International Journal Science and Technology
IJST Vol 5 No. 2 | July 2026 | ISSN: 2828-7223 (print), ISSN: 2828-7045 (online), Page 323-336

Security Analysis and Threat Modeling of the Informatics Engineering Laboratory Information System (SILABTI) Using Attack Tree Methodology

Puji Zulaikasari[1], Avinanta Tarigan[2]
[1,2]Information Systems, Faculty of Computer Science and Information Technology, Gunadarma University, Indonesia

Article History
Received : July 17, 2026
Revised : July 31, 2026
Accepted : August 12, 2026
Published : August 14, 2026

Corresponding author*:
puzulaikasari@staff.gunadarma.ac.id

Cite This Article *[APA Style]:*
Zulaikasari, P., & Tarigan, A. (2026). Security Analysis and Threat Modeling of the Informatics Engineering Laboratory Information System (SILABTI) Using Attack Tree Methodology. *International Journal Science and Technology*, *5*(2), 323–336.

DOI:
https://doi.org/10.56127/ijst.v5i2.2090

Abstract: Academic information systems manage sensitive data whose integrity directly affects academic administration and student outcomes. Legacy intranet-based systems may remain vulnerable to both technical attacks and human-factor threats, particularly when outdated software, unencrypted communication, weak access controls, and privileged user access coexist within the same operational environment. **Objective:** This study aims to analyze the security vulnerabilities of the Informatics Engineering Laboratory Information System (SILABTI) and identify potential pathways for unauthorized modification of practicum grades and graduation statuses using a hierarchical Attack Tree approach. **Methodology:** This study employed an analytical systems-engineering design based on Attack Tree threat modeling. The assessment covered the SILABTI web application, local network environment, MySQL database architecture, and administrative workflows. System boundaries and technical conditions were identified through infrastructure assessment and network reconnaissance, followed by hierarchical decomposition of adversarial objectives into root goals, intermediate sub-goals, and technical execution leaf nodes using AND/OR relationships. The resulting attack pathways were qualitatively evaluated according to technical prerequisites, execution complexity, system exposure, and detection probability. **Findings:** The analysis identified four primary intrusion vectors: credential acquisition, application exploitation, database exploitation, and insider exploitation. Four pathways were considered particularly high-risk: local network sniffing, workstation keylogging, automated brute-force attacks, and insider bribery or coercion. These findings indicate that SILABTI's security risks arise from the interaction of legacy software, network communication weaknesses, endpoint privileges, authentication controls, and human factors. **Implications:** The findings support an eight-point defense-in-depth framework incorporating TLS/HTTPS, tamper-resistant audit logging, role-based access control, network access restrictions, stronger authentication controls, workstation privilege restriction, anti-spoofing measures, and institutional security governance. This framework can guide university IT administrators in strengthening legacy academic information systems. **Originality:** This study contributes a hierarchical threat-decomposition model specifically designed for a legacy intranet-based academic laboratory information system, integrating technical and human-layer attack pathways within a single security assessment framework.

Keywords: Attack Tree; academic information system; cybersecurity; threat modeling; vulnerability assessment.

INTRODUCTION

Web-based information systems have become essential infrastructures for operational and administrative fulfillment across multiple sectors, including higher education. The Laboratory of Informatics Engineering at Gunadarma University deploys an intranet web-based application known as SILABTI (Sistem Informasi Laboratorium Teknik Informatika) to manage practicum data, student scheduling, and grading. However, rapid advancements in web-based technology have been accompanied by an escalation in cybersecurity threats and unauthorized intrusion techniques (Tang, 2023). Attackers utilize various methods, such as injection attacks and access control exploits, to infiltrate target networks and manipulate information for motives ranging from personal recognition to economic gain (Al-Shareeda et al., 2022; Li, 2023). Within academic laboratory systems, practicum grades and graduation statuses for both mandatory and supporting courses represent high-value information. Unauthorized modification of these records constitutes an unlawful act and creates significant academic injustice among students.

