Modeling and Simulating Cyber Attacks Using Attack Trees and Security Testing Tools: A Case Study of an ICT Department

D Jayus Nor Salim
Department of Information Technology, Universitas Tidar, Magelang, Indonesia
Email: djayus.nur@untidar.ac.id

The increasing reliance of higher education institutions on information systems has significantly expanded the cyber attack surface, making academic environments attractive targets for cyber threats. This study proposes an attack tree–based approach to model and simulate potential cyber attacks against an academic information system managed by an ICT department. The research employs a controlled case study design, combining technical attack simulation and analytical modeling to identify realistic attack paths and prioritize mitigation strategies. Cyber attack simulations were conducted in a staging environment using Nmap for network reconnaissance, OWASP ZAP for dynamic web application security testing, and SQLMap for controlled verification of potential SQL injection vulnerabilities. The results of these simulations were systematically mapped into an attack tree model, representing hierarchical attack paths from initial reconnaissance to exploitation and potential impact. Each node in the attack tree was evaluated based on likelihood and impact to support risk prioritization. The findings indicate that the most critical attack paths are associated with web application vulnerabilities and weaknesses in authentication mechanisms, which may lead to unauthorized access and data exposure if left unmitigated. The attack tree model effectively integrates technical evidence from multiple tools into a structured analytical framework, enabling clearer visualization of attack feasibility and mitigation priorities. This study demonstrates that attack tree–based modeling can serve as a practical and systematic approach to strengthening cybersecurity posture in academic ICT departments..
Keywords: Cyber attack simulation, Attack tree, Information security, Academic information systems, ICT department.

Corresponding Author:
D Jayus Nor Salim
Universitas Tidar
Jl. Kapten Suparman No.39, Potrobangsan, Kota Magelang, Jawa Tengah 56116
djayus.nur@untidar.ac.id

1. Introduction

The rapid digital transformation in higher education has significantly increased the dependency of institutions on integrated information systems to manage academic, administrative, and research-related processes. Academic information systems store and process sensitive data, including student identities, academic records, financial information, and institutional documents. As a result, these systems have become attractive targets for cyber attackers seeking unauthorized access, data exfiltration, or service disruption. The growing complexity of cyber threats, combined with heterogeneous IT infrastructures in academic environments, demands systematic and proactive security evaluation approaches[1]. Cyber attacks against educational institutions have increased in frequency and sophistication in recent years. Common attack vectors include web application vulnerabilities, misconfigured services, weak authentication mechanisms, and database exploitation. Despite the adoption of baseline security controls such as HTTPS, firewalls, and content delivery networks (CDNs), many institutions continue to face risks due to insufficient vulnerability assessment, lack of structured attack modeling, and limited integration between technical findings and strategic risk analysis. In many cases, security testing is performed in an ad hoc or reactive manner rather than as part of a structured analytical framework[2].

Attack modeling techniques provide a systematic way to understand how an adversary may exploit vulnerabilities to achieve specific objectives. Among these techniques, the attack tree approach offers a hierarchical and logical representation of potential attack paths, linking technical weaknesses to broader security impacts. Attack trees allow security analysts to decompose a high-level attacker goal into sub-goals and actionable techniques, connected through logical operators (AND/OR). This structure supports both qualitative and quantitative risk evaluation by associating likelihood and impact values with individual nodes[3]. Although attack trees have been widely discussed in cybersecurity research, their practical integration with real-world security testing tools in academic environments remains relatively limited. Many studies focus either on theoretical modeling without empirical validation or on technical penetration testing without structured risk modeling. There is therefore a need for research that bridges these two perspectives by combining controlled cyber attack simulation with systematic attack tree modeling to produce actionable and prioritized mitigation strategies[4].

This study addresses that gap by proposing an attack tree–based cyber attack simulation framework applied to an academic information system managed by an ICT department. The research integrates network reconnaissance using Nmap, dynamic web application testing using OWASP ZAP, and controlled SQL injection verification using SQLMap within a staging environment. The technical findings are subsequently mapped into an attack tree model to identify realistic attack paths, evaluate their likelihood and impact, and determine mitigation priorities[5]. The main contributions of this study are threefold. First, it demonstrates a structured method for integrating multi-tool security testing results into an attack tree framework. Second, it provides empirical evidence from a controlled academic ICT environment to validate the feasibility of modeled attack paths. Third, it offers a practical risk-based prioritization mechanism that supports decision-making for improving cybersecurity posture in academic ICT departments[6].

