Electronic Integrated Computer Algorithm Journal
Vol. 4, No. 1, October 2026, pp 46~54

ISSN: 3031-0350, DOI: 10.62123/enigma.v4i1.235

A STRIDE-Based Threat Taxonomy for Digital Showcase Platforms in Vocational Higher Education

Donny Sanjaya[1*], Indri Sulistianingsih[1], Rizka Salsabila Riadi[2], Jen-Peng Huang[3], Chairul Rizal[4]

[1] *Department of Software Engineering Technology, Department of Computer and Informatics Engineering, Politeknik Negeri Medan, Medan,* Indonesia.
[2] *Department of Multimedia Graphic Engineering Technology, Department of Computer and Informatics Engineering, Politeknik Negeri Medan, Medan,* Indonesia.
[3] *Department of Information Management, Southern Taiwan University of Science and Technology, No. 1, Nan-Tai Street, Yongkang Dist., Tainan 71005,* Taiwan
[4] *Faculty Meta and Technology, Department Management and Information System, Sultan Idris Education University, Tanjong Malim,* Malaysia

DOI : 10.62123/enigma.v4i1.235

Received : August 29, 2026
Revised : September 22, 2026
Accepted : September 26, 2026

Keywords:
CVSS, OWASP Top 10, STRIDE, Threat modeling, Vocational education

ABSTRACT

Digital showcase platforms have become central to how vocational higher education institutions publish student portfolios and project artifacts. These platforms combine high public visibility with the storage of personal data, authentication credentials, and project metadata, which widens their attack surface. Existing security studies concentrate on large academic universities and mostly report tool based test results, so a conceptual threat model tailored to the showcase archetype in vocational education is not yet available. This study formulates a STRIDE based threat taxonomy for digital showcase platforms in vocational higher education. A reference data flow diagram of the platform archetype is decomposed into processes, data stores, external entities, and trust boundaries, and the STRIDE method is applied to each element to elicit threat classes. Each threat class is mapped to the OWASP Top 10:2025 categories and assigned a representative severity using the CVSS v3.1 base metric. The resulting taxonomy organizes twenty-one threat classes across the six STRIDE dimensions and links them to concrete platform assets. The taxonomy supports secure by design decisions for platform maintainers and provides a structured basis for subsequent controlled testing and expert validation.

1. INTRODUCTION

Digital transformation across vocational higher education has produced a class of web platforms that publish student work as digital portfolios. A showcase platform of this kind acts as an academic display window, yet it also operates as a data holding system that stores student personal data, authentication credentials, and project metadata. The public facing nature of such a platform raises its strategic value and, at the same time, exposes a broad attack surface. The showcase platform of the D4 Software Engineering Technology study program at Politeknik Negeri Medan, reachable at showcase.trpl.polmed.ac.id, illustrates this profile. It hosts ninety five projects across six categories and has attracted more than seven hundred thousand views, which signals both visibility and exposure that warrant deliberate security treatment.

The threat environment for the education sector has intensified. Systematic evidence indicates that higher education institutions face a wide spectrum of threat actors and events while operating under tight security budgets and staffing constraints [1]. National level analysis frames the protection of higher education as a policy challenge rather than a purely technical one [2], and studies of universities show fragmented practices in threat intelligence sharing [3]. Surveys of cybersecurity in education further document a persistent gap between rising exposure and institutional readiness [4]. Vocational institutions inherit these pressures with additional characteristics of their own, including heavy student involvement in platform development, rapid contributor turnover as cohorts graduate, and limited dedicated security resources.

Prior work on academic platform security has largely applied vulnerability assessment and penetration testing. Automated multi scanner pipelines have been proposed to broaden coverage of web application weaknesses [5], and empirical comparisons of penetration testing tools give practitioners guidance on tool selection [6]. Comparative studies of web application security parameters [7] and of static and dynamic testing approaches [8] map the tooling landscape, while a penetration taxonomy organizes the testing process, standards, and scoring methods [9]. These studies concentrate on large academic universities and report descriptive, tool centric findings. A conceptual model that explains which threats matter for the vocational showcase archetype, and why, is still missing.

*Corresponding Author Email: d.sanjaya@polmed.ac.id

Threat modeling offers a complementary, design phase perspective. It is a systematic activity that identifies and prioritizes threats before deployment, and STRIDE is among the most widely adopted categorization schemes for this purpose [10], [11]. STRIDE reasons over a data flow diagram, evaluating each process, data store, external entity, and data flow against six threat dimensions, namely Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, and Elevation of privilege. Recent work extends STRIDE to complex and cyber physical settings and combines it with quantitative reasoning [12]. Applying this lens to the showcase archetype yields a structured account of threats that a purely tool driven scan does not provide.

