ISSN: 2473-4901
v12 n6745

Operation North Guard: A Multi-Variant Dynamic Flag Framework for Cybersecurity Education

Kian Esmaeili
kesma9557@ung.edu

Tamirat Abegaz
Tamirat.Abegaz@ung.edu

Department of Computer Science and Information Systems
Univerity of North Georgia
Dahlonega, GA, USA

Abstract

Capture-the-Flag (CTF) environments have become an increasingly popular instructional tool in cybersecurity education because they provide realistic, hands-on opportunities for students to apply concepts in digital forensics, cryptography, networking, reverse engineering, web security, and incident response. Despite their educational value, the adoption of CTFs within graded academic environments presents several challenges, including flag sharing, academic integrity concerns, uneven student skill levels, difficulty balancing challenge complexity, and limited instructor visibility into student engagement and performance.

This paper presents Operation North Guard, a custom Capture-the-Flag platform developed for the Cybersecurity Capstone course at the University of North Georgia. Built using Next.js 14, React 18, TypeScript, and MDX-based challenge modules, the platform introduces a multi-variant dynamic flag framework that assigns unique valid flag variants to students while maintaining a consistent challenge experience. This approach reduces opportunities for unauthorized answer sharing, supports individualized assessment, and strengthens academic integrity without increasing instructor workload.

The paper describes the platform's architecture, challenge development workflow, dynamic flag validation model, progress-tracking mechanisms, user-interface design, and deployment strategy. A library of eight cybersecurity challenges spanning forensics, cryptography, reverse engineering, cloud security, web security, privilege escalation, and governance, risk, and compliance (GRC) was developed and integrated into the platform. Initial system validation, supplemented by informal peer feedback, suggests that the platform provides a scalable, maintainable, and instructor-aligned framework for cybersecurity education while preserving the hands-on learning benefits of CTF-based instruction.

Keywords: Capture-the-Flag (CTF), Cybersecurity Education, Academic Integrity, Dynamic Flag Validation, Experiential Learning, Learning Analytics

©2026 ISCAP (Information Systems and Computing Academic Professionals)

Operation North Guard: A Multi-Variant Dynamic Flag Framework for Cybersecurity Education

Kian Esmaeili and Tamirat Abegaz

1. INTRODUCTION

Hands-on learning is widely recognized as a critical component of cybersecurity education because it enables students to apply theoretical concepts in realistic technical environments. Capture-the-Flag (CTF) exercises have emerged as an effective instructional approach by providing opportunities to analyze digital artifacts, investigate security incidents, exploit vulnerabilities, and interpret system behavior using industry-standard tools. As cybersecurity programs increasingly adopt experiential learning models, CTF environments have become an important mechanism for developing and assessing practical technical skills across multiple domains, including digital forensics, cryptography, networking, reverse engineering, and web security.

Despite their educational value, integrating CTFs into graded academic environments introduces challenges that are less prevalent in competitive or recreational settings. One of the most significant concerns is academic integrity, particularly when static challenge flags can be easily shared among students or reused across semesters. In addition, cybersecurity cohorts often exhibit substantial differences in prior experience and technical proficiency, making it difficult to balance challenge difficulty for diverse learners. Instructors also have limited visibility into student engagement, including hint usage, submission attempts, and problem-solving behavior, reducing their ability to assess learning outcomes and identify students who may require additional support. Furthermore, many existing open-source CTF platforms prioritize competition-oriented functionality rather than features specifically designed to support academic assessment and instructional objectives.

To address these limitations, this paper presents the UNG CTF Platform: Operation North Guard, a custom Capture-the-Flag framework developed for the Cybersecurity Capstone course at the University of North Georgia. The platform combines modern web technologies—including Next.js, React, TypeScript, MDX-based content management, GitHub Pages deployment, and automated CI/CD workflows—with educational design principles intended to support both instruction and assessment.

The primary contribution of this work is a multi-variant dynamic flag framework that assigns unique valid flag variants to individual students while preserving a consistent challenge experience. This approach reduces opportunities for unauthorized answer sharing, strengthens academic integrity, and enables reusable challenge content across multiple course offerings. In addition, the platform provides modular architecture, structured challenge authoring workflow, progressive hint system, and scalable deployment model designed specifically for higher-education cybersecurity instruction.