2. Related Works

Cybersecurity assessment in academic environments has been widely studied, particularly in relation to vulnerability analysis, penetration testing, and risk management frameworks. Previous studies have emphasized the importance of systematic security evaluation in higher education institutions due to their open network architecture and diverse user populations. Web application vulnerabilities, authentication weaknesses, and database misconfigurations have been consistently identified as dominant attack vectors in the education sector[7]. Attack tree modeling, originally introduced as a structured threat modeling technique, has been applied in various domains such as critical infrastructure, cloud computing, and IoT systems. The approach enables hierarchical decomposition of attacker goals into sub-goals and actionable techniques, connected through logical operators (AND/OR). Several studies have extended attack trees with probabilistic evaluation, cost modeling, and risk scoring mechanisms to support decision-making[8].

On the other hand, technical security testing tools such as Nmap, OWASP ZAP, and SQLMap are commonly used in penetration testing and vulnerability assessments. Nmap supports network reconnaissance and service enumeration, OWASP ZAP facilitates dynamic web application security testing, and SQLMap automates detection and verification of SQL injection vulnerabilities. However, many prior works treat these tools as standalone technical instruments without integrating their findings into a structured analytical attack model[9]. A research gap therefore exists in bridging empirical vulnerability testing with formal attack modeling within academic ICT environments. Few studies demonstrate how multi-tool security testing outputs can be systematically mapped into an attack tree structure to enable risk-based prioritization of mitigation strategies. This study contributes by integrating controlled attack simulation with structured attack tree modeling in an academic ICT department context.

3. Method

Methodology

The methodology integrates technical security testing procedures with structured attack modeling. The overall process is illustrated through five main steps on figure 1:

Figure 1 structured attack modeling

a. Environment Preparation
   A staging environment was prepared to replicate the academic information system, including web application modules, authentication mechanisms, database services, and supporting infrastructure components. System configurations and testing scope were documented prior to experimentation.
b. Network Reconnaissance and Service Enumeration (Nmap)
   Nmap was used to identify active hosts, open ports, service versions, and exposed technologies. The reconnaissance phase aimed to determine the initial attack surface and detect potential entry points at the network layer. Output results were sanitized for publication and stored securely for internal validation[10].
c. Web Application Vulnerability Assessment (OWASP ZAP)
   OWASP ZAP was employed to perform dynamic application security testing (DAST). The process included:
   1. Spidering and endpoint discovery
   2. Passive vulnerability scanning
   3. Active scanning of selected parameters

   Detected issues were categorized based on severity levels (High, Medium, Low) and grouped into categories such as Injection, Authentication Weakness, Access Control, and Information Disclosure[11].
d. Controlled SQL Injection Verification (SQLMap)
   SQLMap was used to verify suspected SQL injection vulnerabilities identified during web application testing. Testing was conducted using limited risk configurations to avoid data extraction or service disruption. The objective was to confirm exploit feasibility rather than perform full database compromise[12].
e. Attack Tree Construction and Risk Evaluation
   Validated findings from all tools were mapped into an attack tree structure. The root node represents the attacker's primary objective: unauthorized access to the academic information system. Intermediate nodes represent attack paths such as web exploitation, authentication abuse, and database exploitation. Leaf nodes represent potential impacts, including unauthorized data access and privilege escalation[13].
   Each node was evaluated using a simplified risk scoring approach based on:
   1. Likelihood (derived from tool confirmation and exposure level)
   2. Impact (based on potential data sensitivity and system criticality)

3. Risk priority was calculated qualitatively using a probability–impact (P × I) perspective to determine mitigation urgency.

All simulations were conducted under formal authorization within a controlled staging environment. Sensitive technical outputs such as IP addresses, raw logs, payloads, and command results were anonymized and securely stored. Only summarized and sanitized findings are presented in the published results to protect institutional confidentiality.

Research Design

This study employs a case study–based research design combined with a mixed-methods approach. The qualitative component focuses on contextual understanding of security practices, system architecture, and administrative controls within the ICT department. The quantitative component derives from empirical technical findings generated through controlled cyber attack simulations[14]. The research was conducted in a controlled staging environment configured to replicate the production academic information system. The use of a staging environment ensures that security testing does not disrupt operational services while maintaining realistic system configurations. All testing activities were performed with formal authorization from the responsible ICT authority. The research design consists of three sequential phases:

1. Technical Simulation Phase: Conducting structured reconnaissance, vulnerability identification, and controlled verification using selected security testing tools[15].
2. Attack Tree Modeling Phase: Translating validated technical findings into hierarchical attack tree nodes representing potential attack paths[16].[17]
3. Risk Evaluation and Prioritization Phase: Assessing likelihood and impact for each attack path to determine mitigation priorities[18].