Severity quantification connects elicited threats to prioritization. The Common Vulnerability Scoring System is the dominant metric for expressing the severity of a weakness, and data driven assessment and prioritization of vulnerabilities is an active research area [13]. CVSS scores can be predicted from open sources [14] and combined into cyber risk models [15], and the structure of vulnerability databases has been mapped systematically [16]. Rule based automation has been used to quantify the risk of injection attacks with CVSS metrics [17], and related detection work targets server side cross site scripting [18]. Hybrid scoring systems refine prioritization [19], while critical analyses question how CVSS is applied in practice [20]. These strands motivate a taxonomy that ties each threat to a transparent severity estimate.

This study addresses the gap through three contributions. First, it defines a reference data flow diagram of the digital showcase platform archetype in vocational higher education. Second, it formulates a STRIDE based threat taxonomy in which each threat class is mapped to the OWASP Top 10:2025 catalogue and annotated with a representative CVSS v3.1 base severity. Third, it discusses vocational specific implications and candidate mitigations, and positions the artifact for later empirical validation. The work sits at Technology Readiness Level one to three as fundamental research, and it initiates a security roadmap for academic vocational platforms.

2. LITERATURE REVIEW

2.1 Cybersecurity in Higher Education

Research on cybersecurity in higher education provides the sector context for this study. A systematic review synthesizes mission critical assets, common threat events, threat actors, and vulnerabilities, and it proposes a generic threat model for the sector [1]. Complementary analysis argues that securing higher education is a national policy concern because institutional risk aggregates into systemic exposure [2]. A study of universities shows that threat intelligence sharing is often partial and difficult to manage across autonomous units [3], and a survey of education oriented cybersecurity activities reports uneven awareness and curriculum coverage [4]. These works establish that education platforms face determined adversaries under resource constraints, yet they treat the institution at large rather than a specific application archetype.

2.2 Web Application Vulnerability Assessment and Penetration Testing

A second body of work evaluates web application security through assessment and testing. Automated penetration testing that orchestrates several scanners aims to reduce blind spots and false negatives [5], and empirical comparisons rank tools by detection capability and usability [6]. Broader studies characterize web application security parameters and defensive tactics [7], compare static and dynamic testing for coverage and effort [8], and organize the testing domain into a penetration taxonomy of processes, standards, tools, and scoring methods [9]. This literature is practical and tool centric. It reports what a scan finds on a given system, but it does not explain, at the level of a system model, which threat classes are intrinsic to a showcase platform and how they relate to platform assets.

2.3 Threat Modeling and STRIDE

Threat modeling research supplies the method used here. A systematic review consolidates threat modeling approaches and confirms STRIDE as a dominant categorization scheme built on data flow diagrams [10]. Reviews of threat modelling for advanced persistent threat style attacks catalog approaches and their coverage [11], and hybrid threat models bring quantitative reasoning to complex systems [12]. These studies validate STRIDE as a structured, design phase technique, and they motivate its transfer to the vocational showcase context, where design phase reasoning is valuable because contributor turnover erodes institutional memory of security decisions.

2.4 Vulnerability Scoring with CVSS

A fourth strand concerns severity scoring and prioritization. A survey of data driven vulnerability assessment frames the prioritization problem and its metrics [13]. CVSS values can be predicted from open source intelligence [14] and embedded in cyber risk prediction models that use vulnerability records [15], while a systematic mapping study analyzes the structure of vulnerability databases that feed such models [16]. Applied work automates the quantification of injection attack risk using CVSS metrics [17] and develops server side detection for cross site scripting [18]. Hybrid scoring systems improve the ordering of remediation work [19], and critical commentary calls for careful, context aware use of CVSS rather than mechanical reliance on a single number [20]. Taken together, these works support the use of representative CVSS base scores to rank the threat classes in the proposed taxonomy, while acknowledging the need for context when scores inform decisions.

The reviewed literature shows two separated concerns. Sector studies describe risks at the level of the institution, and application studies report tool findings on specific systems. Neither offers a threat model that is specific to the showcase archetype in vocational education and that links elicited threats to a standard risk catalogue and to transparent severities. The present study connects these

concerns by grounding a STRIDE elicitation in a reference data flow diagram, mapping the result to the OWASP Top 10:2025 catalogue, and attaching representative CVSS v3.1 severities.

3. RESEARH METHODS

This study follows a qualitative and design oriented approach that produces a conceptual artifact, namely a threat taxonomy, rather than a set of measured scan results. The showcase platform of the D4 Software Engineering Technology program at Politeknik Negeri Medan serves as the reference instance that grounds the archetype in a concrete system. The procedure consists of six steps, summarized in Table 1.

