

Lowering the Infrastructure Barrier to Hands-On Cybersecurity Education: Design and Feasibility of Kinetic Breach, a Browser-Based Gamified Investigation Simulator

Husam Abdulatef Ahmed Yousef Harpah¹, Muhammad Faheem Mohd Ezani^{1}, Nuridawati Mustafa¹,
Mohd Aizat Yaacob²

¹Faculty of Information and Communication Technology (FTMK), Universiti Teknikal Malaysia
Melaka (UTeM), Melaka, Malaysia

²VireServe Sdn. Bhd., Petaling Jaya, Selangor, Malaysia

*Corresponding Author

DOI: <https://doi.org/10.47772/IJRISS.2026.100900253>

Received: 20 September 2026; Accepted: 25 September 2026; Published: 07 October 2026

ABSTRACT

Hands-on investigation practice is central to developing cybersecurity competencies such as log analysis and system examination, yet the realistic environments that provide it, such as cyber ranges, virtual-machine laboratories, and container testbeds, impose infrastructure, configuration, and maintenance costs that place practical experience beyond the easy reach of many beginners and resource-constrained institutions. This paper addresses that access problem through the design and technical feasibility evaluation of Kinetic Breach, a browser-delivered platform for practising cybersecurity investigation skills that requires no virtual machine, container, or laboratory software on the learner's part and only a single conventional web application on the institution's self-hosted server. Following a design-science approach, the platform integrates a simulated Linux-like investigation environment, a discovery-based assessment model that admits multiple valid evidentiary paths rather than a single scripted solution, a gamification layer to sustain engagement, and an adaptive AI guidance component that offers progressive hints without disclosing solutions and whose output is kept deliberately separate from the platform's deterministic scoring. Feasibility was assessed through systematic software verification comprising fifty-five automated tests across unit, integration, end-to-end, security, and load classes. All fifty-five tests passed; a complete investigation lifecycle executed correctly through the discovery, scoring, and progression pipeline; the tested application-level security controls behaved as intended; and the system remained stable under modest concurrency within a local environment. These results establish that an accessible, infrastructure-light, gamified, AI-guided investigation trainer is buildable and behaves as designed. The study makes no claim regarding usability, engagement, or learning gains, which require evaluation with human participants and are identified as the primary direction for future work. Kinetic Breach is thus offered as an accessible on-ramp to hands-on cybersecurity investigation and as a design foundation for subsequent educational evaluation.

Keywords: cybersecurity education, hands-on learning, gamification, AI-assisted tutoring, educational accessibility

INTRODUCTION

The accelerating digitalisation of public and private services has made the protection of information systems a core societal concern, and with it the demand for a workforce able to detect, analyse, and investigate security incidents. Industry assessments continue to report a substantial shortfall of qualified practitioners, with global demand estimated at roughly 10.2 million professionals against a workforce of about 5.5 million (ISC2, 2024). As important as the size of this gap is its character: employers increasingly value demonstrable, hands-on competence over theoretical credentials alone, particularly for defensive skills such as log analysis, system

examination, and digital-evidence discovery. These competencies are procedural and investigative; they are acquired through practice rather than through exposition.

This creates a persistent tension in cybersecurity education. Experiential learning theory holds that durable understanding develops through concrete experience and active experimentation rather than passive reception (Kolb, 1984), yet much foundational instruction remains centred on lectures, readings, and written assessment. Learners may therefore grasp incident-response concepts in the abstract while having little opportunity to perform an investigation: to read a log, follow an evidentiary trail, and reason toward a conclusion. The gap is widest for beginners, who often have limited command-line fluency and little prior exposure to system artefacts.

Practical training environments have been developed to close this gap. Cyber ranges, virtual-machine laboratories, and container-based testbeds provide realistic, hands-on experience and are now widely used in higher education (Beuran et al., 2018; Vykopal et al., 2021). Their pedagogical value is not in dispute; their accessibility is. Such environments typically demand considerable infrastructure, specialised configuration expertise, and continuing maintenance, and the cost of developing and operating them remains high (Chouliaras et al., 2023). For institutions with constrained budgets, and for individual beginners without access to managed laboratory resources, these requirements constitute a barrier to entry rather than a pathway. The result is an equity problem as much as a technical one: the learners who would benefit most from low-stakes practice are often those least able to reach the environments that provide it.

Two further difficulties compound the access problem. First, sustained investigation is cognitively demanding, and conventional delivery can struggle to hold learner engagement through it. Gamification, which introduces structured objectives, scoring, progression, and achievements, has been examined across cybersecurity education as a means of raising motivation and participation (Dichev & Dicheva, 2017; Weitzl-Harms et al., 2023). Second, when beginners become stuck during an open-ended investigation, the support available is often poorly calibrated: too little guidance produces frustration and abandonment, while a full walkthrough removes the very reasoning the activity is meant to develop. Intelligent tutoring research has long sought a middle path through adaptive, progressively revealed hints that preserve independent problem-solving (Maniktala et al., 2020a; Mousavinasab et al., 2021), a direction that recent AI-assisted tutors have begun to extend to authentic technical tasks.

This paper responds to the access problem directly. It presents the design and technical feasibility of Kinetic Breach, a browser-based, gamified simulation for practising cybersecurity investigation skills that requires no virtual machine, container, or dedicated laboratory infrastructure on the learner's part. The platform provides a simulated Linux-like workstation in which a learner investigates a security incident by examining logs and system artefacts through a command-line interface, discovers evidence through multiple valid investigative paths rather than a single scripted sequence, documents findings in an investigation report, and receives structured feedback. Engagement is supported through a gamification layer, and contextual assistance is provided by an adaptive AI guidance component, ARIA, designed to offer progressive hints without disclosing solutions.