This sequential design ensures traceability between empirical findings and analytical modeling outcomes.

4. Results and Discussion

Results

Network Reconnaissance and Attack Surface Identification

The initial reconnaissance phase using Nmap revealed that the web portal of the academic information system was accessible via HTTP and HTTPS services. The system was positioned behind a content delivery network (CDN) and reverse proxy layer, which masked direct backend exposure. While this configuration provides an additional defensive layer, application-level fingerprints were still observable. The scan identified web technologies indicating a PHP-based environment and content management components consistent with widely deployed web frameworks. Several administrative and system-related endpoints were detected, including protected directories and REST-style interfaces. Although certain directories returned restricted access responses (e.g., HTTP 401), their presence contributes to the overall attack surface. No publicly accessible SSH service was confirmed during external scanning, suggesting firewall filtering or access restriction mechanisms were in place. From a network-layer perspective, the exposure level was moderate; however, application-layer components presented a larger potential attack surface.

Web Application Vulnerability Findings

Dynamic application security testing using OWASP ZAP identified multiple vulnerability categories across tested endpoints. A total of nine potential security issues were documented across five tested targets within the staging environment, categorized as follows: 2 High severity findings, 5 Medium severity findings, 2 Low severity findings.

The primary vulnerability categories included: Injection-related weaknesses, Authentication mechanism weaknesses, Access control misconfigurations, Minor information disclosure issues. A total of nine potential security findings were identified across the tested targets. The distribution of vulnerabilities based on severity level is presented in Table 1.

Table 1. Vulnerability Severity Distribution

Severity Level	Number of Findings	Percentage (%)
High	2	22.20%
Medium	5	55.60%
Low	2	22.20%
Total	9	100%

As shown in Table 1, medium-severity vulnerabilities represent the majority (55.6%) of the findings. However, the presence of high-severity vulnerabilities (22.2%) indicates the existence of potentially critical attack paths that require immediate mitigation. The distribution of vulnerability severity levels is further illustrated in Figure 2.

