpw==data[MessageSender]=NIM32&data[MessageReceiver]=NIM12&data[MessageTitle]=test&data[MessageContent]=123&act=doCompose&compBtn=Kirim	
Recommendation		
In the POST method, NIM can be replaced by retrieving the NIM from the user session. Enhancing the encryption of the NIM for both the sender and recipient of the message is highly recommended.		

The vulnerability details presented in Table VI are susceptible to discovery and exploitation by penetration testers, primarily owing to the inadequate data encryption utilized by XYZ’s products. The ‘*niu*’ parameter represents encrypted NIM data, which hackers can manipulate, leading to the inadvertent disclosure of students’ personal information as a targeted outcome based on NIM. Vulnerabilities in student information are categorized at a medium severity level.

Students can engage with text messages via XYZ’s academic information system. However, vulnerabilities have been identified that can lead to misusing infor-

TABLE VIII
TAMPERING PARAMETER ON FORMS OF STUDY PLAN.

Tampering Parameters–Modifying Study Plan		
	Matric	Value
Common Vulnerability Scoring System (CVSS) with a base score of 5.3 (medium)	Attack vector	Network
	Attack complexity	High
	Privileges required	Low
	User interaction	None
	Scope	Unchanged
	Confidentiality	None
	Integrity	High
	Availability	None
Vulnerability	Input form of study plan as another student.	
Path	/index.php?pModule=wsaVl5yfncmQptGaoA==&pSub=wsaVl5yfncmQptGaoA==&pAct=0dWjlpylpw==&niu=qJR1Y2hqZJln&prodi=k5hlZGtkkodeMkul%5B%5D=k5ZkY2dma59psppuYmZmlm5tabCJs5XehK6Nk2hiYg%3D%3D&act=addKrs&btnAdd=Tambah	
Recommendation		
NIM can be replaced by getting NIM from the session user. Strong encryption on NIM is highly recommended.		

TABLE IX
TAMPERING PARAMETER ON COURSE GRADE MANAGEMENT.

Parameter Tampering–Course Grade Management		
	Matric	Value
Common Vulnerability Scoring System (CVSS) with a base score of 4.4 (medium)	Attack vector	Network
	Attack complexity	High
	Privileges required	High
	User interaction	None
	Scope	Unchanged
	Confidentiality	None
	Integrity	High
	Availability	None
Vulnerability	Make changes to the grades of the previous semester.	
Path	/index.php?pModule=xdKnmKU=&pSub=yNWVl5yRocefl8yen5mf2A==&pAct=18yZqg==&kelas=k5hlZGtkZJZhbZ1uag==&smt=k5NlbGg=&sia=xxx	
Recommendation		
Do filter or whitelist on the *smt* parameter.		

TABLE X
BROKEN ACCESS CONTROL ON STUDY PLAN, COURSE REPORT, AND TRANSCRIPTS.

Broken Access Control–Study Plan, Course Report, and Transcripts		
	Matric	Value
Common Vulnerability Scoring System (CVSS) with a base score of 3.1 (low)	Attack vector	Network
	Attack complexity	High
	Privileges required	Low
	User interaction	None
	Scope	Unchanged
	Confidentiality	Low
	Integrity	None
	Availability	None
Vulnerability	Allow to disclose other students' information (forms of study plan, course report, and transcripts)	
Path	/index.php?pModule=wsaVl5yfncmQqMqpoaal&pSub=wsaVl5yfncmQqMqpoaal&pAct=0dWdoas=&niu=qJR1Y2hqZJln&prodi=k5hlZGtk&sem=k5hlZGtkZJZhaJVuYw==	
Recommendation		
Apply role-checking to each menu. Use strong encryption on *niu* parameters.		

mation in these messages. The specific vulnerability details presented in Table VII enable hackers to manipulate both message recipients and senders within XYZ's academic portal. It tampers the parameters of the message sender *data[MessageSender]* and *recipient data[MessageReceiver]*. Furthermore, not only the identity of the sender and recipient of the message but also the message information can be misused [39, 58]. The message content itself is also susceptible to exploitation. Tests conducted in this section reveal a CVSS value of 4.3, indicating a medium severity level.