The contribution is deliberately scoped as one of design and feasibility rather than of measured educational effectiveness. The study's objectives are, first, to present an accessible, infrastructure-light design that integrates simulated investigation, discovery-based assessment, gamification, and non-revealing adaptive guidance in a single browser-delivered artefact; and second, to establish the technical feasibility and stability of that design through systematic verification. Claims regarding learning gains, usability, or learner satisfaction fall outside the present scope and are identified as future work, since establishing them requires evaluation with human participants that this study did not undertake.

The remainder of the paper is organised as follows. The next section reviews related work on hands-on cybersecurity education, gamified learning, and AI-assisted guidance, and positions the present contribution within it. The methodology section describes the system design and the verification approach. The results section reports the feasibility evidence obtained. The discussion considers the implications for accessible cybersecurity education together with the study's limitations, and the conclusion summarises the contribution and outlines future evaluation.

LITERATURE REVIEW

This section situates the present work within three bodies of research that bear directly on the accessibility of hands-on cybersecurity education: the design of practical training environments and the infrastructure they demand, the use of gamification to sustain engagement, and the provision of adaptive, AI-assisted guidance during open-ended tasks. It closes by identifying the gap that the proposed system addresses.

Hands-on training environments and their accessibility

The dominant approach to practical cybersecurity education is the provision of realistic, isolated environments in which learners can act without risk to production systems. Integrated frameworks such as CyTRONE (Beuran et al., 2018) and scalable range platforms surveyed by Vykopal et al. (2021) demonstrate the pedagogical strength of this model, supporting authentic exercises that develop procedural competence rather than conceptual recall. Their educational value is well established; the difficulty lies in what they require to operate. Cyber ranges and virtual-machine laboratories are frequently computationally expensive, demand extensive configuration, and carry a continuing maintenance burden, and the cost of developing and sustaining them remains high enough to deter many educational institutions from adopting them independently (Chouliaras et al., 2023; Mills et al., 2024; Nakata & Otsuka, 2021).

A substantial line of work has sought to reduce this overhead through containerisation. Container-based ranges such as CyExec* (Nakata & Otsuka, 2021), GoibhniUWE (Mills et al., 2024), and Cyber Range Lite (Martucci et al., 2026) achieve faster start-up, lower resource consumption, and improved scalability relative to virtual-machine implementations, and represent a meaningful step toward affordability. Even so, these solutions relocate rather than remove the infrastructure requirement: they still depend on server-side orchestration, container tooling, and administrative expertise, and access is typically mediated through virtual private network tunnels or managed deployments. For an individual beginner working from an ordinary personal computer, or an institution without the means to run and maintain such a back end, a barrier to entry remains. Moreover, assessment within many of these environments follows the capture-the-flag convention, in which a challenge is considered solved when a single hidden token is retrieved. This is an efficient but narrow model that rewards reaching a predetermined answer rather than the quality of the investigative process that leads there.

Gamification and engagement in cybersecurity education

Because sustained technical investigation is cognitively demanding, engagement has become a central concern in cybersecurity pedagogy. Gamification, the use of game elements such as goals, scoring, progression, and achievements in non-game contexts, is grounded in motivational theory, particularly self-determination theory's emphasis on competence and autonomy (Ryan & Deci, 2000), and has been widely reviewed as a means of raising interest and participation in learning (Dichev & Dicheva, 2017; Seaborn & Fels, 2015). Within cybersecurity specifically, a systematic mapping by Weitz-Harms et al. (2023) catalogued a large and growing body of gamified applications for undergraduate education, while a multi-study, four-semester evaluation by Kim et al. (2025) provided empirical evidence that gamified laboratories can improve learners' motivation and career interest relative to conventional formats. Recent work has further connected gamification to the equity dimension that motivates the present study, arguing that inclusively designed game elements can broaden participation among learners who might otherwise disengage from technical material (Zhong et al., 2024). Reviews nonetheless caution that much of the evidence remains uneven and that rigorous, outcome-oriented study is still needed (Subhash & Cudney, 2018). A recurring observation is that gamification is most often layered onto quiz-style or capture-the-flag formats, and less frequently integrated with realistic, open-ended investigation.

AI-assisted guidance and intelligent tutoring

The final strand concerns the support learners receive while working. Intelligent tutoring systems have long pursued adaptive assistance that scaffolds problem-solving without supplanting it, and reviews document the effectiveness of well-calibrated, progressively revealed hints (Mousavinasab et al., 2021), as well as the risk that poorly designed help provokes either dependence or avoidance (Maniktala et al., 2020a). The emergence of large

language models has renewed this agenda. Iris, an LLM-based tutor for programming education, deliberately withholds complete solutions, offering subtle hints and counter-questions to preserve independent reasoning, and was received by students as a complement to human instruction rather than a replacement for it (Bassner et al., 2024). A subsequent randomised controlled trial refined the picture in a way that is directly relevant to how such systems should be claimed: constrained, scaffolded AI support raised intrinsic motivation and lowered frustration and cognitive load, yet did not by itself produce greater learning gains than unrestricted assistance (Bassner et al., 2025). The lesson for design is twofold: guidance should be bounded against solution-giving, and its benefits for engagement should not be conflated with proven gains in achievement.