Table 1. Procedure

Step	Description
1. Asset and context characterization	Identify the platform mission, its user roles, and the sensitive assets it holds, namely student personal data, authentication credentials, project metadata, and uploaded media.
2. Architecture decomposition	Represent the archetype as a reference data flow diagram with processes, data stores, external entities, data flows, and trust boundaries, as shown in Figure 1.
3. STRIDE elicitation	Apply STRIDE to each data flow diagram element, deriving threat classes for Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, and Elevation of privilege.
4. Standard mapping	Map each threat class to the OWASP Top 10:2025 catalogue to align the taxonomy with a widely used industry reference.
5. Representative severity	Assign each threat class a CVSS v3.1 base vector that reflects typical exploit characteristics, and compute the corresponding base score and severity band.
6. Synthesis	Consolidate the results into a taxonomy and derive candidate mitigations and prioritization guidance for platform maintainers.

The threat classes were derived by a systematic element by dimension examination. Each element of the reference data flow diagram was assessed against all six STRIDE dimensions in turn, and a candidate threat was recorded when a dimension produced a plausible security violation given the role of the element and the assets it handles. Candidates that shared a single mechanism were consolidated into one class to avoid duplication, and candidates without a realistic path on the archetype were discarded. This procedure yields the twenty-one classes catalogued in Table 4. Table 2 presents the resulting element by dimension coverage, which makes the derivation reproducible and shows that every element and every dimension was examined.

Table 2. Element by dimension elicitation matrix mapping each data flow diagram element to the STRIDE dimensions and the threat classes it participates in.

DFD element	S	T	R	I	D	E
Web frontend	-	T1	-	I1, I4	D2	-
Application server	S1, S2, S3	T2, T4	R1, R2, R3	I2, I3, I4	D2, D3	E1, E2, E3, E4
Upload and media handler	-	T3	-	-	D1	E3
External entities	S3	-	R3	-	-	-
User and credential store	S1	-	-	I1	-	-
Project metadata store	-	T1, T2, T4	R2	I3	-	-
Media and file store	-	T3	-	-	D1	-

Each threat class was mapped to a single primary OWASP Top 10:2025 category, selected by the dominant security consequence of the class rather than by the mechanism alone. When a class could plausibly belong to more than one category, the category matching its principal impact was chosen and the secondary reading was noted in the discussion. For example, unrestricted file upload is mapped to Injection because its principal risk on this archetype is the execution of attacker controlled content, with Software and Data Integrity Failures as a secondary reading.

Each class was then assigned a CVSS v3.1 base vector that reflects the typical conditions of its exploitation on the archetype. The base score and severity band were computed from that vector with the standard CVSS v3.1 base metric, and both the vector and the score are reported in Table 4 so that every severity is independently verifiable. The scores are representative estimates for the typical realization of each class. They are analytical values derived from a system model and are not measurements obtained from the live platform.

The taxonomy is designed for a two stage empirical validation that is planned as the next phase of the research. In the first stage, controlled and non destructive testing aligned with the OWASP Web Security Testing Guide is applied to the reference instance to confirm the observability and the exploitability of each threat class, using the affected element in Table 4 as the test target and treating the class as confirmed when a corresponding finding is reproduced. In the second stage, a focus group discussion with security practitioners and platform maintainers assesses the completeness of the taxonomy, the correctness of the STRIDE and

OWASP assignments, and the usefulness of the prioritization tiers, with agreement measured across reviewers. Coverage of the classes in the first stage and the level of reviewer agreement in the second stage are the criteria that will indicate whether the taxonomy is empirically adequate.