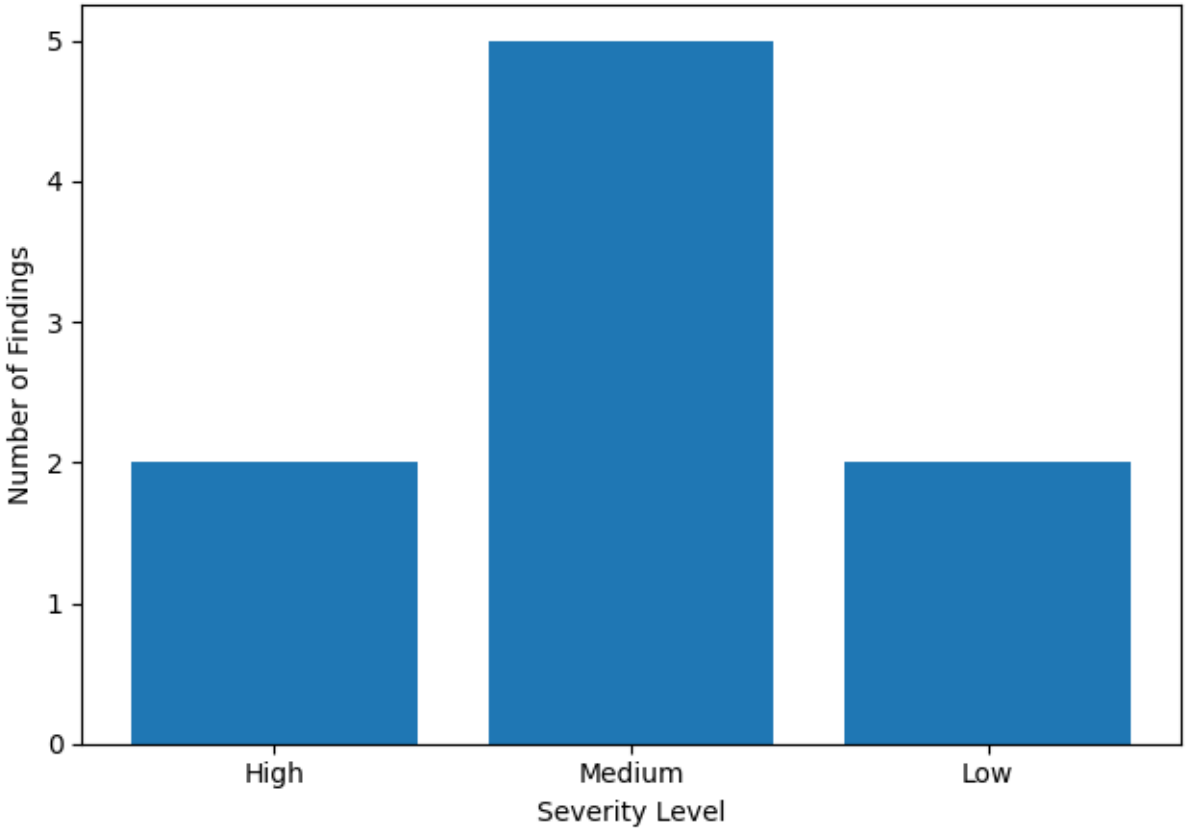

Figure 2. Vulnerability Severity Distribution

As illustrated in Figure 2, medium-severity vulnerabilities dominate the findings, while high-severity issues, although fewer in number, represent critical risk exposure due to their potential impact. High-severity findings were associated with authentication weaknesses and potential injection vectors that could enable unauthorized access if successfully exploited. Medium-severity findings included insufficient input validation and indirect object reference patterns. Low-severity issues were primarily related to information disclosure through headers and non-critical system responses.

SQL Injection Verification

SQLMap was used to verify suspected SQL injection vulnerabilities detected during dynamic scanning. Controlled testing confirmed the feasibility of SQL injection behavior in two endpoints within the staging environment. One endpoint demonstrated response variations consistent with boolean-based injection patterns, while another indicated error-based injection characteristics.

Testing was conducted without performing data extraction or database dumping operations. The objective was limited to verifying exploit feasibility under controlled conditions. The confirmation of injection vectors significantly increased the likelihood score of related attack tree nodes. The identified vulnerabilities were further categorized based on their technical characteristics. The distribution of findings by vulnerability category is summarized in Table 2.

Table 2. Vulnerability Categories Identified

Vulnerability Category	Number of Findings	Dominant Severity
Injection	2	High–Medium
Authentication Weakness	2	High
Access Control Issues	3	Medium
Information Disclosure	2	Low–Medium
Total	9	—

Attack Tree Modeling Outcome

The hierarchical structure of the modeled attack paths is illustrated in Figure 3.