Previous engineering studies addressing computer and information systems security have evolved across three primary technical domains. First, preventive control and secure software engineering studies have focused on implementing structured recommendations and engineering approaches to build cyber-resilient systems and mitigate software vulnerabilities from the development phase (National Institute of Standards and Technology (National Institute of & Technology, 2021, 2022). Second, detection and security assessment research has emphasized continuous security testing, multi-modal vulnerability detection frameworks, and machine learning approaches to identify anomalies, software weaknesses, and insider threats (Oliveira, 2022). Third, threat modeling and analytical vulnerability assessments have utilized structured methodologies such as formal threat modeling techniques and Attack Trees to systematically map adversary intrusion pathways, integrate vulnerability data with incident response, and evaluate security risks across complex web applications (Piètre-Cambacédès & Bouillon, 2022). Despite extensive literature within these three domains, existing research predominantly examines cloud-native enterprise infrastructures, leaving a notable engineering research gap regarding hierarchical threat modeling for legacy, intranet-based academic laboratory systems where technical software vulnerabilities intersect heavily with human-factor insider threats.

To address this unresolved gap, the primary technical objective of this study is to perform an engineering vulnerability assessment of SILABTI by modeling structured attack trees that decompose the root goal of unauthorized grade manipulation into intermediate sub-goals and leaf-node technical execution steps (Konarski & Ptak, 2020). This study aims to conduct an in-depth technical analysis of the SILABTI system architecture and evaluate its structural resilience under defined operational boundary conditions and security threats. To achieve this, the research develops an Attack Tree model that hierarchically decomposes unauthorized intrusion goals from the root objective of altering practicum grades down to specific technical execution leaf nodes. By mapping these hierarchical vectors, the study systematically evaluates high-risk intrusion pathways based on execution feasibility, technical prerequisites, and system exposure (Kasturi et al., 2024). Ultimately, this analysis serves to formulate an eight-point defense-in-depth mitigation framework that integrates cryptographic transport protocols, access control hardening, endpoint privilege restrictions, and administrative governance to secure academic laboratory environments.

The primary technical contribution of this research is the establishment of a structured threat-decomposition model tailored specifically to academic laboratory information systems, bridging the gap between theoretical vulnerability assessment and modern cybersecurity control standards. Furthermore, this study proceeds on the engineering hypothesis that transitioning from an implicitly trusted intranet model to a multi-layered security architecture will significantly reduce the operational feasibility of credential sniffing, automated brute-force attacks, and insider data manipulation when compared to unhardened legacy configurations.

RESEARCH METHOD

The primary engineering object investigated in this study is the Informatics Laboratory Information System (SILABTI), an intranet-based academic administration platform deployed across the faculty server infrastructures of Gunadarma University at Campus E (Kelapa Dua) and Campus J (Kalimalang). The unit of analysis encompasses the system's complete sociotechnical ecosystem, including its web application interface, local area network (LAN) topology, MySQL relational database architecture, and the administrative workflows executed by permanent laboratory assistants. The empirical scope specifically

targets the integrity of practicum grading data and graduation statuses, which represent high-value digital assets vulnerable to unauthorized modification.

This study adopts an analytical, systems-engineering research design based on hierarchical Attack Tree modeling to systematically map, evaluate, and mitigate security vulnerabilities within the SILABTI environment. Rather than evaluating software exploits or network configurations in isolation, the research design models the infrastructure from an adversarial perspective to capture multi-stage penetration pathways. Methodologically, the investigation is executed through a three-stage procedural workflow that integrates system boundary identification, iterative threat modeling, and defensive engineerin recommendations. The complete sequence of this analytical workflow is conceptualized in Figure 1.