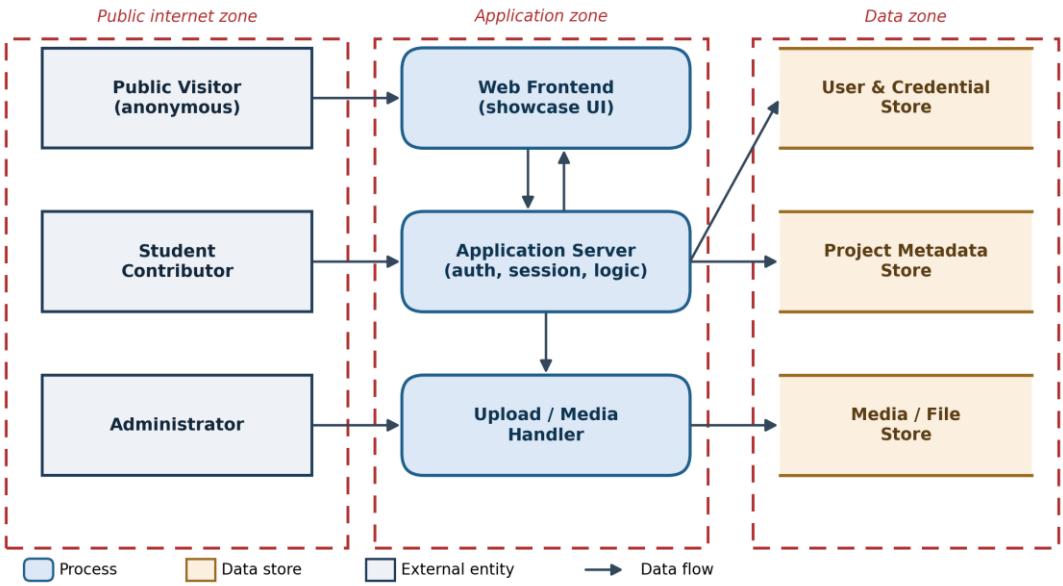


Figure 1. Reference data flow diagram of the digital showcase platform archetype.

Table 3. Elements of the reference data flow diagram and the assets they expose.

Element	Type	Description and key assets
Web frontend	Process	Renders public portfolio pages and forms; primary entry point for anonymous and authenticated requests.
Application server	Process	Performs authentication, session management, access control, and business logic; central mediation point.
Upload and media handler	Process	Accepts and processes project files and images; handles content type and storage operations.
Public visitor	External entity	Anonymous consumer of published portfolios; untrusted source that crosses the public boundary.
Student contributor	External entity	Authenticated author of projects; subject to rapid turnover across cohorts.
Administrator	External entity	Privileged operator of the platform; high value target for privilege abuse.
User and credential store	Data store	Holds account records and authentication secrets; highest confidentiality demand.
Project metadata store	Data store	Holds project descriptions and ownership records; integrity sensitive.
Media and file store	Data store	Holds uploaded artifacts; exposed to unrestricted upload and resource abuse.

4. RESULT AND DISCUSSIONS

This section presents the reference data flow diagram and its elements, the STRIDE based threat taxonomy, and a discussion that relates the taxonomy to the OWASP Top 10:2025 shift, to the characteristics of vocational education, and to its regulatory and educational implications.

4.1 Reference Data Flow Diagram

Figure 1 shows the reference data flow diagram of the showcase archetype. Three external entities interact with the platform, namely an anonymous public visitor, a student contributor, and an administrator. Three processes handle interaction and logic, namely the web frontend, the application server that performs authentication, session handling, and business logic, and the upload and media handler. Three data stores hold the user and credential records, the project metadata, and the uploaded media. Two trust boundaries separate the public internet from the application zone and the application zone from the data zone. Table 3 lists these elements and the assets they expose.

4.2 STRIDE-Based Threat Taxonomy

Table 4 presents the STRIDE based threat taxonomy. Twenty-one threat classes are grouped under the six STRIDE dimensions. Each class names the affected data flow diagram element, the corresponding OWASP Top 10:2025 category, and a representative CVSS v3.1 base score with its severity band. Figure 2 summarizes the base scores by threat class, which supports relative prioritization at a glance.

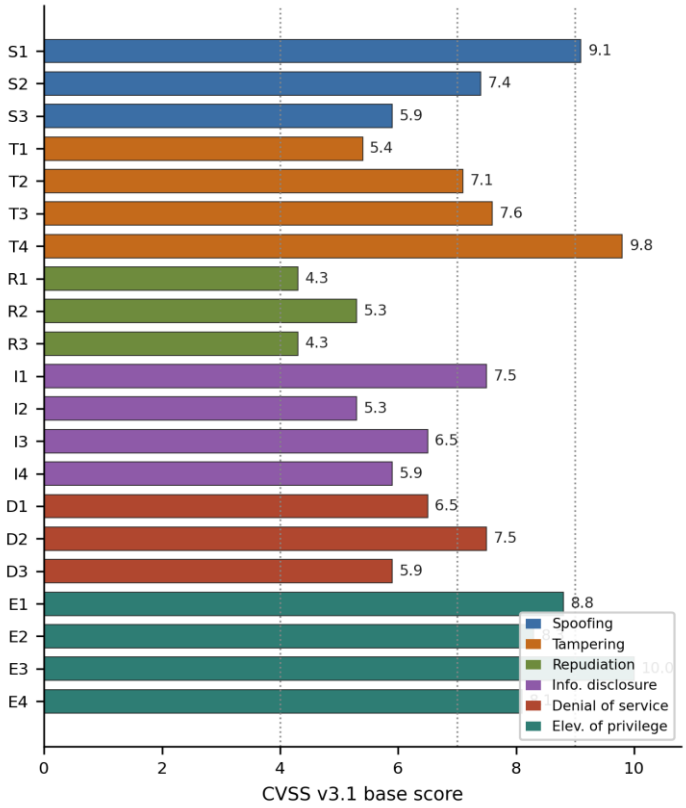


Figure 2. Representative CVSS v3.1 base scores of the twenty-one threat classes, grouped by STRIDE dimension.