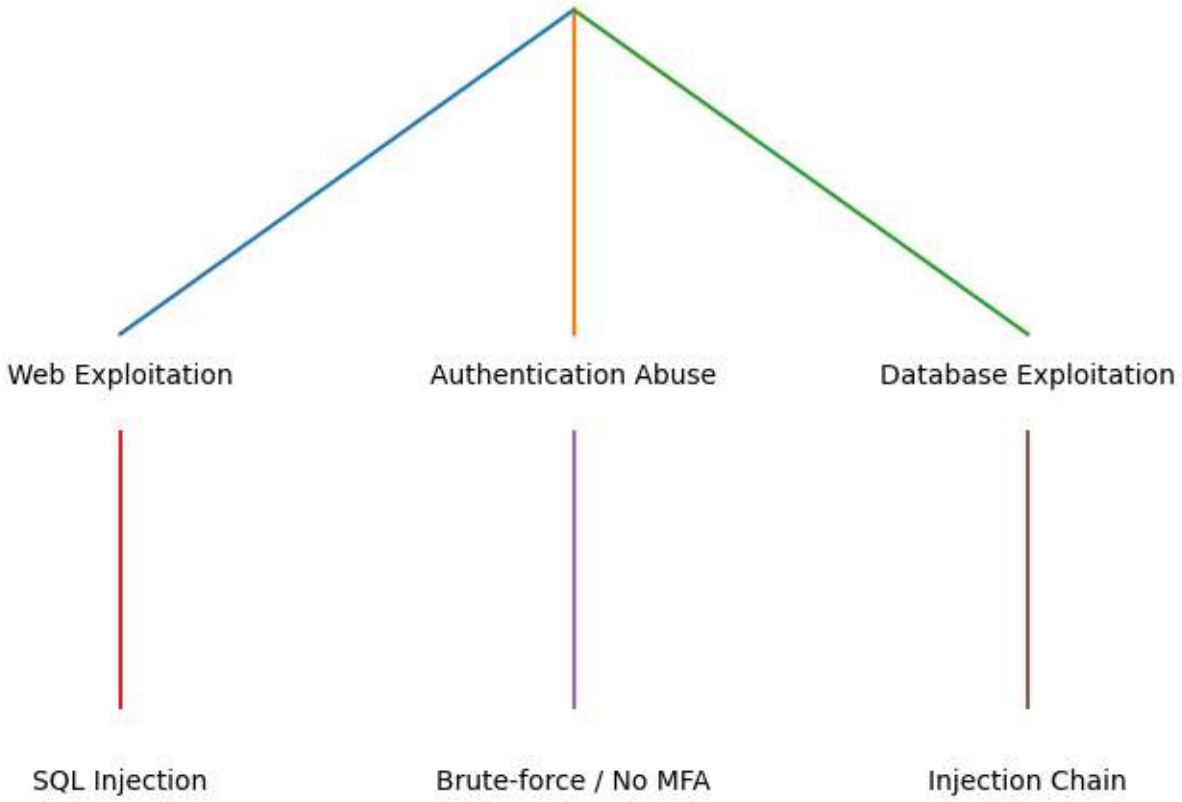

Figure 3. Attack Tree Modeling Outcome

As illustrated in Figure 3, the attack tree decomposes the primary attacker objective into three major attack paths. The model demonstrates how application-level vulnerabilities and authentication weaknesses can independently or collectively lead to unauthorized system access. This hierarchical representation clarifies

attack feasibility and supports structured risk prioritization. Based on validated findings, an attack tree model was constructed with the root objective defined as Three primary attack paths were identified:

1. Exploitation of web application vulnerabilities
2. Abuse of authentication mechanisms
3. Database exploitation via injection techniques

Sub-nodes included reconnaissance activities, injection exploitation, brute-force attempts, and privilege escalation scenarios. Leaf nodes represented potential impacts such as unauthorized data access, administrative account takeover, and academic record exposure.

Based on the validated findings, three primary attack paths were identified and evaluated using a qualitative probability–impact perspective. The prioritization results are summarized in Table 3.

Table 3. Primary Attack Paths and Risk Prioritization

Attack Path ID	Attack Description	Likelihood	Impact	Risk Level
AP-1	Web exploitation via SQL injection	High	High	Critical
AP-2	Authentication abuse (brute-force / no MFA)	High	High	Critical
AP-3	Database exposure through injection chain	Medium	High	High

As shown in Table 3, attack paths related to SQL injection and authentication abuse present the highest risk level due to their combination of high likelihood and high impact. These paths represent critical security priorities that require immediate mitigation. In contrast, database exposure through chained exploitation remains high risk but demonstrates slightly lower likelihood due to required preconditions. The qualitative risk positioning of the identified attack paths is illustrated in Figure 4.