2. PROJECT OVERVIEW

Operation North Guard is a modular Capture-the-Flag platform designed to support both cybersecurity instruction and graded assessment. Its primary objective is to provide students with realistic, hands-on challenges while reducing opportunities for flag sharing through individualized valid flag variants.

The platform includes a curated senior-level challenge library, progressive hints, persistent progress tracking, structured MDX-based challenge modules, and a responsive interface. It is implemented using Next.js, React, TypeScript, and GitHub-based deployment workflows, providing a lightweight and reproducible architecture suitable for academic use.

Challenges are aligned with instructor-defined learning outcomes and technical competencies emphasized in the Cybersecurity Capstone course. The platform supports realistic investigative scenarios across multiple cybersecurity domains while enabling challenge content to be reused across course offerings.

©2026 ISCAP (Information Systems and Computing Academic Professionals)

3. RELATED WORK

Capture-the-Flag (CTF) environments are widely used in cybersecurity education because they provide hands-on opportunities for students to apply theoretical concepts in realistic technical scenarios. Prior research has shown that CTF-based learning improves student engagement, motivation, and practical skill development while promoting higher-order problem solving (Vykopal et al., 2020; Chapman et al., 2014; Švábenský et al., 2018). Despite these benefits, existing CTF platforms have largely been designed for competitive rather than graded academic environments, leaving challenges related to academic integrity, learner diversity, and instructional support.

Beyond individual challenges, CTFs share many characteristics with cyber ranges and other experiential learning environments that seek to simulate realistic operational conditions. These platforms allow students to practice technical skills in controlled settings while developing investigative methodologies that closely resemble professional cybersecurity workflows. Existing platforms such as CTFd, FBCTF, RootTheBox, and PicoCTF have demonstrated the effectiveness of challenge-based cybersecurity learning but were primarily designed for competition-oriented rather than graded academic environments (CTFd LLC, n.d.; Meta Open Source FBCTF, n.d.; RootTheBox Development Team, n.d.).

Despite these advantages, the literature also identifies several limitations that continue to hinder the effective adoption of CTFs within formal academic programs. Researchers have highlighted challenges related to academic integrity, uneven student skill levels, limited instructional analytics, and the difficulty of balancing educational objectives with competition-oriented platform designs. These limitations motivate the development of Operation North Guard and provide the foundation for the design decisions discussed throughout the remainder of this paper.

3.1 Academic integrity challenges in CTF-based courses:

One of the most persistent challenges associated with the use of Capture-the-Flag (CTF) environments in academic settings is maintaining assessment integrity. Most widely used CTF platforms, including CTFd, FBCTF, and RootTheBox, rely on static flag validation models in which every participant submits the same correct answer (CTFd LLC, n.d.; Meta Open Source FBCTF, n.d.; RootTheBox Development Team, n.d.). While appropriate for competitive and recreational learning environments, this approach introduces significant concerns in graded coursework because students can share valid solutions, undermining assessment integrity.

The problem becomes more pronounced when identical challenge sets are reused across multiple semesters, allowing previously disclosed flags to be reused without completing the underlying analysis. Common mitigation strategies—including screenshots, command histories, system logs, and walkthrough reports—provide additional evidence of student work but increase administrative overhead and remain susceptible to unauthorized sharing.

These limitations have motivated interest in individualized validation mechanisms that reduce the value of answer sharing while preserving the accessibility of challenge-based learning. Dynamic flag generation addresses this challenge by assigning unique valid solutions to individual students while maintaining an equivalent challenge experience. Although individualized assessment techniques have been explored in research prototypes and specialized educational systems, they remain relatively uncommon in widely adopted academic CTF platforms.

These challenges motivated the multi-variant dynamic flag framework proposed in Operation North Guard. By assigning student-specific flag variants while maintaining a consistent challenge experience, the platform strengthens academic integrity without imposing additional burdens on students or instructors.

3.2 Skill Level Disparities Among Students

Cybersecurity cohorts frequently include students with diverse technical backgrounds, resulting in substantial differences in prior knowledge and practical experience. As noted by Vykopal et al. (2020), these differences can significantly influence student performance and engagement within Capture-the-

Flag environments. When challenge difficulty is not appropriately structured, novice learners may become discouraged while more experienced students may find challenges insufficiently engaging.