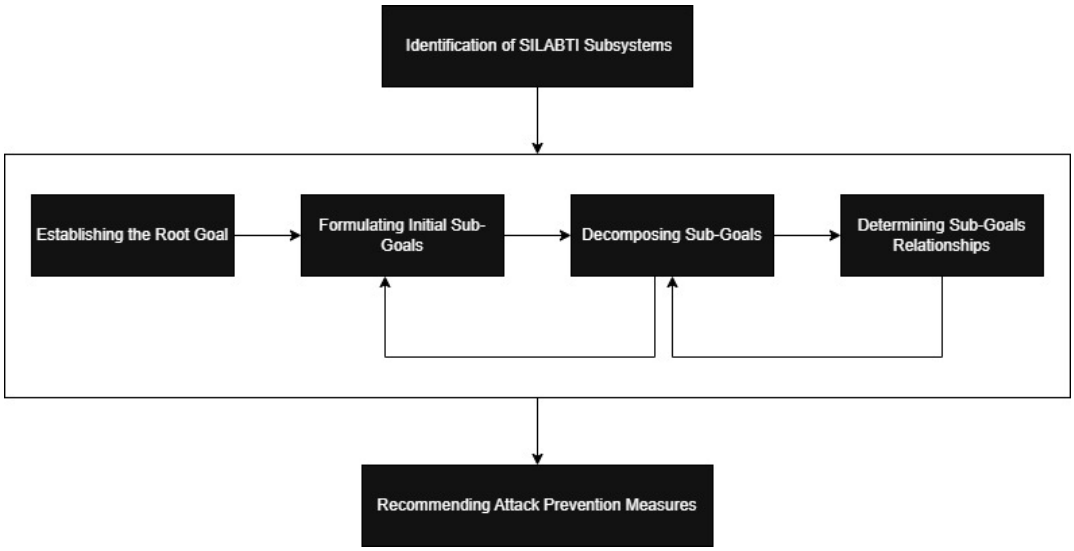

Figure 1. Procedural Workflow of the Hierarchical Attack Tree Security Analysis

As illustrated in Figure 1, the execution of the research design progresses through three consecutive operational stages:

1. Identification of SILABTI Subsystems: The preliminary stage delineates the physical and logical boundaries of the SILABTI architecture by auditing the underlying software environment; specifically its legacy PHP version 5.1.6 runtime and MySQL version 5.0.24 database engine; using network reconnaissance tools such as NMAP, while GraphViz was utilized to visualize and verify the hierarchical attack tree models
2. Iterative Attack Tree Modeling Process: The core analytical phase applies Bruce Schneier's Attack Tree methodology within a cyclical modeling loop designed to

uncover all operationally viable intrusion vectors. Because cyber penetration paths are non-linear, this stage operates through an iterative four-step sequence:

a. Establishing the Root Goal: The modeling hierarchy is anchored by defining the top-level node (Level 1 Root Goal), which represents the ultimate adversarial objective: modifying practicum grades or graduation statuses within the SILABTI database.
b. Formulating Initial Sub-Goals: The root goal is broken down into Level 2 intermediate objectives representing broad adversarial approaches, including credential harvesting, web application code exploitation, direct database breaches, and human-layer insider collusion.
c. Decomposing Sub-Goals: Each intermediate objective undergoes granular decomposition into Level 3 technical leaf nodes, which specify the exact tools, network conditions, or physical access required to execute the exploit. As indicated by the feedback loops in Figure 1, the discovery of complex leaf-level execution requirements continuously refines and expands the initial sub-goals to ensure total architectural coverage.
d. Determining Sub-Goal Relationships: Boolean logic gates are assigned to define the functional dependencies among nodes. OR gates represent alternative, independent attack pathways where achieving any single sub-goal satisfies the parent objective, whereas AND gates represent sequential prerequisite chains requiring the simultaneous fulfillment of all connected sub-tasks. This relational mapping is continuously re-evaluated against the decomposed sub-goals to eliminate acyclic errors and ensure logical consistency across the tree.

3. Recommending Attack Prevention Measures: In the final stage, the validated Attack Tree is evaluated as an engineering diagnostic framework to formulate defense-in-depth countermeasures. By assessing the technical skill prerequisites, execution simplicity, and system exploitability of each mapped path, critical vulnerabilities are ranked. Countermeasures are then engineered to systematically disrupt the Boolean chains, integrating cryptographic network controls, role-based access governance, hardware filtering, and human resource ethics to safeguard the system.

RESULT

Primary Attack Tree Architecture (Level 1 Decomposition)

Empirical network diagnostic evaluations and vulnerability assessments of the Informatics Laboratory Information System (SILABTI) server environment reveal that unauthorized practicum grade modification is achievable through four primary, independent intrusion vectors. As conceptualized in Figure 2, these four foundational vectors comprising credential acquisition, web application exploitation, direct database penetration, and human-layer insider collaboration are structured hierarchically under the Level 1 root objective using Boolean OR logic. This architectural configuration demonstrates that an adversary does not need to execute a complete compromise of the laboratory's physical hardware or software stack to alter academic records. Instead, successfully breaching any single Level 1 pathway provides sufficient privilege to manipulate practicum graduation statuses within the underlying MySQL 5.0.24 database. A structural evaluation of these primary vectors indicates that the threat landscape is evenly divided between technical software exploits and socio-technical or credential-harvesting vulnerabilities. Consequently, from an engineering security standpoint, patching web application source code is mathematically insufficient to guarantee database integrity if local network traffic sniffing or human-layer collusion pathways remain unmitigated.