Table 4. STRIDE-based threat taxonomy for the digital showcase platform archetype, with OWASP Top 10:2025 mapping and representative CVSS v3.1 base severity.

ID	Threat class	Affected element	OWASP Top 10:2025	CVSS v3.1 vector	CVSS	Severity
Spoofing (S)						
S1	Credential stuffing against student and administrator login	Application server; user store	A07:2025 Authentication Failures	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N	9.1	Critical
S2	Session hijacking via insecure session token handling	Application server	A07:2025 Authentication Failures	AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N	7.4	High
S3	Forged external identity through weak federated or reset flow	External entity; application server	A07:2025 Authentication Failures	AV:N/AC:H/PR:N/UI:R/S:U/C:H/I:L/A:N	5.9	Medium
Tampering (T)						
T1	Stored cross site scripting through project description or metadata	Web frontend; metadata store	A05:2025 Injection	AV:N/AC:L/PR:L/UI:R/S:C/C:L/I:L/A:N	5.4	Medium

T2	Object reference tampering to modify another student project	Application server; metadata store	A01:2025 Broken Access Control	AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:H/A:N	7.1	High
T3	Unrestricted file upload altering stored artifacts	Upload handler; media store	A05:2025 Injection	AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:H/A:L	7.6	High
T4	SQL injection that modifies portfolio records	Application server; metadata store	A05:2025 Injection	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H	9.8	Critical
Repudiation (R)						
R1	Insufficient logging of portfolio changes prevents attribution	Application server	A09:2025 Logging and Monitoring Failures	AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:L/A:N	4.3	Medium
R2	Audit trail tampering removes evidence of actions	Application server; metadata store	A08:2025 Software and Data Integrity Failures	AV:N/AC:H/PR:L/UI:N/S:U/C:N/I:H/A:N	5.3	Medium
R3	Shared or ambiguous accounts blur accountability under turnover	External entity; application server	A09:2025 Logging and Monitoring Failures	AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:L/A:N	4.3	Medium
Information disclosure (I)						
I1	Exposure of student personal data through public profiles	Web frontend; user store	A01:2025 Broken Access Control	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N	7.5	High
I2	Verbose error messages leak stack traces and internal paths	Application server	A10:2025 Mishandling of Exceptional Conditions	AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N	5.3	Medium
I3	Object reference exposure of unpublished projects	Application server; metadata store	A01:2025 Broken Access Control	AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N	6.5	Medium
I4	Weak or missing transport encryption exposes credentials	Web frontend; application server	A04:2025 Cryptographic Failures	AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:N/A:N	5.9	Medium
Denial of service (D)						
D1	Resource exhaustion through unthrottled uploads	Upload handler; media store	A06:2025 Insecure Design	AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	6.5	Medium
D2	Application layer flooding of search and listing endpoints	Web frontend; application server	A06:2025 Insecure Design	AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H	7.5	High
D3	Availability loss from outdated third party components	Application server	A03:2025 Software Supply Chain Failures	AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:H	5.9	Medium

Elevation of privilege (E)						
E1	Vertical escalation from student role to administrator functions	Application server	A01:2025 Broken Access Control	AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H	8.8	High
E2	Object reference access to administrative endpoints	Application server	A01:2025 Broken Access Control	AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:L	8.3	High
E3	Remote code execution via injection or malicious upload	Application server; upload handler	A05:2025 Injection	AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H	10.0	Critical
E4	Privilege escalation through a vulnerable third party component	Application server	A03:2025 Software Supply Chain Failures	AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H	8.1	High

4.3 Alignment with the OWASP Top 10:2025 shift

The taxonomy reflects the structural change introduced by the OWASP Top 10:2025 catalogue, which is the first revision since 2021. Broken access control remains the dominant risk and now absorbs server side request forgery, which is consistent with the prominence of access control threats in the taxonomy, spanning tampering, information disclosure, and elevation of privilege. Security misconfiguration rises in the catalogue, and the taxonomy expresses this through information disclosure that follows from verbose error handling. Two new categories appear directly in the model. Software supply chain failures explain availability loss and privilege escalation that originate in outdated or compromised third party components, and mishandling of exceptional conditions captures the leakage of stack traces and internal paths through unhandled errors. The presence of these classes in a small academic platform shows that the systemic and operational concerns emphasized by the 2025 catalogue are relevant well beyond large enterprises.