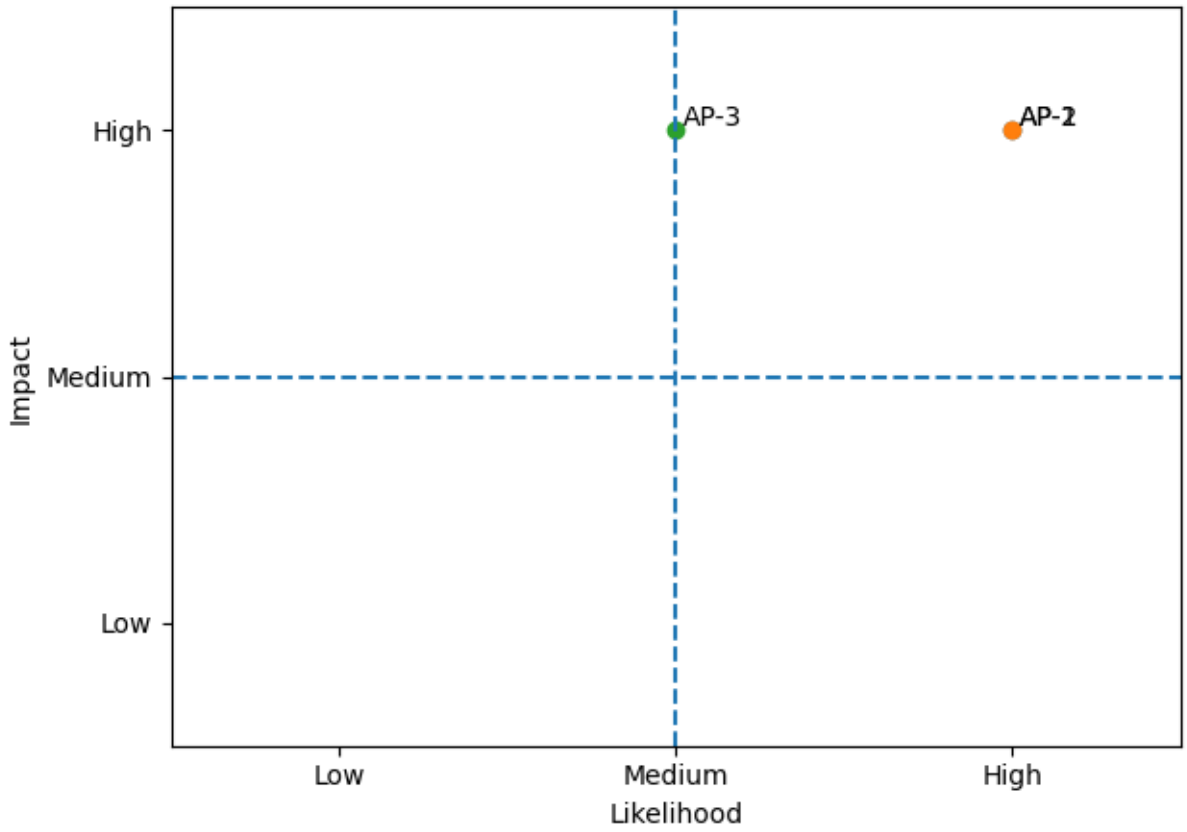

Figure 4 Probability–Impact Risk Matrix

As illustrated in Figure 4, AP-1 and AP-2 are positioned in the high-likelihood and high-impact quadrant, categorizing them as critical risk priorities. AP-3, while associated with high impact, demonstrates slightly lower likelihood due to required preconditions. This visualization supports mitigation prioritization based on

combined probability and impact considerations. Qualitative risk evaluation using a probability–impact perspective indicated that the highest-risk attack path involved a combination of authentication weakness and injection exploitation. This path presented both high likelihood (confirmed vulnerability) and high impact (sensitive data exposure).

Discusion

The findings of this study are consistent with prior research indicating that web application vulnerabilities and authentication weaknesses remain dominant attack vectors in higher education environments. Previous studies on academic cybersecurity assessments have similarly identified injection flaws and broken authentication as high-risk categories due to their direct impact on confidentiality and system integrity. In alignment with established vulnerability reports in educational institutions, this study confirms that application-layer weaknesses pose a greater practical threat than network-layer exposure when perimeter protections such as CDNs and HTTPS are implemented.

Compared to studies that rely solely on penetration testing reports, this research extends existing work by integrating empirical vulnerability findings into a structured attack tree model. While earlier research often presents vulnerabilities as isolated technical issues, the attack tree approach used in this study provides a hierarchical representation of how multiple weaknesses can be logically chained to achieve high-impact objectives. This structured modeling enhances risk visibility and supports mitigation prioritization based on combined likelihood and impact. From a technical perspective, the confirmed injection vectors highlight the continued necessity of secure coding practices, including strict input validation, parameterized queries, error-handling control, and consistent patch management. The identification of authentication-related weaknesses further emphasizes the importance of implementing rate-limiting mechanisms, multi-factor authentication (MFA), and session hardening controls. These measures reduce the feasibility of privilege escalation and account takeover scenarios identified in the modeled attack paths. From a managerial and governance perspective, the results suggest that cybersecurity practices in academic ICT departments should transition from reactive vulnerability handling to structured, model-driven security assessment. The integration of attack tree modeling into periodic security audits can improve risk communication between technical teams and decision-makers. Rather than reporting isolated technical findings, the hierarchical model provides management with clearer visibility into how vulnerabilities translate into strategic institutional risk. Furthermore, the study demonstrates the value of combining multiple security testing tools within a controlled workflow. By linking reconnaissance (Nmap), dynamic application testing (OWASP ZAP), and controlled exploitation verification (SQLMap), the research ensures traceability between detection and risk modeling. This integrated methodology strengthens analytical rigor compared to single-tool assessments.