Educational CTF platforms such as PicoCTF demonstrate that progressive challenge design, calibrated difficulty levels, and contextual learning support improve accessibility and student success (Chapman et al., 2014). Similarly, challenge metadata, guided hints, and clearly defined learning objectives help support learners without diminishing the investigative nature of challenge-based learning (Švábenský et al., 2018).

Operation North Guard addresses these challenges through structured difficulty classifications, topic tags, learning objectives, progressive hints, and a balanced scoring model. Together, these features support students with varying experience levels while encouraging analytical problem solving and continued engagement.

3.3 Limitations in Learning Analytics

Effective cybersecurity instruction requires more than simply determining whether students successfully solve challenges. Instructors also benefit from understanding submission attempts, hint usage, time spent on individual tasks, and overall problem-solving behavior, as these metrics provide valuable insight into student engagement, challenge effectiveness, and learning outcomes.

Despite the growing popularity of Capture-the-Flag platforms in educational settings, many open-source CTF systems provide only limited assessment data, typically recording challenge completion, submission timestamps, and scoreboard rankings. While appropriate for competitive environments, these metrics provide limited insight into how students learn, where they struggle, or which concepts present the greatest difficulty.

Learning analytics research emphasizes that meaningful educational data can improve instructional decision-making, support personalized learning, and enable earlier intervention for struggling students (Pardo & Siemens, 2014; Siemens & Baker, 2012). In cybersecurity education, richer behavioral data can also help instructors evaluate challenge effectiveness and better understand student learning patterns.

Although the current implementation of Operation North Guard does not include a centralized instructor analytics dashboard, the platform was intentionally designed to support future analytics integration. Persistent state management, structured challenge metadata, local progress tracking, and a modular architecture provide a foundation for collecting and analyzing student interaction data in future versions of the system.

3.4 Exploring Alternative Integrity Mechanisms: OCR and Perceptual Hashing

Before selecting a dynamic flag–based integrity mechanism, several alternative validation approaches were evaluated during platform development. One approach required students to submit screenshots as evidence of challenge completion. These screenshots would then be processed using an Optical Character Recognition (OCR) engine, such as Tesseract, to automatically extract and validate challenge results. Prior research has shown that OCR accuracy depends on factors including image quality, text clarity, background complexity, and device-specific characteristics (Smith, 2007; Elgibreen & Almazroi, 2019). Variations in operating systems, display settings, screenshot formats, and image compression can further reduce recognition accuracy, introducing false positives, false negatives, and inconsistent user experiences.

Perceptual hashing techniques, such as those described by Zauner (2010), were also evaluated for validating screenshots while tolerating minor visual differences. Unlike traditional cryptographic hashing, perceptual hashing identifies visually similar images despite small modifications. However, image transformations such as cropping, scaling, compression, overlays, and brightness adjustments can alter perceptual hash values, limiting their suitability for high-confidence academic assessment.

Although both approaches offered partial solutions, they introduced limitations in reliability, usability, and validation accuracy. Consequently, they were rejected in favor of the proposed dynamic multi-

variant flag framework, which provides individualized validation without relying on screenshots or image-processing techniques.

3.5 Dynamic Multi-Variant Flags as a Novel Academic Integrity Solution

The limitations identified in existing CTF platforms and alternative validation mechanisms motivated the development of a multi-variant dynamic flag framework for academic environments. Unlike traditional CTF systems that rely on a single static solution for each challenge, the proposed approach assigns students individualized valid flag variants while preserving an equivalent challenge experience. This design reduces the value of answer sharing and helps ensure that successful submissions reflect genuine engagement with challenge content.

Within Operation North Guard, each challenge contains multiple valid flag variants that are deterministically assigned to individual students. By linking flag variants to student-specific identifiers and challenge metadata, the platform automatically validates submissions while maintaining consistency, scalability, and ease of administration. The approach enables challenges to be reused across semesters without substantial redesign and reduces the administrative overhead associated with manual verification techniques such as screenshots, walkthrough reports, or command-history reviews.

In addition to strengthening academic integrity, the framework supports individualized assessment while preserving the accessibility and usability that make CTF-based learning effective. Students continue to engage with the same technical objectives, artifacts, and investigative workflows, ensuring fairness while reducing opportunities for unauthorized collaboration. Although dynamic validation mechanisms remain relatively uncommon within widely adopted educational CTF platforms, they offer a promising approach for addressing assessment authenticity and challenge reuse. The multi-variant flag framework proposed in Operation North Guard serves as the primary contribution of this work.