Modifying the study plan by adding or removing courses while pretending to be someone else constitutes data misuse. It can undoubtedly harm students rather than just being a prank. Table VIII illustrates that this vulnerability can manipulate a student's study plan. By tampering with the niu parameter as NIM, all parameters become encrypted, limiting access to study plans based on the study program (*prodi*) and course code (*kodeMkul[]*). This test receives a reasonably high vulnerability score of 5.3, even though it is classified as a medium level.

Table IX illustrates vulnerabilities in course grade management that can be exploited by manipulating the semester code within the *smt* parameter. A limitation of *smt* parameters is the reliance on a filter or whitelist, which confines semester access exclusively to the current (active) semester [28, 59]. Consequently, lecturers can only input or modify grades for the ongoing semester. However, this vulnerability permits a lecturer to substitute a student's grades from a previous semester, posing a significant risk, particularly if an attacker gains access to a lecturer account (potentially through a SQL Injection vulnerability). Rated at 4.4, this vulnerability is considered medium, yet the misuse of lecturer accounts can result in defamation of lecturers.

Learning outcomes are sometimes private information for students. The vulnerabilities identified in the research can potentially expose such personal data to unauthorized individuals. The vulnerabilities outlined in Table X can reveal various forms of study plans, course reports, or transcripts of other students by manipulating the *niu* parameter. Although this vulnerability is rated at only 4.4 and considered medium risk, the misuse of lecturer accounts can result in slander of that lecturer.

The limited capabilities of human resources and the complexity of the academic system pose challenges to the independent development of academic information systems. The research uncovers vulnerabilities that have never been disclosed, highlighting the difficulty in securing academic information systems. Identifying security flaws in each system is imperative [60, 61]. Then, safeguarding the academic system is as crucial as ensuring the security of financial systems. Security concerns demand serious, prompt, and optimal attention. Moreover, many universities are content with the current state, managing academic information systems as end users. Public and private universities may opt for third-party services due to their effectiveness and efficiency. However, the lack of awareness about data and information security within the academic system user community often leads to serious consideration only after a hacker attack occurs. The research findings are crucial in anticipating vulnerabilities in XYZ's academic information system.

IV. Conclusion

Based on penetration testing results on the academic information system of XYZ's products, at least eight vulnerabilities are identified. These comprised two high-level, five mid-level, and one low-level vulnerabilities. The vulnerabilities included SQL Injection, which has the potential to manipulate and delete databases, allowing unauthorized access to related databases. Recommendations to address SQL Injection involve securing input data, implementing validation, applying specific filters, or employing input whitelisting. Broken access control can be mitigated by scrutinizing sessions or roles to prevent unauthorized access. The interference parameters in the system, notably the issue of weak encryption, are a significant concern as they enable attackers to guess the encryption.

The research highlights substantial problems in the encryption method's weakness, which can lead to unauthorized exploration and misuse of sensitive academic data, even beyond the system. The 36 universities can enhance data integrity by autonomously addressing vulnerabilities uncovered in the research. Alternatively, they can leverage third-party security technologies, such as firewalls, to fortify their security infrastructure. Additionally, universities are urged to promote awareness among the academic community regarding the security of academic data. The scope of the research is limited to analyzing the security in XYZ's academic system. It should be noted that other academic systems may pose unique vulnerabilities and risks. While OWASP and PTES have primarily emphasized web application security, a more comprehensive approach is required to address network and infrastructure security concerns. The inflexible nature of these methodologies can hinder their effectiveness in dealing with complex security testing scenarios that require flexibility and agility.

Furthermore, there is a need to refine the penetration testing method, particularly in the information-gathering section, which demands the knowledge and experience of the pen-tester. Penetration testing requires a thorough comprehension of the hazards present in the system under examination. This test involves recognizing important assets, vulnerabilities susceptible to exploitation by attackers, and other possible risks. Regular evaluation of academic systems' security is essential to safeguard against emerging assaults and vulnerabilities.

Acknowledgement

The authors express their gratitude to the Faculty of Engineering and the University of Bengkulu for their invaluable support and resources throughout the research, which aided in the development and execution of the research.