Positioning of the present work

Read together, these strands reveal a persistent gap. Efforts to make hands-on practice affordable have reduced, but not eliminated, the infrastructure that stands between a beginner and a realistic exercise; assessment in such environments tends to reward a single correct endpoint rather than sound investigative reasoning; and gamification and adaptive guidance, though individually well studied, are seldom combined within one realistic investigation task. The present work addresses this gap by integrating four elements in a single browser-delivered artefact that requires no virtual machine, container, or laboratory infrastructure on the learner's part: a simulated investigation environment; discovery-based assessment that admits multiple valid evidentiary paths; a gamification layer to sustain engagement; and a non-revealing adaptive AI guidance component whose output is kept deliberately separate from the deterministic scoring of the learner's work. Consistent with the evidence reviewed above, the contribution is advanced as one of accessible design and technical feasibility, with claims about learning outcomes reserved for future evaluation.

METHODOLOGY

This study follows a design-science research approach, in which a purposeful artefact is constructed to address an identified problem and is then evaluated against defined criteria (Hevner et al., 2004; Peffers et al., 2007). The problem, established in the preceding sections, is the infrastructure barrier that limits beginners' access to realistic, hands-on cybersecurity investigation practice. The artefact is Kinetic Breach, a browser-delivered platform for practising log analysis and system investigation. Consistent with the scope declared earlier, the evaluation reported here is a feasibility evaluation conducted through systematic software verification; it establishes that the design functions correctly and stably within its intended boundaries, and does not attempt to measure learning outcomes, which would require study with human participants. The platform was developed iteratively, allowing requirements, design, and implementation to be refined across successive cycles.

System architecture and delivery model

Kinetic Breach adopts a layered client-server architecture organised into presentation, application, service, and data layers, a structure chosen to reduce coupling and to allow new investigation content to be introduced without altering the core processing logic. The presentation layer is a React.js single-page application delivered to a standard web browser; it is the learner's only required software. The application layer exposes an HTTP/REST interface implemented in Node.js and Express.js and coordinates authentication, session management, and communication with the underlying services. The service layer holds the platform's business logic: terminal command processing, evidence discovery and evaluation, learner-behaviour analysis, scoring, progression, and AI-assisted guidance. The data layer uses PostgreSQL, organised into twenty-one relational tables normalised to third normal form, to persist accounts, sessions, command history, discoveries, progression, achievements, reports, and guidance records.

A defining feature of the design is the separation of persistent learner state from investigation content. Rather than storing complete virtual environments in the database, the platform maintains scenarios as version-controlled content files (base filesystem templates, objectives, discovery definitions, and evidence artefacts), from which the simulated environment is constructed at request time. This content-driven organisation keeps scenarios reproducible, makes new material straightforward to author, and holds the runtime footprint low. Two external services are integrated where required: Google OAuth 2.0 for optional authentication, and a large-language-model provider accessed by the guidance component. Critically, the provider is reached through an

abstraction layer that permits different providers, or a local mock implementation, to be used interchangeably, so that the platform's core functionality does not depend on any single external service.

The significance of this arrangement for the access problem is that it reduces the learner-side requirement to a web browser and the institution-side requirement to a single conventional web application, in contrast to per-learner virtual machines or the server-side orchestration and container tooling that lighter cyber ranges still demand (Mills et al., 2024; Nakata & Otsuka, 2021). Figure 1 presents the layered architecture.