Overall, the literature demonstrates strong support for the use of CTFs in cybersecurity education while identifying challenges related to academic integrity, student skill disparities, instructional analytics, and assessment authenticity. Operation North Guard addresses these gaps through scalable platform architecture and an individualized dynamic flag validation model tailored to higher-education cybersecurity programs.

4. SYSTEM DESIGN AND METHODOLOGY

The UNG CTF Platform: Operation North Guard was designed as a modern, modular, and scalable cybersecurity education platform that supports both hands-on learning and academic assessment. The system combines contemporary web technologies—including Next.js 14, React 18, TypeScript, MDX-based content management, and GitHub-driven deployment workflows—with instructional design principles intended to promote accessibility, maintainability, and academic integrity.

A primary design objective was the development of an individualized challenge-validation framework capable of reducing opportunities for answer sharing while preserving the usability and engagement benefits commonly associated with Capture-the-Flag environments. To support this goal, the platform incorporates a multi-variant dynamic flag model, structured challenge metadata, progressive hint systems, persistent state management, and a modular architecture that enables future expansion through analytics, authentication services, and backend integrations.

This section presents the overall system architecture, challenge development pipeline, state management strategy, dynamic flag validation framework, user-interface design considerations, and software development workflow used throughout the implementation of Operation North Guard.

4.1 Architectural Overview

Operation North Guard was developed using a modern component-based web architecture centered on Next.js 14, React 18, and TypeScript. The platform adopts a static-site deployment model that enables efficient hosting through GitHub Pages while minimizing infrastructure requirements and operational complexity. This design supports scalability, maintainability, and ease of deployment while remaining accessible to students and instructors.

The architecture is organized into several logical layers. The presentation layer is implemented using the Next.js App Router, React components, Tailwind CSS, and Framer Motion, providing a responsive and interactive user experience across desktop and mobile devices. The content layer is powered by MDX, allowing challenge descriptions, metadata, downloadable artifacts, and instructional content to be stored as version-controlled files within the repository. Supporting libraries such as next-mdx-remote and gray-matter enable dynamic content rendering and metadata extraction during the build process.

Application state is managed through a persistent Zustand store that maintains challenge progress, hint usage, solved challenges, and user preferences across browsing sessions. This approach provides lightweight state management without introducing the complexity of a centralized backend service. The security layer incorporates hashed flag validation and challenge-specific verification mechanisms designed to prevent direct exposure of valid solutions within client-accessible code.

Deployment is automated through a GitHub Actions continuous integration and continuous deployment (CI/CD) pipeline. During each deployment cycle, the application is statically exported and published to GitHub Pages, enabling a reproducible deployment process while maintaining compatibility with a fully serverless architecture. Figure 1 illustrates the high-level architecture of Operation North Guard and the relationships among its major system components.

4.2 File and Directory

The project follows a modular directory structure designed to support maintainability, scalability, and separation of concerns. Figure 1 presents a simplified view of the repository organization. The application layer is implemented within the Next.js App Router framework, while reusable interface elements are contained within a dedicated components directory. Challenge content is stored separately as MDX modules, allowing educational material to be developed independently from application logic. Core platform functionality, including content loading, state management, and utility functions, is encapsulated within the lib directory.

This organization promotes a clear separation between presentation, content, and application state, reducing coupling among system components and simplifying future maintenance. The modular structure also supports extensibility by enabling new challenge categories, interface components, and platform features to be added with minimal impact on existing functionality

```
ctf-platform/
|-- app/                                # Next.js App
    Router (root)
|   |-- layout.tsx                      # Global layout
    with theme + nav
|   |-- page.tsx                        # Homepage
|   |-- globals.css                     # Tailwind CSS base
|   |-- challenges/                     # Challenge listing
|   |   |-- page.tsx
|   |   `-- [slug]/                     # Dynamic challenge
     pages
|   |       `-- page.tsx
|   |-- scoreboard/                     # Scoreboard and
    metrics
|   `-- about/                          # Documentation
    page
|
|-- components/                         # UI components
|   |-- ui/                             # Buttons, cards,
    dialogs
|   |-- challenge-card.tsx
|   |-- progress-tracker.tsx
|   `-- toast.tsx
|
|-- lib/                                # Core logic
|   |-- content.ts                      # MDX loader
|   |-- store.ts                        # Zustand
    persistent store
|   `-- utils.ts                        # Helper utilities
|
|-- content/                            # MDX challenge
    files
|   `-- challenges/
|
|-- public/                             # Static assets
`-- scripts/                            # Build + hashing
    scripts
```