Author Contribution

Writing—original draft, F. U., and R. N.; Methodology, F. U.; Formal analysis, F. U., and R. N.; Pentester, R. N.; Analysis result review, F. U. All authors have read and agreed to the published version of the manuscript.

References

[1] M. Kim and D. Kim, "A suggestion on the LDA-based topic modeling technique based on ElasticSearch for indexing academic research results," *Applied Sciences*, vol. 12, no. 6, pp. 1–10, 2022.

[2] A. Reis, P. Martins, J. Borges, A. Sousa, T. Rocha, and J. Barroso, "Supporting accessibility in higher education information systems," in *Universal Access in Human–Computer Interaction. Design and Development Approaches and Methods: 11th International Conference, UAHCI 2017*. Springer, 2017, pp. 227–237.

[3] R. Bruzgiene and K. Jurgilas, "Securing remote access to information systems of critical infrastructure using two-factor authentication," *Electronics*, vol. 10, no. 15, pp. 1–18, 2021.

[4] A. I. Kusumarini and H. B. Seta, "Information system security analysis to determine server security vulnerability with Penetration Testing Execution Standard (PTES) method at VWX University," in *2021 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS*. IEEE, 2021, pp. 7–12.

[5] Q. Dai, “Designing an accounting information management system using big data and cloud technology,” *Scientific Programming*, vol. 2022, pp. 1–11, 2022.

[6] H. Lu, Y. Zhu, Q. Lin, T. Wang, Z. Niu, and E. Herrera-Viedma, “Heterogeneous knowledge learning of predictive academic intelligence in transportation,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 4, pp. 3737–3755, 2020.

[7] I. P. A. Sudrastawa and K. Y. E. Ayanto, “Sensitive personal data publication on higher education information system websites in Indonesia,” in *2019 2nd International Conference of Computer and Informatics Engineering (IC2IE)*. IEEE, 2019, pp. 93–98.

[8] B. K. Yousafzai, S. A. Khan, T. Rahman, I. Khan, I. Ullah, A. Ur Rehman, M. Baz, H. Hamam, and O. Cheikhrouhou, “Student-performulator: Student academic performance using hybrid deep neural network,” *Sustainability*, vol. 13, no. 17, pp. 1–21, 2021.

[9] Direktorat Operasi Keamanan Siber Badan Siber dan Sandi Negara, “Laporan Tahunan Monitoring Keamanan Siber 2021,” 2021. [Online]. Available: https://cloud.bssn.go.id/s/Lx8Ry3w2Ew3NJa7

[10] A. B. Cengiz, G. Kalem, and P. S. Boluk, “The effect of social media user behaviors on security and privacy threats,” *IEEE Access*, vol. 10, pp. 57 674–57 684, 2022.

[11] S. Zheng, Y. Wu, S. Wang, Y. Wei, D. Mu, H. He, D. Han, J. Liao, and H. Chen, “PTVis: Visual narrative and auxiliary decision to assist in comprehending the penetration testing process,” *IEEE Access*, vol. 8, pp. 194 523–194 540, 2020.

[12] 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,” *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 592–603, 2021.

[13] G. Canfora, A. Di Sorbo, S. Forootani, A. Pirozzi, and C. A. Visaggio, “Investigating the vulnerability fixing process in OSS projects: Peculiarities and challenges,” *Computers & Security*, vol. 99, 2020.

[14] Candiwan, P. K. Sari, and N. Nurshabrina, “Assessment of information security management on indonesian higher education institutions,” in *Advanced Computer and Communication Engineering Technology: Proceedings of ICOCOE 2015*. Springer International Publishing, 2016, pp. 375–385.

[15] I. G. N. Mantra, M. S. Hartawan, H. Saragih, and A. Abd Rahman, “Web vulnerability assessment and maturity model analysis on indonesia higher education,” *Procedia Computer Science*, vol. 161, pp. 1165–1172, 2019.

[16] A. Almaarif and M. Lubis, “Vulnerability Assessment and Penetration Testing (VAPT) framework: Case study of government’s website,” *International Journal on Advanced Science Engineering and Information Technology*, vol. 10, no. 5, pp. 1874–1880, 2020.