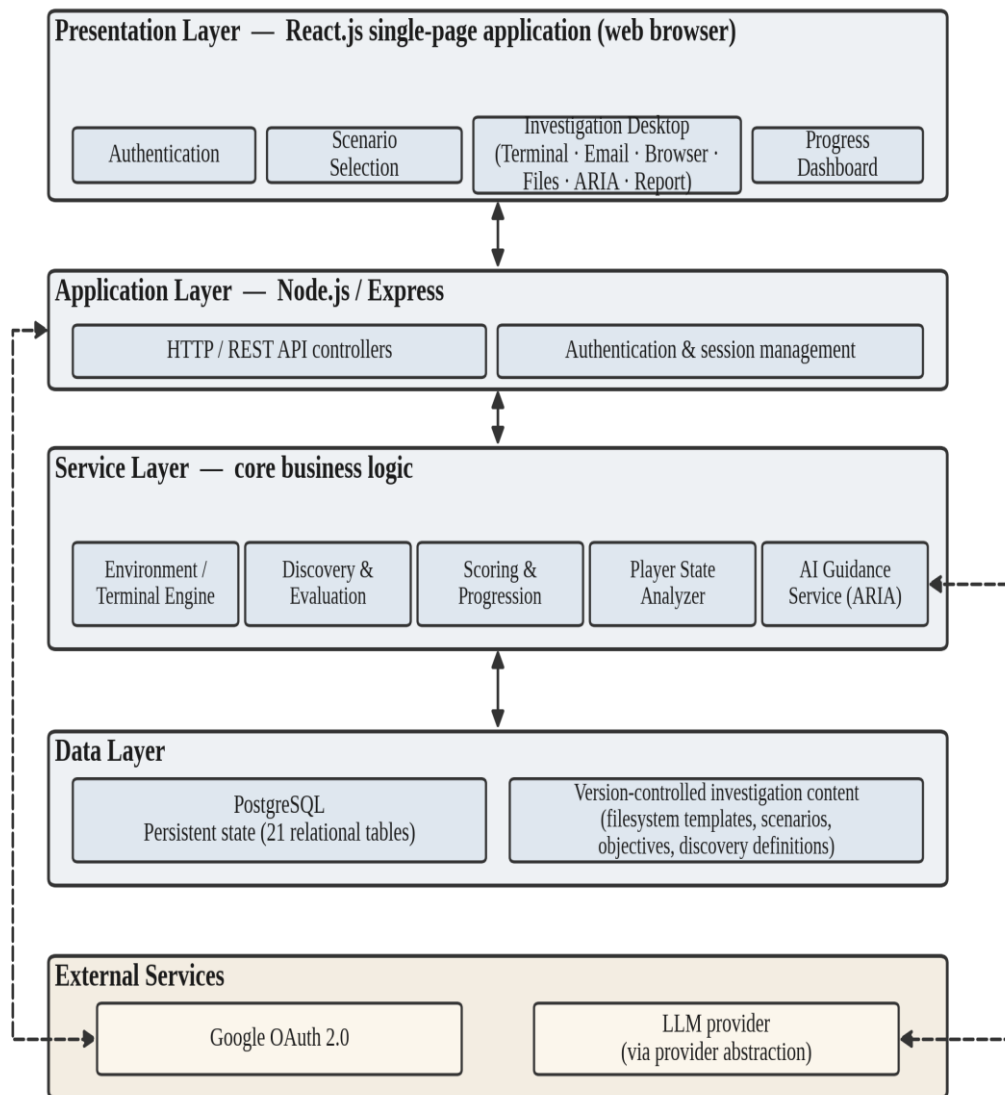


Figure 1: Layered client–server architecture of Kinetic Breach

Simulated investigation environment

Within the browser, learners work inside an Investigation Desktop that integrates the tools needed for a scenario (a command-line terminal, together with email, browser, files, report, and guidance applications), so that the investigation proceeds without navigating away from the workspace. The terminal is the primary instrument: learners issue supported commands to examine logs and system artefacts, and each command is processed through a pipeline that validates the active session, parses and checks the command against the supported syntax, resolves it against the scenario's virtual filesystem and investigation definitions, records the interaction, updates investigation state, and returns simulated output. Because the environment is simulated rather than a live operating system, learner actions remain safely bounded while still exercising authentic investigative reasoning. Figure 2 illustrates the command-processing and guidance pipeline.

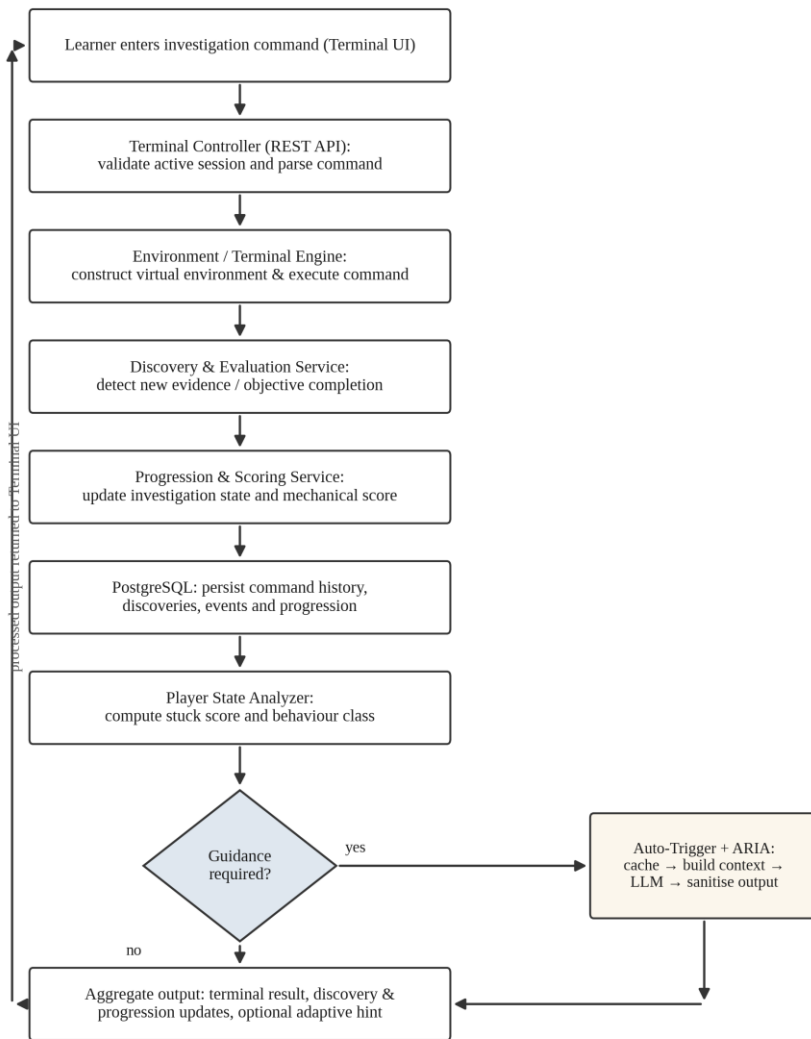


Figure 2: Command-processing and adaptive-guidance pipeline

Discovery-based, multi-path investigation

Assessment of progress departs from the single-solution convention common to capture-the-flag formats. A rule-based Discovery Trigger System evaluates each valid command against a set of discovery definitions attached to the scenario, where every discovery carries one or more trigger conditions expressed over criteria such as command matching, target or path resolution, pattern matching, and file filtering. Because different commands and techniques can satisfy the same trigger, several investigative approaches can reveal the same piece of evidence. Learners are therefore free to pursue multiple valid evidentiary paths toward an objective rather than reproducing a prescribed sequence. This arrangement is consistent with discovery-oriented and experiential conceptions of learning (Kolb, 1984), and it is better suited to the open-ended character of real investigation.