Despite these contributions, several limitations must be acknowledged. First, the study was conducted within a controlled staging environment rather than a live production system. Although the staging configuration was designed to replicate operational conditions, certain real-world exposure variables may not be fully represented. Second, the risk evaluation employed a qualitative probability–impact approach rather than a fully quantitative probabilistic model. While sufficient for prioritization purposes, future research could incorporate Bayesian modeling or probabilistic attack graphs for more granular risk computation. Third, the study focused primarily on technical vulnerabilities and did not deeply analyze human-factor risks such as password reuse behavior or social engineering exposure, which are also significant contributors to cybersecurity incidents in academic environments. Overall, the discussion confirms that the integration of empirical security testing with structured attack tree modeling provides both technical validation and strategic risk insight. The approach not only identifies vulnerabilities but also

clarifies their systemic implications, thereby supporting more informed cybersecurity governance within academic ICT departments.

5. Conclusion

This study proposed and implemented an attack tree–based cyber attack simulation framework for evaluating the security posture of an academic information system managed by an ICT department. By integrating network reconnaissance (Nmap), dynamic web application testing (OWASP ZAP), and controlled SQL injection verification (SQLMap), the research demonstrated how empirical security testing results can be systematically transformed into a structured attack tree model for risk prioritization. The findings confirm that application-layer vulnerabilities and authentication weaknesses represent the most critical attack paths, even when baseline infrastructure protections such as HTTPS and CDN mechanisms are implemented. The constructed attack tree successfully illustrated how individual vulnerabilities may be logically chained to achieve high-impact outcomes, such as unauthorized access and potential data exposure. This structured modeling approach enhances the clarity of cybersecurity risk evaluation and supports more strategic mitigation planning within academic ICT environments.

Nevertheless, this study has several limitations. The simulations were conducted in a controlled staging environment rather than a fully operational production system, which may limit the representation of real-world exposure conditions. In addition, risk prioritization was based on a qualitative probability–impact assessment rather than a fully quantitative probabilistic model. The research also primarily focused on technical vulnerabilities, without extensive evaluation of human-factor risks such as user behavior, social engineering susceptibility, or organizational security culture. Future research may extend this work in several directions. First, incorporating quantitative probabilistic modeling techniques, such as Bayesian networks or probabilistic attack graphs, could provide more granular risk estimation. Second, integrating automated security testing tools into continuous integration/continuous deployment (CI/CD) pipelines may enable real-time vulnerability monitoring. Third, expanding the framework to include socio-technical factors and human-centered risk analysis would provide a more comprehensive cybersecurity assessment model for academic institutions. In conclusion, the integration of empirical cyber attack simulation with structured attack tree modeling offers a practical and replicable approach for improving cybersecurity posture in academic ICT departments. By acknowledging current limitations and outlining future research opportunities, this study contributes to the development of more systematic and evidence-based cybersecurity governance in higher education environments.

6. References

[1] A. Limanovskaja and V. Davidavičienė, “Digital Transformation in Higher Education: Challenges and Transformation Directions,” *Economics and Culture*, vol. 22, no. 2, pp. 83–92, Dec. 2025, doi: 10.2478/jec-2025-0016.

[2] I. Simplice, O. Fidel, C. G. Kennedy, K. Okokpujie, and S. Gabriel, “Enhancing Information System Security: A Vulnerability Assessment of a Web Application Using OWASP Top 10 List,” *Lecture Notes in Networks and Systems*, vol. 914 LNNS, pp. 385–397, 2024, doi: 10.1007/978-981-97-0573-3_31.

[3] A. Roy, D. S. Kim, and K. S. Trivedi, “Attack countermeasure trees (ACT): Towards unifying the constructs of attack and defense trees,” *Security and Communication Networks*, vol. 5, no. 8, pp. 929–943, 2012, doi: 10.1002/sec.299.

[4] R. Vigo, F. Nielson, and H. R. Nielson, "Automated generation of attack trees," *Proceedings of the Computer Security Foundations Workshop*, vol. 2014-January, pp. 337–350, Nov. 2014, doi: 10.1109/CSF.2014.31.