[17] D. Kellezi, C. Boegelund, and W. Meng, “Securing open banking with model-view-controller architecture and OWASP,” *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–13, 2021.

[18] B. Ksiezopolski, K. Mazur, M. Miskiewicz, and D. Rusinek, “Teaching a hands-on CTF-based web application security course,” *Electronics*, vol. 11, no. 21, pp. 1–21, 2022.

[19] K. B. Jalbani, M. Yousaf, M. S. Sarfraz, R. Jamili Oskouei, A. Hussain, and Z. Memon, “Poor coding leads to dos attack and security issues in web applications for sensors,” *Security and Communication Networks*, vol. 2021, pp. 1–11, 2021.

[20] S. Ramanauskaitė, N. Urbonaitė, Š. Grigaliūnas, S. Preidys, V. Trinkūnas, and A. Venčkauskas, “Educational organization’s security level estimation model,” *Applied Sciences*, vol. 11, no. 17, pp. 1–19, 2021.

[21] F. Z. Lidanta, A. Almaarif, and A. Budiyono, “Vulnerability analysis of wireless LAN networks using penetration testing execution standard: A case study of cafes in Palembang,” in *2021 International Conference on ICT for Smart Society (ICISS)*. IEEE, 2021, pp. 1–5.

[22] J. R. B. Higuera, J. B. Higuera, J. A. S. Montalvo, J. C. Villalba, and J. J. N. Pérez, “Benchmarking approach to compare web applications static analysis tools detecting OWASP top ten security vulnerabilities,” *Computers, Materials & Continua*, vol. 64, no. 3, pp. 1555–1577, 2020.

[23] J. Li, “Vulnerabilities mapping based on OWASP-SANS: A survey for Static Application Security Testing (SAST),” *Annals of Emerging Technologies in Computing (AETiC)*, vol. 4, no. 3, pp. 1–5, 2020.

[24] F. M. Tudela, J.-R. Bermejo Higuera, J. Bermejo Higuera, J.-A. Sicilia Montalvo, and M. I. Argyros, “On combining static, dynamic and interactive analysis security testing

tools to improve OWASP top ten security vulnerability detection in web applications,” *Applied Sciences*, vol. 10, no. 24, pp. 1–24, 2020.

[25] A. Bhardwaj, S. B. H. Shah, A. Shankar, M. Alazab, M. Kumar, and T. R. Gadekallu, “Penetration testing framework for smart contract blockchain,” *Peer-to-Peer Networking and Applications*, vol. 14, pp. 2635–2650, 2021.

[26] S. Y. Enoch, Z. Huang, C. Y. Moon, D. Lee, M. K. Ahn, and D. S. Kim, “HARMer: Cyber-attacks automation and evaluation,” *IEEE Access*, vol. 8, pp. 129 397–129 414, 2020.

[27] J. Shahid, M. K. Hameed, I. T. Javed, K. N. Qureshi, M. Ali, and N. Crespi, “A comparative study of web application security parameters: Current trends and future directions,” *Applied Sciences*, vol. 12, no. 8, pp. 1–23, 2022.

[28] S. Ibarra-Fiallos, J. B. Higuera, M. Intriago-Pazmiño, J. R. B. Higuera, J. A. S. Montalvo, and J. Cubo, “Effective filter for common injection attacks in online web applications,” *IEEE Access*, vol. 9, pp. 10 378–10 391, 2021.

[29] M. N. Zakaria, P. A. Phin, N. Mohmad, S. A. Ismail, M. N. Kama, and O. Yusop, “A review of standardization for penetration testing reports and documents,” in *2019 6th International Conference on Research and Innovation in Information Systems (ICRIIS)*. IEEE, 2019, pp. 1–5.

[30] M. Albahar, D. Alansari, and A. Jurcut, “An empirical comparison of pen-testing tools for detecting web app vulnerabilities,” *Electronics*, vol. 11, no. 19, pp. 1–25, 2022.

[31] A. Shanley and M. N. Johnstone, “Selection of penetration testing methodologies: A comparison and evaluation,” in *13th Australian Information Security Management Conference*, 2015, pp. 65–72.