4.4 Vocational specific characteristics

Three features of the vocational showcase archetype shape its risk profile. High public visibility increases the incentive for opportunistic scanning and defacement, which raises the practical likelihood of the injection and access control classes. Rapid contributor turnover, as cohorts graduate, degrades accountability and elevates the repudiation classes and shared account risks, and it also weakens the continuity of security decisions taken during development. Limited dedicated security staffing makes design phase controls more attractive than reactive testing, because a taxonomy that is embedded in the platform design survives personnel change better than tacit knowledge. These characteristics justify a secure by design treatment for platforms that might otherwise be considered low value.

The personal data classes in the taxonomy also carry legal weight that extends beyond Indonesia. Exposure of student personal data through public profiles (I1), object reference exposure of unpublished projects (I3), and weak transport encryption (I4) each describe a failure that data protection law treats as a breach of the controller's security obligation. In Indonesia, Law No. 27 of 2022 on Personal Data Protection requires controllers to secure personal data and to notify data subjects and the supervisory authority of a protection failure within three days [21]. Taiwan's Personal Data Protection Act requires agencies to adopt security maintenance measures and to notify data subjects when personal data is stolen, disclosed, or altered [22]. Malaysia's Personal Data Protection Act 2010 sets a security principle for data users, and its 2024 amendment introduced mandatory notification of personal data breaches [23]. The three regimes differ in scope, particularly in how they apply to public institutions, yet they converge on two duties that the taxonomy makes actionable: preventing unauthorized disclosure through access control and encryption, and maintaining the logging needed to detect and report an incident, which links the disclosure classes to the repudiation classes R1 and R3. A maintainer who addresses these classes therefore reduces technical risk and regulatory exposure at the same time, which strengthens the case for treating showcase platforms as regulated data systems across the region.

The taxonomy also has value as a teaching artifact. In vocational programs, students are often the developers and maintainers of the showcase platform, so the system they secure is the system they build. The element by dimension matrix in Table 2 can serve as a structured exercise in which each cohort decomposes a familiar platform, elicits threats with STRIDE, and justifies severities with CVSS, an activity aligned with curriculum designs that favor applied security practice [4]. Embedding this exercise in coursework turns the taxonomy into a mechanism for transferring security knowledge between cohorts, which directly addresses the continuity problem caused by contributor turnover.

Table 6. Multi-browser compatibility testing results

Aspect tested	Chrome	Firefox	Edge	Brave
Plugin loading	Success	Success	Success	Success

Display rendering	Clear	Clear	Clear	Clear
JavaScript execution	Success	Success	Success	Success
Keyboard navigation	Success	Success	Success	Success
Focus indicator	Clear	Clear	Clear	Clear
Voice command	Stable	Failed	Unstable	Failed

Display, text, and keyboard features behaved consistently across the four browsers, so the larger part of the plugin can be regarded as satisfying the robust principle. The differences are concentrated in voice command. Firefox and Brave do not yet support the speech recognition interface of the Web Speech API, which caused the feature to fail, while Edge produced unstable responses even though recognition was available. This pattern reinforces the observation that differing implementations across rendering engines remain a source of inconsistency in modern web features [3][4], and it indicates that dependence on built-in browser APIs is the principal weakness of the proposed design. The engine coverage of this test is itself partial. Chrome, Edge, and Brave are all Blink based, so the four browsers in fact represent only two engines, Blink and Gecko. WebKit based browsers such as Safari were not examined, and their handling of speech recognition, CSS filters, and focus outlines may differ from what is reported here. Work on responsive layout failures indicates that such divergences tend to surface in layout rather than in script execution [20], so the display features that appeared stable here are precisely the ones that would need re-examination on WebKit.

4.5 Prioritization and mitigations

The representative severities translate into a tiered plan that a maintainer can execute within limited institutional resources. The tiers order work by severity and by the effort each control demands, so a small team can act on the highest return items first and defer hardening tasks without leaving the platform exposed.

Tier one addresses the critical classes, namely SQL injection, remote code execution, and credential stuffing (S1, T4, E3). These carry the largest base scores and the lowest exploitation cost, so they are handled first through parameterized queries, strict server side input validation, upload restrictions that enforce content type and size, rate limiting, and protection against automated credential abuse. Tier two addresses the high severity access control and escalation classes (T2, I1, D2, E1, E2, E4) through centralized authorization checks and the removal of insecure direct object references, which a single reviewer can verify endpoint by endpoint. Tier three addresses the medium severity information disclosure, repudiation, and supply chain classes through disciplined error handling that avoids leaking internal detail, transport encryption, tamper resistant logging that preserves attribution across contributor turnover, and a dependency inventory with timely updates. This ordering gives a resource constrained maintainer a defensible sequence of work that maps each action to a named threat class and to a concrete control.