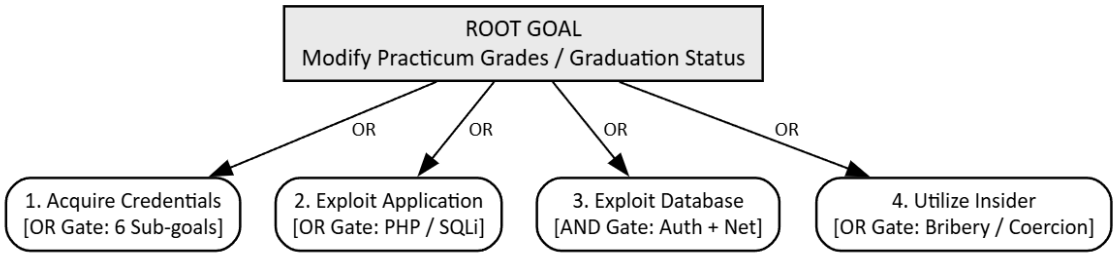

Figure 2. Level 1 Primary Attack Tree Architecture for SILABTI

Decomposed Intrusion Pathways (Level 2 and Level 3 Analysis)

Granular decomposition of the four primary Level 1 vectors into Level 2 intermediate sub-goals and Level 3 technical execution leaf nodes isolates the specific operational prerequisites required to complete an intrusion. Table 1 summarizes the hierarchical relationship between these decomposed attack paths, their governing Boolean logic gates, and the empirical technical mechanisms required for execution. The data summarized in Table 1 reveals a pronounced operational dependency on local intranet access across nearly

all technical intrusion pathways. Specifically, six of the eight technical leaf nodes mandate that an adversary first establish an active connection to the laboratory's wired Ethernet or wireless network infrastructure. Furthermore, while application and database exploitation paths require specialized proficiency in PHP 5.1.6 syntax and SQL injection parameters, credential acquisition pathways including Address Resolution Protocol (ARP) poisoning, packet sniffing, and workstation keystroke logging rely on readily accessible, open-source packet manipulation utilities. This disparity in technical execution barriers indicates that an attacker is significantly more likely to bypass the application layer entirely, opting instead to intercept unencrypted local network traffic or exploit shared staff workstations to capture valid authentication credentials.

Tabel 1. Hierarchical Matrix of SILABTI Attack Paths and Technical Execution Prerequisites

Root Vector (Level 1)	Intermediate Goal (Level 2)	Execution Leaf Nodes (Level 3)	Logic Gate	Required Technical Mechanism / Tool
1. Credential Acquisition	1.1 Network Sniffing	1.1.1 Network Access 1.1. 2 Packet Sniffer Software	AND	Local LAN/Wi-Fi connection; Wireshark / Tcpdump
	1.2 Written Records	1.2.1 Waste Disposal Search 1.2.2 Physical Archive Search	OR	Physical laboratory access; administrative oversight
	1.3 Brute Force Attack	1.3.1 Network Access 1.3.2 Password Cracker Tool	AND	Automated cracking script; Hydra / John the Ripper
	1.4 Assistant Coercion	1.4.1 Monetary Bribery 1.4.2 Intimidation / Threats	OR	Social engineering; financial inducement
	1.5 Phishing / Spoofing	1.5.1 Replicating Web UI 1.5.2 Network Access 1.5.3 ARP Poisoning Attack	AND	Spoofed PHP login page; Ettercap / Arpspoof
	1.6 Keylogger Deployment	1.6.1 Workstation Access 1.6.2 Keylogger Installation	AND	Physical/manual workstation access; silent monitoring tool
2. Application Exploitation	2.1 PHP Vulnerabilities	2.1.1 HP 5.x Legacy Flaw	AND	Exploit payloads targeting unpatched memory and parsing