[32] A. Shanley, “Penetration testing frameworks and methodologies: A comparison and evaluation,” Master’s thesis, School of Science, Edith Cowan University, 2016.

[33] R. Akhilesh, O. Bills, N. Chilamkurti, and M. J. M. Chowdhury, “Automated penetration testing framework for smart-home-based IoT devices,” *Future Internet*, vol. 14, no. 10, pp. 1–18, 2022.

[34] S. Zhou, J. Liu, D. Hou, X. Zhong, and Y. Zhang, “Autonomous penetration testing based on improved deep Q-network,” *Applied Sciences*, vol. 11, no. 19, pp. 1–15, 2021.

[35] Sufatrio, J. Vykopal, and E. C. Chang, “Collaborative paradigm of teaching penetration testing using real-world university applications,” in *Proceedings of the 24th Australasian Computing Education Conference*, 2022, pp. 114–122.

[36] M. Alenezi, M. Nadeem, and R. Asif, “SQL injection attacks countermeasures assessments,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 2, pp. 1121–1131, 2021.

[37] M. Liu, K. Li, and T. Chen, “DeepSQLi: Deep semantic learning for testing SQL injection,” in *Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2020, pp. 286–297.

[38] A. Valenza, L. Demetrio, G. Costa, and G. Lagorio, “WAF-A-MoLE: An adversarial tool for assessing ML-based WAFs,” *SoftwareX*, vol. 11, pp. 1–4, 2020.

[39] A. Tedyyana, F. Ratnawati, E. Syam, and F. P. Putra, “Threat modeling in application security planning citizen service complaints,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 28, no. 2, pp. 1020–1027, 2022.

[40] A. Alanda, D. Satria, M. I. Ardhana, A. A. Dahlan, and H. A. Mooduto, “Web application penetration testing using SQL injection attack,” *JOIV: International Journal on Informatics Visualization*, vol. 5, no. 3, pp. 320–326, 2021.

[41] Y. Jiang and Y. Atif, “A selective ensemble model for cognitive cybersecurity analysis,” *Journal of Network and Computer Applications*, vol. 193, pp. 1–16, 2021.

[42] J. Brown, T. Saha, and N. K. Jha, “GRAVITAS: Graphical reticulated attack vectors for Internet-of-things aggregate security,” *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 3, pp. 1331–1348, 2021.

[43] X. Ning and J. Jiang, “In the mind of an insider attacker on cyber-physical systems and how not being fooled,” *IET Cyber-Physical Systems: Theory & Applications*, vol. 5, no. 2, pp. 153–161, 2020.

[44] J. Reyes, W. Fuertes, P. Arévalo, and M. Macas, “An environment-specific prioritization model for information-security vulnerabilities based on risk factor analysis,” *Electronics*, vol. 11, no. 9, pp. 1–24, 2022.

[45] W. Wang, F. Shi, M. Zhang, C. Xu, and J. Zheng, “A vulnerability risk assessment method based on heterogeneous information network,” *IEEE Access*, vol. 8, pp. 148 315–148 330, 2020.

[46] M. A. Hassan, Z. Shukur, and M. Mohd, “A penetration testing on Malaysia popular e-wallets and m-banking apps,” *International Journal of*

Advanced Computer Science and Applications, vol. 13, no. 5, pp. 692–703, 2022.

[47] A. A. Tubis, S. Werbinska-Wojciechowska, M. Góralczyk, A. Wroblewski, and B. Zietek, "Cyber-attacks risk analysis method for different levels of automation of mining processes in mines based on Fuzzy theory use," *Sensors*, vol. 20, no. 24, pp. 1–23, 2020.

[48] M. I. Lunesu, R. Tonelli, L. Marchesi, and M. Marchesi, "Assessing the risk of software development in agile methodologies using simulation," *IEEE Access*, vol. 9, pp. 134 240–134 258, 2021.

[49] Y. Kristiyanto and Ernastuti, "Analysis of deauthentication attack on IEEE 802.11 connectivity based on IoT technology using external penetration test," *CommIT (Communication and Information Technology) Journal*, vol. 14, no. 1, pp. 45–51, 2020.