The representative severities for the injection classes can be refined with automated quantification. Prior work shows that the CVSS base metrics of injection attacks can be derived through rule based automation [17], which offers a route to replace the analytical estimates for SQL injection (T4, 9.8) and remote code execution (E3, 10.0) with scores computed from observed exploit conditions during the planned testing stage. For stored cross site scripting (T1, 5.4), server side detection based on HTML semantic parsing [18] complements output encoding by intercepting malicious payloads before they reach other users of the platform. Because CVSS expresses technical severity rather than organizational risk [20], these refinements should be combined with the environmental context of each deployment before they inform remediation decisions.

4.6 Comparison with prior work and limitations

Compared with tool centric assessments of academic platforms [5], [6], [8], the taxonomy explains why particular threats arise from the platform structure and links them to a standard catalogue and to transparent severities, rather than reporting an inventory of scan hits. It complements the sector level threat model for higher education [1] by descending to a specific application archetype. Several limitations apply. The taxonomy describes a single archetype grounded in one reference instance, so generalization to other platform designs requires care. The CVSS values are representative base scores rather than measurements, and environmental factors may raise or lower them for a specific deployment. The artifact has not yet been validated empirically. These limitations define the immediate next steps of the research.

5. CONCLUSSION

The contribution of this study is a conceptual artifact. The taxonomy, its mapping to the OWASP Top 10:2025 catalogue, and the representative severities are analytical results derived from a system model, and they are separate from the empirical validation identified below as future work. This study formulated a STRIDE based threat taxonomy for digital showcase platforms in vocational higher education. A reference data flow diagram of the archetype was decomposed and analyzed with STRIDE, and the resulting twenty-one threat classes were mapped to the OWASP Top 10:2025 catalogue and annotated with representative CVSS v3.1 base severities. The taxonomy makes explicit which threats are intrinsic to a publicly visible student portfolio platform, how they relate to platform assets, and how they should be prioritized under the resource constraints typical of vocational institutions. It also shows that the systemic and operational concerns emphasized by the 2025 catalogue, including software supply chain failures and the mishandling of exceptional conditions, are present in small academic systems.

The taxonomy provides a secure by design reference that is more durable than tacit knowledge in an environment of rapid contributor turnover. Future work will validate the artifact through controlled and non destructive testing aligned with the OWASP Web Security Testing Guide and through expert focus group discussion, extend the archetype to related academic platforms, and

refine the severity estimates with environmental context. These steps advance the broader roadmap toward applied security engineering for academic vocational platforms

ACKNOWLEDGEMENT

The authors thank Politeknik Negeri Medan, through its Center for Research and Community Service (P3M), for funding this research under the Fundamental Research scheme, contract number B/320/PL5/PT.01.05/2026. The authors also thank the management of the Software Engineering Technology study program for permission to use the program's showcase platform as the reference instance. They are grateful to the cybersecurity experts who took part in the focus group discussion and to the five student research assistants of the Software Engineering Technology program, whose feedback and support helped shape the taxonomy. Finally, the authors thank the anonymous reviewers for constructive comments that improved the clarity and reproducibility of this work.

REFERENCES

[1] J. B. Ulven and G. Wangen, "A systematic review of cybersecurity risks in higher education," Future Internet, vol. 13, no. 2, p. 39, 2021, doi: 10.3390/fi13020039.

[2] N. S. Fouad, "Securing higher education against cyberthreats: from an institutional risk to a national policy challenge," Journal of Cyber Policy, vol. 6, no. 2, pp. 137-154, 2021, doi: 10.1080/23738871.2021.1973526.

[3] A. Piazza, S. Vasudevan, and M. Carr, "Cybersecurity in UK universities: mapping (or managing) threat intelligence sharing within the higher education sector," Journal of Cybersecurity, vol. 9, no. 1, p. tyad019, 2023, doi: 10.1093/cybsec/tyad019.

[4] M. Ismail, N. T. Madathil, M. Alalawi, S. Alrabaee, M. Al Bataineh, S. Melhem, and D. Mouheb, "Cybersecurity activities for education and curriculum design: a survey," Computers in Human Behavior Reports, vol. 16, p. 100501, 2024, doi: 10.1016/j.chbr.2024.100501.