Deterministic assessment and its separation from AI review

Learner performance is quantified by a deterministic, weighted scoring model that evaluates both the effectiveness and the efficiency of the investigation. Investigation progress contributes up to 50 points in proportion to the weighted required discoveries unlocked; command efficiency contributes up to 30 points, measured against a baseline equal to the number of required discoveries and penalised for excess commands; objective completion contributes up to 20 points; and five points are deducted for each hint requested. The result is bounded between 0 and 100 and serves as the authoritative basis for progression, badge qualification, and difficulty-scaled experience points.

A deliberate design decision keeps this mechanical score entirely separate from the platform's qualitative AI review. The AI review provides narrative feedback on the learner's report but is excluded from the numerical progression calculation; the two are never averaged into a single figure. This separation is motivated by evidence that AI assistance can raise task performance without producing commensurate learning gains (Bassner et al., 2025): binding a deterministic, auditable score to the learner's own actions preserves the integrity of assessment, while the AI review contributes formative commentary without silently inflating the record.

Adaptive guidance without solution disclosure

Support during investigation is handled by three cooperating components. A Player State Analyzer combines behavioural indicators (recent commands, progress, unsuccessful attempts, repetition, inactivity, and proximity to expected actions) into a composite stuck score (0–100) and a behavioural classification. An Auto-Trigger Hint System uses this state, guarded by safeguards such as a maximum hint count, cooldown intervals, and a check that a hint has not already been issued for the current step, to decide whether assistance should be offered without the learner having to ask (Maniktala et al., 2020b). When guidance is warranted, the AI Guidance Service, ARIA, resolves an appropriate hint level and, unless a suitable cached hint exists, assembles a context from the learner's recent activity, progress, and current objective and submits it to the language model. The returned response is processed and sanitised to prevent disclosure of solutions or expected commands before it is delivered and logged.

This design realises the calibrated middle ground identified in the literature: assistance that scaffolds without solving, which intelligent-tutoring and recent LLM-tutor research associate with sustained independent problem-solving (Bassner et al., 2024; Maniktala et al., 2020a; Mousavinasab et al., 2021). Hint caching additionally limits repeated model calls, and, together with the provider abstraction, moderates the operating cost of the guidance feature.

Gamification and progression

To sustain engagement through demanding investigation, a gamification layer converts performance into experience points, levels, ranks, clearance tiers, badges, and dynamically derived achievement and investigation-style profiles. Experience is awarded from the mechanical score scaled by scenario difficulty, with an independence bonus for completing a scenario without hints. This incentive is aligned with the platform's emphasis on self-directed reasoning. These elements draw on the motivational rationale for gamification in cybersecurity education and its documented association with learner engagement and interest (Kim et al., 2025; Weitz-Harms et al., 2023).

Reference scenario

The platform's content model was instantiated with a fully authored reference scenario based on an SSH brute-force intrusion, in which the learner investigates authentication logs and related artefacts to reconstruct the incident. The scenario exercises the full pipeline (environment construction, multi-path discovery, deterministic scoring, adaptive guidance, and report submission), and it served as the vehicle for the verification described next. The authoring of additional scenarios is identified as future work.

Evaluation method

Feasibility was assessed through automated verification spanning five complementary classes of tests. Unit tests examined individual service functions in isolation; integration tests checked the correct interaction of backend components and their effects on database state; end-to-end tests exercised a complete investigation lifecycle from environment deployment through command execution, discovery, scoring, and report submission; security tests probed authentication, access-control (including insecure direct object reference), input validation, resistance to SQL injection, and non-exposure of configuration secrets; and a load test issued concurrent requests to observe throughput and latency behaviour. All verification was carried out in the local development environment used for the project. The absence of large-scale stress testing beyond this environment, and of any

evaluation involving human learners, is acknowledged as a boundary of the present study and is revisited in the discussion. The quantitative outcomes of these tests are reported in the following section.

RESULTS

This section reports the outcomes of the verification described in the methodology. Because the study's claim concerns feasibility, the results are presented as evidence that the design operates correctly and stably within its intended boundaries; they are not offered as measures of learning effectiveness.

Overview of verification

The suite comprised 55 automated tests (28 pre-existing unit tests and 27 introduced for integration, end-to-end, security, and load verification), executed against the running backend and PostgreSQL database in the local development environment (Node.js and Express, React with Vite, and PostgreSQL). All 55 tests passed, a 100% pass rate, and a regression cycle confirmed that the original 28 unit tests continued to pass after the additional suites were added. Disposable synthetic data were used throughout, and post-run inspection recorded zero residual test users and zero orphaned sessions, indicating that the tests left the database in a consistent state. Table 1 summarises the verification by class.

Table 1: Summary of automated verification by test class

Test class	Tests	Passed	Duration	Result
Unit (backend terminal parser)	13	13	158.93 ms	Pass
Unit (frontend terminal paste)	15	15	2.47 s	Pass
Integration	12	12	10.86 s	Pass
End-to-end (full lifecycle)	1	1	6.08 s	Pass
Security	11	11	7.83 s	Pass
Load and latency (300 requests)	3	3	2.49 s	Pass
Total	55	55	n/a	100% pass

Functional verification

Unit testing validated low-level logic in isolation: backend command parsing and output redirection (13 tests) and frontend terminal paste planning and execution (15 tests). Integration testing then confirmed that backend components operate correctly together and persist to PostgreSQL across authentication, session creation, terminal execution, discovery triggering, objective completion, report handling, submission, evaluation, and progression. The assessment gate behaved as designed: an incomplete investigation was correctly rejected with a validation error, while a completed one was frozen and passed to evaluation and progression. This demonstrates that the core services interoperate rather than merely passing in isolation.