		Exploit (5.1.6–5.2.8) 2.1.2 Web Traffic Monitoring 2.1.3 Network Access		vulnerabilities inherent to legacy PHP 5.x releases (including v5.1.6 and v5.2.8); active LAN connection
	2.2 SQL Injection (SQLi)	2.2.1 Network Access 2.2.2 Source Code Analysis 2.2.3 Database Version Data	AND	Modified SQL query parameters via HTTP POST/GET
3. Database Exploitation	3.1 Direct DB Authentication	3.1.1 Network Sniffing 3.1.2 Database Brute Force	OR	Packet capture of MySQL port 3306; direct SQL login test
	3.2 Network Authorization	Validated IP Address	N/A	Authorized MAC address / Local IP routing
4. Insider Exploitation	4.1 Willing Collaborator	Mutual Symbiosis Agreement	N/A	Interpersonal relationship with authorized assistant
	4.2 Paid Collaborator	Financial Transaction	N/A	Bribery of authorized laboratory personnel

Comparative Analysis of High-Risk Critical Attack Vectors

By evaluating the complete Attack Tree against the technical expertise required, execution complexity, and probability of system detection, four specific attack vectors emerge as critical, high-severity security threats. Table 2 provides a comparative evaluation of these four high-risk vectors comprising local network sniffing, workstation keystroke logging, automated brute-force authentication cracking, and human-layer insider bribery against key operational risk metrics. The comparative matrix in Table 2 demonstrates that the most critical vulnerabilities threatening SILABTI are characterized by low technical prerequisite barriers combined with an exceptionally low probability of detection. Local network sniffing operates as a completely passive interception mechanism; because SILABTI transmits HTTP session traffic across the intranet without Transport Layer Security (TLS/HTTPS) encryption, packet analyzers capture valid administrator credentials in plaintext without triggering system alarms or leaving network audit footprints. Similarly, workstation keystroke logging exploits unrestricted operating system privileges on shared staff computers, recording alphanumeric inputs via stealthy background processes that evade standard antivirus detection. Most critically, human-layer

insider bribery exhibits the highest risk severity because unauthorized grade alterations are executed using legitimately authorized assistant credentials. Consequently, such infractions leave no technical anomalies in standard application logs, proving that technical software defenses remain ineffective unless reinforced by immutable audit trail logging and ethical human resource governance.

Table 2. Comparative Vulnerability Assessment of High-Risk Critical Attack Vectors

Critical Attack Vector	Targeted System Layer	Required Technical Expertise	Execution Simplicity	Detection Probability	Risk Severity Level
Local Network Sniffing	Communication Protocol	Low to Moderate	High	Very Low (Passive)	Critical
Workstation Keylogging	Staff Operating System	Low	High	Very Low (Invisible)	Critical
Automated Brute Force	Application / DB Auth	Moderate	Moderate	Moderate to High	High
Insider Bribery / Coercion	Human / Administrative	None	Very High	Extremely Low	Critical
Critical Attack Vector	Targeted System Layer	Required Technical Expertise	Execution Simplicity	Detection Probability	Risk Severity Level

DISCUSSION

The empirical vulnerability assessment and hierarchical Attack Tree modeling of the SILABTI infrastructure establish that the security posture of an intranet academic information system is defined by its weakest socio-technical link. The mathematical decomposition of the root goal into 4 primary vectors, 12 intermediate sub-goals, and 19 execution leaf nodes demonstrates that unauthorized practicum grade modification can be achieved through multiple non-technical and low-complexity technical pathways. Specifically, local network sniffing, workstation keystroke logging, automated brute-force authentication cracking, and human-layer insider bribery emerge as the four most critical vectors. These empirical findings prove that protecting legacy web applications requires an integrated defense strategy that addresses network protocol vulnerabilities, operating system access rights, and human governance simultaneously.

The physical and logical susceptibility of SILABTI to these critical vectors is directly rooted in its legacy system architecture and local network deployment model. SILABTI was developed using PHP version 5.1.6 and is supported by MySQL version 5.0.24 software environments that lack the secure coding structures and memory protections enforced in modern web standards. When an application operates over unencrypted HTTP

protocol within a shared local intranet, every session token, username, and password transmitted between the permanent assistant's browser and the web server is broadcast across the network medium in plaintext. Consequently, an attacker who gains access to the LAN can deploy packet capture utilities (e.g., Wireshark or Tcpdump) or execute Address Resolution Protocol (ARP) poisoning to intercept these authentication payloads passively, requiring zero exploitation of the web application itself.