[5] A. Bhardwaj, V. Sapra, and L. Sapra, "Evading Firewalls & Enumerate SNMP Using Advanced NMAP Techniques," *2023 3rd Asian Conference on Innovation in Technology, ASIANCON 2023*, 2023, doi: 10.1109/ASIANCON58793.2023.10270155.

[6] J. Bryans, L. S. Liew, H. N. Nguyen, G. Sabaliauskaite, S. Shaikh, and F. Zhou, "A Template-Based Method for the Generation of Attack Trees," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12024 LNCS, pp. 155–165, 2020, doi: 10.1007/978-3-030-41702-4_10.

[7] Z. C. S. S. Hlaing and M. Khaing, "A Detection and Prevention Technique on SQL Injection Attacks," *2020 IEEE Conference on Computer Applications, ICCA 2020*, Feb. 2020, doi: 10.1109/ICCA49400.2020.9022833.

[8] F. Dorfhuber, J. Eisentraut, and J. Křetínský, "Learning Attack Trees by Genetic Algorithms," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 14446 LNCS, pp. 55–73, 2023, doi: 10.1007/978-3-031-47963-2_5.

[9] A. Chaturvedi, B. Lakhani, T. Agarwal, Mohana, M. Moharir, and A. R. Ashok Kumar, "A Comprehensive Vulnerability Tools Analysis for Security and Control in IT Environment and Organizations," *5th International Conference on Electronics and Sustainable Communication Systems, ICESC 2024 - Proceedings*, pp. 612–618, 2024, doi: 10.1109/ICESC60852.2024.10689860.

[10] F. Barman, N. Alkaabi, H. Almenhali, M. Alshedi, and R. Ikuesan, "A Methodical Framework for Conducting Reconnaissance and Enumeration in the Ethical Hacking Lifecycle," *European Conference on Information Warfare and Security, ECCWS*, 2023, Accessed: Feb. 21, 2026. [Online]. Available: https://www.scopus.com/pages/publications/85167621456?origin=scopusAI

[11] J. Koman and M. Janiszewski, "SCAnME - scanner comparative analysis and metrics for evaluation," *International Journal of Information Security 2025 24:3*, vol. 24, no. 3, pp. 147-, May 2025, doi: 10.1007/s10207-025-01054-8.

[12] M. Baklizi, M. Alkhazaleh, M. B. Y. Alzghoul, A. Maaita, J. Zraqou, and M. AlShaikh-Hasan, "Evaluating the effectiveness of Havij for structured query language injection exploitation in web applications," *Bulletin of Electrical Engineering and Informatics*, vol. 14, no. 6, pp. 4823–4833, Dec. 2025, doi: 10.11591/eei.v14i6.10751.

[13] A. Sethapanee, T. Nimitrchai, and S. Fugkeaw, "AutoRat: Automated Risk Assessment Tool for Network Mapper Scanning," *Lecture Notes in Networks and Systems*, vol. 453 LNNS, pp. 99–110, 2022, doi: 10.1007/978-3-030-99948-3_10.

[14] A. Zanke, T. Weber, P. Dornheim, and M. Engel, "Assessing information security culture: A mixed-methods approach to navigating challenges in international corporate IT departments," *Comput. Secur.*, vol. 144, p. 103938, Sep. 2024, doi: 10.1016/j.cose.2024.103938.

[15] R. Jhawar, B. Kordy, S. Mauw, S. Radomirović, and R. Trujillo-Rasua, "Attack trees with sequential conjunction," *IFIP Adv. Inf. Commun. Technol.*, vol. 455, pp. 339–353, 2015, doi: 10.1007/978-3-319-18467-8_23.

[16] B. Kordy, L. Piètre-Cambacédès, and P. Schweitzer, "DAG-based attack and defense modeling: Don't miss the forest for the attack trees," *Comput. Sci. Rev.*, vol. 13–14, no. C, pp. 1–38, 2014, doi: 10.1016/j.cosrev.2014.07.001.

[17] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing, "Automated generation and analysis of attack graphs," *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy*, vol. 2002-January, pp. 273–284, 2002, doi: 10.1109/SECPRI.2002.1004377.

[18] Y. Cherdantseva *et al.*, "A review of cyber security risk assessment methods for SCADA systems," *Comput. Secur.*, vol. 56, pp. 1–27, Feb. 2016, doi: 10.1016/j.cose.2015.09.009.