End-to-end lifecycle

A single API-driven end-to-end test exercised the complete lifecycle on the SSH brute-force reference scenario, proceeding from registration and authentication through scenario selection, session creation, environment retrieval, investigation commands, discovery completion (five persisted discoveries verified), report saving, submission, evaluation, and progression. The scripted run completed in 6.08 seconds and produced a maximum mechanical score of 100/100, composed of the path score (50/50), command-usage score (30/30), and conclusion score (20/20) with no hint penalties, after which the session was marked complete and the completion badge was

recorded. This confirms that the discovery, deterministic scoring, submission, and progression components compute correctly and interoperate across an entire session. It is a verification of the pipeline under an ideal scripted path, not a measure of learner performance.

Security verification

All 11 security checks passed. Access-control tests confirmed that unauthenticated requests were rejected, invalid credentials failed without leaking database detail, and tampered tokens were refused; session-ownership checks prevented one user from reading or acting on another user's session, addressing the insecure direct object reference risk. Input-handling tests processed missing and malformed fields safely and absorbed an oversized 5,000-character command without crashing the application. Injection tests confirmed that representative SQL payloads produced neither authentication bypass nor data modification, and that SQL-like text entered at the simulated terminal was treated as investigation input rather than executed. Duplicate submission of a completed investigation was blocked, and inspected responses did not expose sensitive configuration such as the token secret, model API key, or database password. Within the prototype's scope, these results indicate that the tested application-level controls behave as intended.

Performance and load

A basic benchmark issued 100 requests at a concurrency level of 10 to each of three non-destructive endpoints, 300 requests in total, all completing successfully. Table 2 reports throughput and latency. Across the three endpoints, throughput averaged approximately 478.6 requests per second and latency approximately 20.3 milliseconds, with 95th-percentile latency at or below roughly 31 milliseconds, indicating stable behaviour at the tested concurrency on a single machine. Consistent with the study's scope, these figures are reported as evidence of local stability rather than of production-scale scalability; distributed and high-concurrency stress testing was not undertaken.

Table 2: Load and latency results (100 requests per endpoint, concurrency level 10)

Metric	/api/health	/api/scenarios	/api/desktop
Successful / total requests	100 / 100	100 / 100	100 / 100
Throughput (req/s)	504.98	486.27	444.62
Mean latency (ms)	18.77	20.07	21.93
Median latency (ms)	16.08	18.18	22.62
95th percentile (ms)	26.81	31.38	26.96
Maximum latency (ms)	85.71	102.02	27.95

Documented behaviours

Testing also recorded several implementation behaviours that did not constitute test failures but are noted for transparency: usernames are restricted to alphanumeric characters; a missing authentication token returns HTTP 401 whereas a forged token returns HTTP 403; and a repeated submission of a completed session is intercepted at the lookup layer, returning HTTP 404 rather than the submission engine's HTTP 400, while remaining effectively blocked. These reflect current implemented behaviour rather than violations of the tested requirements.

Taken together, the results establish that the Kinetic Breach prototype functions correctly and stably across its core workflow, security controls, and basic load within the tested environment. What these results do and do not license is considered in the following section.

DISCUSSION

The results support a bounded but meaningful conclusion. The prototype demonstrates that a realistic simulated investigation, discovery-based assessment, a gamification layer, and non-revealing adaptive guidance can be combined and delivered through a web browser alone, and that the resulting system is functionally correct across its core workflow, sound against the application-level security checks applied, and stable under modest concurrency. In design-science terms, this establishes the feasibility of the artefact against its functional criteria (Hevner et al., 2004). It does not, in itself, establish that the artefact improves learning, which is a separate question of validation reserved for future study.

Implications for accessible cybersecurity education

The problem that motivated this work was access. The feasibility evidence indicates that the design addresses it at both ends of the delivery chain: the learner requires only a web browser, with no virtual machine, container, or laboratory software to install, while the institution hosts a single conventional web application rather than per-learner virtual machines or the server-side orchestration that lighter container ranges continue to require (Chouliaras et al., 2023; Mills et al., 2024; Nakata & Otsuka, 2021). The content-driven architecture, which constructs environments from version-controlled files at request time rather than storing them in the database, further lowers the marginal cost of adding and maintaining material. For resource-constrained institutions, and for beginners who lack command-line fluency or the means to run laboratory software, this lowers the barrier to a first hands-on experience. Framed against the equity concern raised earlier, an infrastructure-light and engagement-oriented design of this kind is consistent with arguments for broadening participation in cybersecurity through more inclusive learning experiences (Zhong et al., 2024).

Design principles emerging from the study

Beyond the artefact itself, three design decisions are transferable to other efforts in this space. First, multi-path discovery: by allowing different commands and techniques to satisfy the same objective, the discovery mechanism assesses investigative reasoning rather than reproduction of a scripted answer, a closer fit to the open-ended character of real investigation and to experiential conceptions of learning (Kolb, 1984) than the single-token capture-the-flag convention. Second, the separation of deterministic scoring from AI review: keeping an auditable, action-derived score independent of AI-generated commentary protects the integrity of assessment. This decision is directly supported by evidence that AI assistance can raise measured performance without commensurate learning gains (Bassner et al., 2025); folding AI output into the grade would risk exactly that inflation, and the principle generalises to other AI-augmented learning tools. Third, non-revealing, safeguarded guidance: ARIA's progressive and sanitised hints, gated by behavioural triggers and cooldowns, operationalise the calibrated support that tutoring research associates with preserved independent problem-solving (Bassner et al., 2024; Maniktala et al., 2020a; Mousavinasab et al., 2021), while hint caching and provider abstraction reduce operating cost and single-vendor dependence. Finally, the gamification layer is integrated with the investigation task itself rather than layered onto quizzes or isolated challenges, aligning engagement mechanics with the activity they are meant to sustain (Kim et al., 2025; Weitz-Harms et al., 2023).