[5] K. Abdulghaffar, N. Elmrabit, and M. Yousefi, "Enhancing web application security through automated penetration testing with multiple vulnerability scanners," Computers, vol. 12, no. 11, p. 235, 2023, doi: 10.3390/computers12110235.

[6] M. Albahar, D. Alansari, and A. Jurcut, "An empirical comparison of pen-testing tools for detecting web app vulnerabilities," Electronics, vol. 11, no. 19, p. 2991, 2022, doi: 10.3390/electronics11192991.

[7] 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, p. 4077, 2022, doi: 10.3390/app12084077.

[8] S. Qadir, E. Waheed, A. Khanum, and S. Jehan, "Comparative evaluation of approaches and tools for effective security testing of web applications," PeerJ Computer Science, vol. 11, p. e2821, 2025, doi: 10.7717/peerj-cs.2821.

[9] K. U. Sarker, F. Yunus, and A. Deraman, "Penetration taxonomy: a systematic review on the penetration process, framework, standards, tools, and scoring methods," Sustainability, vol. 15, no. 13, p. 10471, 2023, doi: 10.3390/su151310471.

[10] S. M. Khalil, H. Bahsi, and T. Korõtko, "Threat modeling of industrial control systems: a systematic literature review," Computers & Security, vol. 136, p. 103543, 2024, doi: 10.1016/j.cose.2023.103543.

[11] M. Tatam, B. Shanmugam, S. Azam, and K. Kannoorpatti, "A review of threat modelling approaches for APT-style attacks," Heliyon, vol. 7, no. 1, p. e05969, 2021, doi: 10.1016/j.heliyon.2021.e05969.

[12] F. Valenza, E. Karafili, R. V. Steiner, and E. C. Lupu, "A hybrid threat model for smart systems," IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 5, pp. 4403-4417, 2023, doi: 10.1109/TDSC.2022.3213577.

[13] T. H. M. Le, H. Chen, and M. A. Babar, "A survey on data-driven software vulnerability assessment and prioritization," ACM Computing Surveys, vol. 55, no. 5, pp. 1-39, 2022, doi: 10.1145/3529757.

[14] P. Kühn, D. N. Relke, and C. Reuter, "Common vulnerability scoring system prediction based on open source intelligence information sources," Computers & Security, vol. 131, p. 103286, 2023, doi: 10.1016/j.cose.2023.103286.

[15] A. Negahdari Kia, F. Murphy, B. Sheehan, and D. Shannon, "A cyber risk prediction model using common vulnerabilities and exposures," Expert Systems with Applications, vol. 237, p. 121599, 2024, doi: 10.1016/j.eswa.2023.121599.

[16] X. Li, S. Moreschini, Z. Zhang, F. Palomba, and D. Taibi, "The anatomy of a vulnerability database: a systematic mapping study," Journal of Systems and Software, vol. 201, p. 111679, 2023, doi: 10.1016/j.jss.2023.111679.

[17] A. Kurniawan, M. Y. Darus, M. A. Mohd Ariffin, Y. Muliono, and C. R. Pardomuan, "Automation of quantifying security risk level on injection attacks based on common vulnerability scoring system metric," Pertanika Journal of Science and Technology, vol. 31, no. 3, pp. 1245-1265, 2023, doi: 10.47836/pjst.31.3.07.

[18] C. R. Pardomuan, A. Kurniawan, M. Y. Darus, M. A. Mohd Ariffin, and Y. Muliono, "Server-side cross-site scripting detection powered by HTML semantic parsing inspired by XSS Auditor," Pertanika Journal of Science and Technology, vol. 31, no. 3, pp. 1353-1377, 2023, doi: 10.47836/pjst.31.3.14.

[19] A. Sharma, S. Sabharwal, and S. Nagpal, "A hybrid scoring system for prioritization of software vulnerabilities," Computers & Security, vol. 129, p. 103256, 2023, doi: 10.1016/j.cose.2023.103256.

[20] J. Spring, E. Hatleback, A. Householder, A. Manion, and D. Shick, "Time to change the CVSS?," IEEE Security & Privacy, vol. 19, no. 2, pp. 74-78, 2021, doi: 10.1109/MSEC.2020.3044475.

[21] Republic of Indonesia, Law No. 27 of 2022 on Personal Data Protection, Jakarta, Indonesia, 2022.

[22] Republic of China (Taiwan), Personal Data Protection Act, as amended Nov. 2025, Taipei, Taiwan, 2025.

[23] Malaysia, Personal Data Protection Act 2010 (Act 709), as amended by the Personal Data Protection (Amendment) Act 2024 (Act A1727), Kuala Lumpur, Malaysia, 2024.