Furthermore, the vulnerability of staff workstations to keystroke logging stems from unrestricted operating system access privileges within the physical laboratory environment. Permanent assistants routinely utilize shared or semi-public workstations in staff rooms where practicum participants often request physical assistance or submit module assignments. Because these workstations are configured with default administrator-level privileges for everyday users, an attacker can execute unattended USB insertions or install stealthy software keyloggers in user space without triggering administrative elevation prompts. In the application layer, SQL injection (SQLi) vulnerabilities persist because legacy PHP 5.1.6 web applications frequently assemble database queries through direct string concatenation without mandatory prepared statements or parameterized variable binding. When input parameters are not sanitized, an attacker can manipulate HTTP request fields to inject arbitrary SQL logic directly into the MySQL 5.0.24 execution engine, altering grade tables without valid credentials. Finally, automated brute-force attacks remain viable because the application lacks algorithmic rate-limiting, CAPTCHA verification, or progressive account lockout policies, allowing password crackers to cycle through credential dictionaries unimpeded. Although SILABTI operates on PHP version 5.1.6, the application remains broadly susceptible to documented exploits affecting PHP 5.x variants (up through version 5.2.8), as these legacy releases share unpatched structural vulnerabilities in memory handling and input parsing

The findings of this study reinforce and extend seminal information systems security literature while providing valuable insights for contemporary digital trust and systems governance frameworks. Classic deterrence and security compliance studies argue that user awareness of countermeasures and formal security policies significantly reduces systems misuse. However, our empirical Attack Tree modeling demonstrates that policy awareness alone is insufficient when system architecture permits anonymous technical exploitation (e.g., sniffing) or when employees rationalize security policy violations through economic

inducement (insider bribery). This aligns with behavioral security models indicating that compliance depends heavily on organizational culture and perceived detection risk**.**

To systematically eliminate the four critical attack vectors and harden SILABTI's infrastructure against multi-layered intrusions, an eight-point defense-in-depth engineering mitigation framework must be implemented:

1. Mandatory Cryptographic Encryption (TLS/HTTPS): The laboratory server must enforce Transport Layer Security (TLS) across all HTTP sessions. Deploying end-to-end HTTPS encryption ensures that all authentication credentials, session cookies, and grading data transmitted between browsers and the SILABTI server are mathematically encrypted. This immediately neutralizes local LAN sniffing and packet capture attacks.
2. Immutable Audit Trail Logging: An automated, tamper-resistant audit logging engine must be integrated into the database and application layer. The logging system must capture timestamps, IP addresses, authenticated account IDs, and the exact before-and-after values of any modified grade records. This provides forensic transparency to detect anomalous modifications and deters insider abuse.
3. Role-Based Access Control (RBAC): SILABTI's authorization architecture must be re-engineered around strict RBAC principles. Permanent assistants should only receive access permissions for the specific practicum modules, class sections, and grading tables assigned to their teaching schedule, preventing horizontal privilege escalation across unrelated academic courses.
4. Hardware Address Authentication (MAC Filtering): The laboratory's managed switches and wireless routers must activate Media Access Control (MAC) address filtering. Only preregistered physical MAC addresses belonging to authorized staff workstations should be granted network routing to the SILABTI server subnet, blocking alien devices from connecting to the intranet.
5. Strong Password Complexity and Periodic Administrator Credential Rotation: The application must enforce stringent credential policies requiring a minimum password length that combines uppercase letters, lowercase letters, numeric digits, and cryptographic symbols to mitigate automated brute-force attacks. Furthermore, a mandatory 90-day password rotation policy must be strictly enforced with highest priority assigned to Server Administrator accounts. Because server administrators possess root-level database access capable of overriding application logs,

scheduling frequent and mandatory credential rotations for administrative profiles is critical to minimizing the operational window of vulnerability if a superuser token is compromised.

6. Workstation Privilege Restriction (Non-Administrator OS Accounts): All permanent assistants must be assigned standard, non-administrator user accounts on their physical laboratory workstations. Revoking local administrator rights prevents assistants or unauthorized visitors from installing unauthorized software keyloggers, packet sniffers, or unauthorized hardware devices.
7. Phishing and Network Spoofing Countermeasures: To protect against ARP poisoning and spoofed login applications, the network infrastructure must implement Dynamic ARP Inspection (DAI) and DHCP Snooping on local network switches. Staff must also be trained to verify SSL certificate authenticities before entering credentials.
8. Human Resource Ethical Governance and Digital Trust Building: Technical controls must be reinforced by institutional ethics programs. Permanent assistants must undergo formal academic integrity training that instills professional ethics and explicitly warns against the legal and moral consequences of bribery. Establishing a strong culture of digital trust ensures that laboratory personnel defend system integrity rather than acting as vulnerability vectors.