Figure 1. High-Level Platform Directory Structure

ISSN: 2473-4901
v12 n6745

4.3 Technical Architecture Details

Operation North Guard is built using the Next.js 14 App Router architecture, providing a modular, component-based framework for routing, content delivery, and user-interface management. This architecture supports maintainability, performance, and scalability.

The platform uses static-site generation through Next.js, allowing deployment on GitHub Pages without a dedicated application server. This approach reduces infrastructure requirements while providing reliable, low-cost hosting for educational environments.

Challenge content is delivered through an MDX-based content engine that combines Markdown authoring with React components. This design allows authors to create interactive learning experiences while maintaining a structured and reusable content format. Each challenge begins with a YAML front-matter section defining metadata such as the challenge identifier, title, category, difficulty, point value, tags, and student-specific flag validation metadata (Figure 2).

The MDX body contains the challenge scenario, mission briefing, learning objectives, downloadable artifacts, and progressive hints. During the build process, gray-matter parses the front matter while the MDX content is transformed into React components for rendering. This separation of challenge content from application logic allows instructors and future developers to create or modify challenges without changing the underlying source code.

```yaml
---
id: network-breach
title: Network Breach Analysis
difficulty: medium
category: forensics
points: 150
tags:
  - wireshark
  - dns
  - analysis
flag:
  format: "ung{.*}"
  salt: "ung_salt_network-breach_2025"
  variants:
    - id: variant-a
      sha256: "9d13fa6b97e3d5e2e0b58e4df2f55209dc9ecf137dd261950c5f12bed4d94190"
    - id: variant-b
      sha256: "b92979e3a495ccedccf1645c6bdac9e2f45fb6e4063e15ca4488936ec024f79e"
```

Figure 2. Example MDX Challenge Metadata Structure.

4.4 State Management Using Zustand

Because the current implementation of Operation North Guard does not utilize a backend database, a client-side state management solution was required to maintain user progress and platform state. Zustand was selected due to its lightweight architecture, strong TypeScript integration, minimal boilerplate requirements, and built-in persistence capabilities. These characteristics make it particularly well suited for static web applications that require reliable local data management without introducing the complexity of a centralized backend.

The Zustand store maintains essential platform information, including completed challenge solves, revealed hints, student metadata, accumulated scores, and session timing metrics. By centralizing application state within a single store, the platform ensures consistent data access across components while simplifying future expansion efforts. A simplified representation of the store interface is shown in Figure 3.

©2026 ISCAP (Information Systems and Computing Academic Professionals)

ISSN: 2473-4901
v12 n6745

```
interface CTFStore {
  solves: Solve[];
  hintsRevealed: Record<string, number[]>;
  startTime: number;
  studentName: string;
  studentId: string;

  addSolve: (solve: Solve) => void;
  revealHint: (challengeId: string, hintIndex:
      number) => void;
  setStudentInfo: (name: string, id: string) => void
      ;
  getTotalScore: () => number;
}
```

Figure 3. Simplified Zustand State Management Interface.

To preserve student progress between sessions, the store is configured to persist data within the browser's local storage. As a result, challenge completions, revealed hints, and scoring information remain available after page refreshes or browser restarts on the same device. This persistence mechanism improves usability while maintaining compatibility with the platform's static-site architecture. Furthermore, the centralized state model provides a foundation for future enhancements, including cloud-synchronized progress tracking, instructor analytics dashboards, and multi-user account support.

4.5 Dynamic Multi-Variant Flag Validation

The primary security mechanism within Operation North Guard is its multi-variant dynamic flag validation framework. Traditional Capture-the-Flag platforms typically rely on static flags, meaning that every participant submits the same correct solution. While this approach is suitable for competitive environments, it introduces academic integrity concerns when challenges are used for graded assessment. Once a valid flag is shared, other students may obtain credit without completing the underlying analysis.