[50] Y. Jiang and Y. Atif, "An approach to discover and assess vulnerability severity automatically in cyber-physical systems," in *13th International Conference on Security of Information and Networks*, 2020, pp. 1–8.

[51] M. Keramati, "New vulnerability scoring system for dynamic security evaluation," in *2016 8th International Symposium on Telecommunications (IST)*. IEEE, 2016, pp. 746–751.

[52] K. Sridhar, A. Householder, J. Spring, and D. W. Woods, "Cybersecurity information sharing: Analysing an email corpus of coordinated vulnerability disclosure," in *The 20th Annual Workshop on the Economics of Information Security*, 2021, pp. 1–39.

[53] R. Anderson and B. Schneier, "Guest editors' introduction: Economics of information security," *IEEE Security & Privacy*, vol. 3, no. 1, pp. 12–13, 2005.

[54] Q. Liu and Y. Zhang, "VRSS: A new system for rating and scoring vulnerabilities," *Computer Communications*, vol. 34, no. 3, pp. 264–273, 2011.

[55] R. Sharma and R. K. Singh, "An improved scoring system for software vulnerability prioritization," *Quality, IT and Business Operations: Modeling and Optimization*, pp. 33–43, 2018.

[56] S. Shamshad, F. Riaz, R. Riaz, S. S. Rizvi, and S. Abdulla, "An enhanced architecture to resolve public-key cryptographic issues in the Internet of Things (IoT), employing quantum computing supremacy," *Sensors*, vol. 22, no. 21, pp. 1–23, 2022.

[57] A. Fukami, R. Stoykova, and Z. Geradts, "A new model for forensic data extraction from encrypted mobile devices," *Forensic Science International: Digital Investigation*, vol. 38, pp. 1–10, 2021.

[58] S. Bae, S. Gros, and B. Kulcsár, "Can AI abuse personal information in an EV fast-charging market?" *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 8759–8769, 2021.

[59] C. Gupta, R. K. Singh, and A. K. Mohapatra, "GeneMiner: A classification approach for detection of XSS attacks on web services," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–12, 2022.

[60] K. Razikin and A. Widodo, "General cybersecurity maturity assessment model: Best practice to achieve Payment Card Industry-Data Security Standard (PCI-DSS) compliance," *CommIT (Communication and Information Technology) Journal*, vol. 15, no. 2, pp. 91–104, 2021.

[61] E. Mustikawati, D. Perdana, and R. M. Negara, "Network security analysis in VANET against black hole and jellyfish attack with Intrusion Detection System algorithm," *CommIT (Communication and Information Technology) Journal*, vol. 11, no. 2, pp. 77–83, 2017.

Appendix

The Appendix can be seen in the next page.

TABLE A1
COMPARISON OF VULNERABILITY SCORING.

Vulnerability Scoring Methods	Strengths	Weaknesses
Common Vulnerability Scoring System (CVSS)	It is the most commonly used vulnerability scoring method. It provides risk indicators for mitigation. The vulnerability calculator is available, featuring a database that lists vulnerabilities and facilitates easy scoring. It is supported and has been utilized in numerous tests and studies.	The provided calculator only calculates fundamental vulnerability. It is not diversity in limited vulnerability.
Computer Emergency Response Team/Coordination Center (CERT/CC)	It is capable of reporting vulnerabilities, primarily serving companies and vendors. It maintains confidentiality and provides solutions to identified vulnerabilities.	It was popular in the early 2000s. The formula for determining vulnerability is not publicly available.
X-Force IBM	It conducts a quantitative vulnerability assessment with three levels, supported by more than 40,000 databases consisting of vulnerabilities, threats, and security checks.	Instructions on conducting vulnerability assessments are not publicly available.
Vulnerability Rating and Scoring System (VRSS)	It is a method that calculates vulnerability both qualitatively and quantitatively. It evaluates vulnerabilities based on previous steps to determine various vulnerabilities at a certain level.	The vulnerabilities discovered are not time-based. The resulting score is highly biased towards vulnerability.