CONCLUSION

This study has successfully conducted a rigorous vulnerability assessment and hierarchical threat modeling of the Informatics Laboratory Information System (SILABTI) using Bruce Schneier's Attack Tree methodology. By systematically decomposing the root objective of unauthorized practicum grade modification into four primary Level 1 vectors, twelve Level 2 sub-goals, and nineteen Level 3 execution leaf nodes, the empirical analysis demonstrated that legacy academic information systems are vulnerable across both software and human layers. Crucially, the comparative risk assessment identified four critical, high-risk attack vectors, workstation keystroke logging, automated brute-force attacks, and human-layer insider bribery. These findings confirm that securing academic

database integrity requires an integrated security strategy rather than isolated software patching.

The scientific and technical contribution of this research lies in the formulation of an integrated, defense-in-depth risk governance model that bridges technical cybersecurity controls with organizational human resource governance. By synthesizing cryptographic transport encryption (TLS/HTTPS), hardware MAC address filtering, immutable audit trail logging, strict Role-Based Access Control (RBAC), OS workstation privilege demotion, and ethical integrity training into a unified mitigation architecture, this study extends classical information systems security literature. The proposed engineering framework provides a reference model for university IT departments and systems administrators seeking to harden legacy web-based intranets against multi-stage, socio-technical cyber intrusions.

Despite its analytical depth, this study exhibits methodological limitations that warrant consideration and addressment in future research. First, the vulnerability assessment was conducted within a static local intranet environment utilizing specific legacy configurations (PHP 5.1.6 and MySQL 5.0.24); consequently, the empirical attack tree matrices may require structural adaptation when applied to modern, cloud-hosted, or microservices-based academic platforms. Second, while the study models adversarial pathways and formulates mitigations conceptually, quantitative penetration testing metrics (such as Mean Time to Compromise or statistical attack frequency distributions) were not formally measured. Future research should empirically validate the proposed eight-point mitigation framework by deploying quantitative vulnerability scanning, red-team penetration testing, and structural equation modeling to measure the impact of ethical governance and security controls on user digital trust and institutional resilience.

REFERENCES

Al-Shareeda, M. A., Manickam, S., & Sari, S. A. (2022). A Survey of SQL Injection Attacks, Their Methods, and Prevention Techniques. 2022 International Conference on Data Science and Intelligent Computing (ICDSIC),

Kasturi, S., Li, X., Li, P., & Pickard, J. (2024). A proposed approach to integrate application security vulnerability data with incidence response systems. *American Journal of Networks and Communications*, *13*(1), 19-29. https://doi.org/10.11648/j.ajnc.20241301.12

Konarski, P., & Ptak, M. (2020). Method for attack tree data transformation and import into IT risk analysis expert systems. *Applied Sciences*, *10*(23), 8423. https://doi.org/10.3390/app10238423

Li, M. (2023). An Effective Vulnerability Detection Framework for Web Applications Using Multi-Modal Data Representation. *Computers & Security*, *132*, 103362. https://doi.org/10.1016/j.cose.2023.103362

National Institute of, S., & Technology. (2021). *Developing Cyber-Resilient Systems: A Systems Security Engineering Approach* (NIST Special Publication 800-160 Volume 2 Revision 1, Issue. https://doi.org/10.6028/NIST.SP.800-160v2r1

National Institute of, S., & Technology. (2022). *Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities* (NIST Special Publication 800-218, Issue. https://doi.org/10.6028/NIST.SP.800-218

Oliveira, A. V. d. S. (2022). Web Application Security Testing and Assessment: A Systematic Literature Review. *Journal of Systems and Software*, *186*, 111195. https://doi.org/10.1016/j.jss.2021.111195

Piètre-Cambacédès, L., & Bouillon, C. (2022). Boolean Logic Driven Attack Trees (BDAT): A Tree-Based Formalism for Cyber-Security Analysis. *Computers & Security*, *114*, 102580. https://doi.org/10.1016/j.cose.2021.102580

Tang, Y. (2023). A Comprehensive Survey on Web Application Vulnerabilities Analysis and Detection. *Computers & Security*, *129*, 103204. https://doi.org/10.1016/j.cose.2023.103204