Trade-offs

Two trade-offs are inherent in the design. The first is fidelity against accessibility. A simulated environment secures safety, reproducibility, and zero learner-side setup, but it does so at the cost of the fidelity and adversarial richness that a live cyber range provides. For beginners forming foundational log-analysis and investigation skills this trade-off is defensible; for advanced or team-based adversarial training it would be limiting. The two approaches are therefore complementary rather than competing. The simulated trainer serves as an accessible on-ramp that can precede, not replace, higher-fidelity environments. The second concerns the guidance component: its richest behaviour draws on an external language model, which carries cost and availability considerations at odds with a pure zero-cost ideal. The provider abstraction, the mock implementation, and hint caching mitigate this, and because the deterministic core (environment construction, discovery, scoring, and gamification) operates independently of the model, the platform degrades gracefully when AI guidance is unavailable.

Limitations

The most consequential limitation is the absence of any evaluation involving human learners. No usability, user-acceptance, engagement, or learning-outcome study was conducted, and the paper accordingly makes no claim that the platform is easy to use, engaging in practice, or effective for learning. The engagement and guidance benefits reported in the reviewed literature motivate the design but are not demonstrated for this artefact. A second limitation concerns content breadth: a single fully authored scenario establishes the content model in principle but does not evidence the effort or the pedagogical value of a broader scenario library. A third concerns the scope of verification, which was carried out by the developer in a single local environment without distributed stress testing, browser-level automation, or a continuous-integration pipeline; the security and performance findings are therefore properly read as application-level checks on a local prototype rather than as production assurances. More broadly, verification confirms that the system does what it was designed to do, which is distinct from validating that the design achieves its educational aim (Hevner et al., 2004). These boundaries define a clear evaluation agenda rather than undermining the contribution: the design is shown to be buildable and sound, while whether it teaches effectively remains to be tested.

CONCLUSION

This paper set out to address a persistent equity problem in cybersecurity education: realistic, hands-on investigation practice is pedagogically important yet gated behind infrastructure such as cyber ranges, virtual-machine laboratories, and container testbeds, whose cost, configuration, and maintenance place it beyond the easy reach of many beginners and resource-constrained institutions. In response, it presented the design and technical feasibility of Kinetic Breach, a browser-delivered platform that combines a simulated investigation environment, discovery-based assessment admitting multiple valid evidentiary paths, a gamification layer, and a non-revealing adaptive AI guidance component, requiring only a web browser on the learner's side and a single conventional web application on the institution's side.

The contribution is twofold. First, the paper advances an accessible, infrastructure-light design whose distinguishing decisions are transferable beyond the specific artefact: multi-path discovery over single-solution scoring, the deliberate separation of deterministic assessment from AI-generated review, and safeguarded guidance that scaffolds without disclosing solutions. Second, it establishes the technical feasibility of that design through systematic verification: fifty-five automated tests across unit, integration, end-to-end, security, and load classes passed in full, a complete investigation lifecycle executed correctly through the assessment and progression pipeline, and the system behaved stably under modest concurrency within the tested environment.

These findings are deliberately bounded. Verification demonstrates that the design is buildable and behaves as intended; it does not establish that the platform is usable, engaging, or effective for learning, and the paper makes no such claim. The evaluation gap therefore defines the work that follows. The most immediate priority is evaluation with human learners: usability and user-acceptance testing, and a study of engagement and learning outcomes, ideally compared against conventional or higher-fidelity alternatives. Such evaluation would allow the design's educational value to be tested rather than assumed. Beyond this, expanding the single reference scenario into a broader, progressively structured library would evidence the content model at scale, and strengthening the verification regime through distributed load testing, browser-level automation, and continuous integration would extend the assurances established here toward production readiness. Positioned in this way, Kinetic Breach is offered as an accessible on-ramp to hands-on cybersecurity investigation and as a design foundation on which that educational value can now be evaluated. It is intended to complement, rather than replace, the higher-fidelity environments that remain essential for advanced training.

DATA AND CODE AVAILABILITY

The source code for Kinetic Breach is archived on Zenodo under restricted access (Ahmed Yousef Harpah et al., 2026; DOI: 10.5281/zenodo.22885484). The record's descriptive metadata are publicly available so that the artefact can be cited, but the source files are not openly downloadable: access is granted by the corresponding author on reasonable request and is contingent on a signed non-disclosure agreement. The software is released under a proprietary, all-rights-reserved licence and may not be redistributed.

This study did not involve human participants, and no personal or experimental data were collected. The artefacts underlying the feasibility evaluation reported here (the automated test suites and their results, together with the version-controlled reference-scenario definitions) form part of the same archived repository and are available under the identical restricted-access conditions. During peer review, temporary access for editors and reviewers can be arranged on request.