To address this limitation, Operation North Guard implements a validation model based on salted SHA-256 hashing and multiple valid flag variants. Each challenge may contain several acceptable flag values, with every variant associated with a unique cryptographic hash. During flag submission, the user-provided input is combined with a challenge-specific salt and hashed before comparison against the stored value. A simplified representation of the validation process is shown in Figure 4.

```
const validateFlag = (input: string, challenge: Challenge, studentId: string) => {
    // 1. Deterministically assign a variant based on student and challenge IDs
    const variantId = assignVariant(studentId, challenge.id);

    // 2. Find the predefined variant for this challenge
    const variant = challenge.flag.variants.find(v => v.id === variantId);

    if (!variant) {
        throw new Error("Variant not found for this challenge.");
    }

    // 3. Hash the submitted input using the challenge salt
    const salt = `ung_salt_${challenge.id}_2025`;
    const inputHash = crypto.createHash("sha256")
        .update(salt + input)
        .digest("hex");

    // 4. Compare against the assigned variant's stored hash
    return inputHash === variant.sha256;
};
```

Figure 4. Simplified Dynamic Flag Validation Process.

Because only hashed values are stored, valid flags cannot be directly recovered from challenge data. Furthermore, the use of multiple valid flag variants reduces the effectiveness of answer sharing, as a

ISSN: 2473-4901
v12 n6745

flag assigned to one student may not be valid for another. This approach also allows challenge artifacts to be reused across multiple semesters without compromising assessment integrity.

To reinforce consistency across the platform, all challenge solutions follow a standardized flag format (e.g., ung{...}). The flag submission interface, shown in Figure 5, provides students with a consistent validation experience while clearly communicating challenge point values and submission requirements.

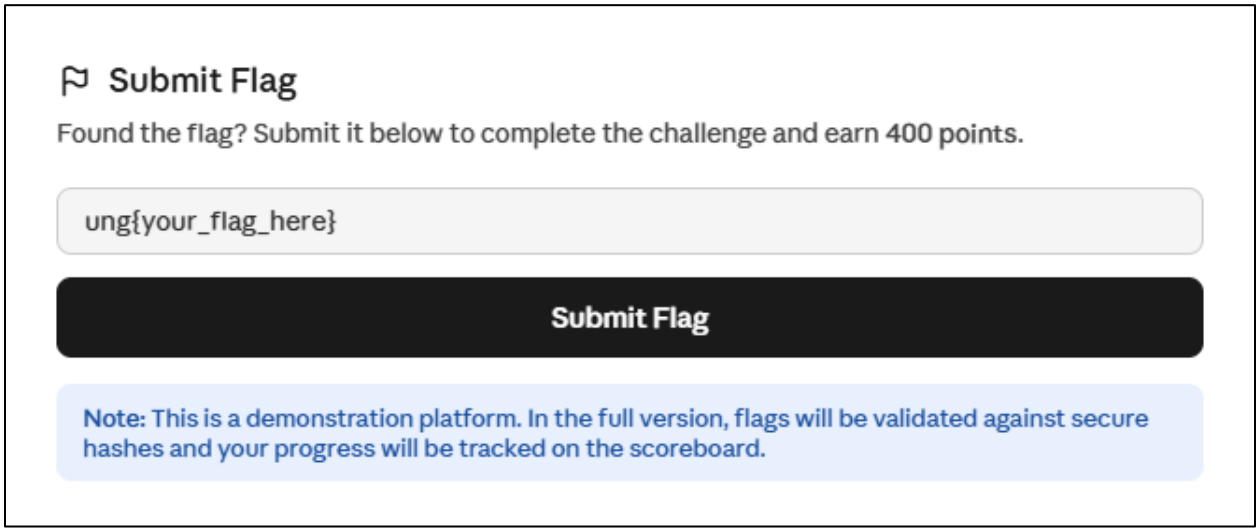

Figure 5. Flag Submission Interface with Standardized Flag Format.

4.6 User Interface and Experience Design

Operation North Guard was designed with an emphasis on clarity, consistency, accessibility, and ease of use. The challenge catalog uses a card-based layout that displays difficulty, category, and point values, allowing students to quickly identify relevant challenges and monitor their progress.