REFERENCES

1. Ahmed Yousef Harpah, H. A., Mohd Ezani, M. F., Mustafa, N., & Yaacob, M. A. (2026). Kinetic Breach: A Browser-Based Gamified Cybersecurity Investigation Simulator (Source Code) [Computer software]. Zenodo. <https://doi.org/10.5281/zenodo.22885484>
2. Bassner, P., Frankford, E., & Krusche, S. (2024). Iris: An AI-driven virtual tutor for computer science education. In *Proceedings of the 2024 Conference on Innovation and Technology in Computer Science Education (ITICSE 2024)* (Vol. 1, pp. 394–400). Association for Computing Machinery. <https://doi.org/10.1145/3649217.3653543>
3. Bassner, P., Lenk-Ostendorf, B., Beinstingel, R., Wasner, T., & Krusche, S. (2025). Less stress, better scores, same learning: The paradox of AI support in programming education. *Computers and Education: Artificial Intelligence*. Advance online publication. <https://doi.org/10.1016/j.caeai.2025.100537>
4. Beuran, R., Tang, D., Pham, C., Chinen, K., Tan, Y., & Shinoda, Y. (2018). Integrated framework for hands-on cybersecurity training: CyTrONE. *Computers & Security*, 78, 43–59. <https://doi.org/10.1016/j.cose.2018.06.001>
5. Choularas, N., Kantzavelou, I., Maglaras, L., Pantziou, G., & Ferrag, M. A. (2023). A novel autonomous container-based platform for cybersecurity training and research. *PeerJ Computer Science*, 9, e1574. <https://doi.org/10.7717/peerj-cs.1574>
6. Dichev, C., & Dicheva, D. (2017). Gamifying education: What is known, what is believed and what remains uncertain: A critical review. *International Journal of Educational Technology in Higher Education*, 14, Article 9. <https://doi.org/10.1186/s41239-017-0042-5>
7. Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75–105. <https://doi.org/10.2307/25148625>
8. ISC2. (2024). ISC2 cybersecurity workforce study 2024. <https://www.isc2.org/research>
9. Kim, J. B., Zhong, C., & Liu, H. (2025). The impact of gamification on cybersecurity learning: Multi-study analysis. *Communications of the Association for Information Systems*, 56, 57–96. <https://doi.org/10.17705/1CAIS.05603>
10. Kolb, D. A. (1984). *Experiential learning: Experience as the source of learning and development*. Prentice Hall.
11. Maniktala, M., Cody, C., Barnes, T., & Chi, M. (2020a). Avoiding help avoidance: Using interface design changes to promote unsolicited hint usage in an intelligent tutor. *International Journal of Artificial Intelligence in Education*, 30(4), 637–667. <https://doi.org/10.1007/s40593-020-00213-3>
12. Maniktala, M., Cody, C., Isvik, A., Lytle, N., Chi, M., & Barnes, T. (2020b). Extending the Hint Factory for the assistance dilemma: A novel, data-driven HelpNeed predictor for proactive problem-solving help. *Journal of Educational Data Mining*, 12(4), 24–65. <https://doi.org/10.5281/zenodo.4399683>
13. Martucci, L. A., Magnusson, J., Vehkajärvi, T., & Karlsson, J. (2026). The Cyber Range Lite. In L. Drevin, W. S. Leung, & S. von Solms (Eds.), *Information security education: Empowering people through information security education (IFIP Advances in Information and Communication Technology, Vol. 742)*. Springer. https://doi.org/10.1007/978-3-031-94924-1_12
14. Mills, A., White, J., & Legg, P. (2024). GoibhniUWE: A lightweight and modular container-based cyber range. *Journal of Cybersecurity and Privacy*, 4(3), 615–628. <https://doi.org/10.3390/jcp4030029>
15. Mousavinasab, E., Zarifsanaiy, N., Niakan Kalhori, S. R., Rakhshan, M., Keikha, L., & Ghazi Saeedi, M. (2021). Intelligent tutoring systems: A systematic review of characteristics, applications, and evaluation methods. *Interactive Learning Environments*, 29(1), 142–163. <https://doi.org/10.1080/10494820.2018.1558257>
16. Nakata, R., & Otsuka, A. (2021). CyExec*: A high-performance container-based cyber range with scenario randomization. *IEEE Access*, 9, 109095–109114. <https://doi.org/10.1109/ACCESS.2021.3101245>

17. Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 45–77. <https://doi.org/10.2753/MIS0742-1222240302>
18. Ryan, R. M., & Deci, E. L. (2000). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American Psychologist*, 55(1), 68–78. <https://doi.org/10.1037/0003-066X.55.1.68>
19. Seaborn, K., & Fels, D. I. (2015). Gamification in theory and action: A survey. *International Journal of Human-Computer Studies*, 74, 14–31. <https://doi.org/10.1016/j.ijhcs.2014.09.006>
20. Subhash, S., & Cudney, E. A. (2018). Gamified learning in higher education: A systematic review of the literature. *Computers in Human Behavior*, 87, 192–206. <https://doi.org/10.1016/j.chb.2018.05.028>
21. Vykopal, J., Čeleda, P., Seda, P., Švábenský, V., & Tovaříňák, D. (2021). Scalable learning environments for teaching cybersecurity hands-on. In 2021 IEEE Frontiers in Education Conference (FIE) (pp. 1–9). IEEE. <https://doi.org/10.1109/FIE49875.2021.9637180>
22. Weitzl-Harms, S., Spanier, A., Hastings, J., & Rokusek, M. (2023). A systematic mapping study on gamification applications for undergraduate cybersecurity education. *Journal of Cybersecurity Education, Research and Practice*, 2023(1), Article 9. <https://digitalcommons.kennesaw.edu/jcerp/vol2023/iss1/9/>
23. Zhong, C., Kim, J. B., & Liu, H. (2024). The art of inclusive gamification in cybersecurity training. *IEEE Security & Privacy*, 22(5), 40–51. <https://doi.org/10.1109/MSEC.2024.3427666>