Each challenge is presented through a dedicated page containing the scenario description, learning objectives, downloadable artifacts, progressive hints, and flag submission interface. The platform also includes progress tracking that displays accumulated points, completed challenges, and overall completion percentage.

A responsive interface ensures compatibility across desktop, tablet, and mobile devices while maintaining consistent user experience. Tailwind CSS provides a maintainable styling framework, and Framer Motion enhances interactive elements with lightweight animations.

4.7 Scoring, Hints, and Difficulty Calibration

Operation North Guard incorporates a scoring and hint framework designed to balance educational accessibility with technical rigor. Challenge point values are assigned according to expected effort, complexity, and required technical proficiency. Easy challenges are typically worth 100–200 points and focus on foundational concepts, medium challenges range from 200–350 points and require deeper analytical reasoning or the integration of multiple techniques, and hard challenges range from 350–500 points and involve advanced investigative workflows, multi-stage problem solving, or specialized cybersecurity knowledge.

The platform homepage, shown in Figure 6, provides an overview of available challenges, accumulated points, challenge categories, and overall progression. This centralized interface allows students to quickly identify challenge areas, monitor performance, and navigate the challenge library while maintaining a professional appearance aligned with the University of North Georgia branding.

©2026 ISCAP (Information Systems and Computing Academic Professionals)

ISSN: 2473-4901
v12 n6745

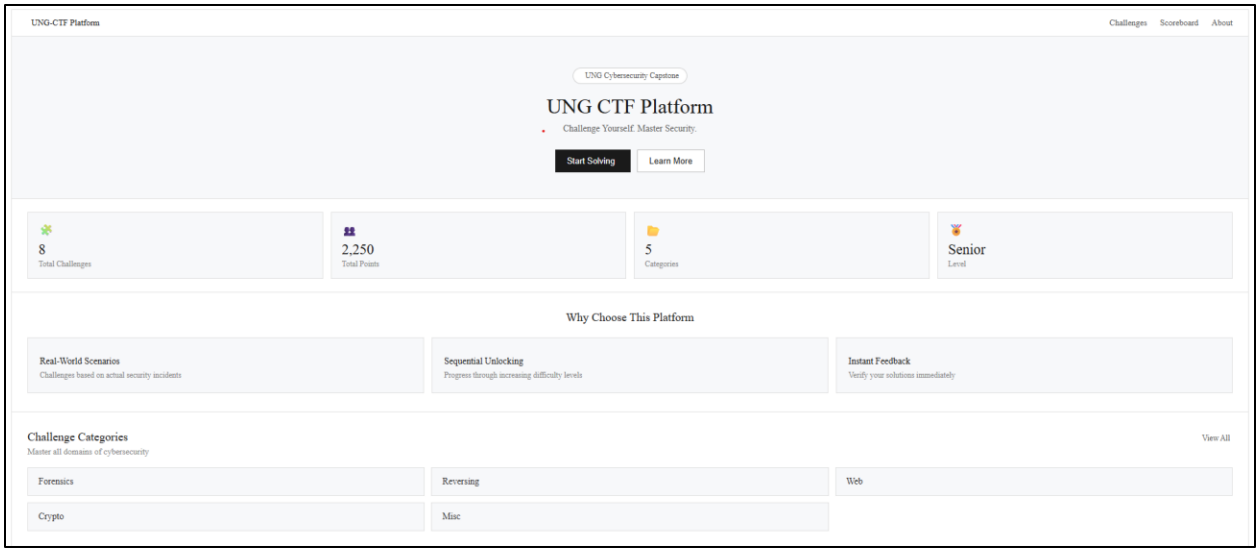

Figure 6. Platform Homepage Displaying Overview Metrics, Feature Highlights, And Challenge Category Navigation., feature highlights, and challenge category navigation.

To support students with varying levels of experience, each challenge includes a progressive hint system. Rather than revealing solutions directly, hints are structured to provide increasingly specific guidance as students work through a challenge. Early hints typically direct students toward relevant tools, artifacts, or investigative approaches, while later hints provide more targeted assistance. This approach helps reduce frustration for novice learners while preserving the analytical nature of the challenge-solving process.

Figure 7 illustrates the hint interface implemented within the platform. Hints are presented through collapsible panels that can be revealed incrementally as needed. This design encourages independent problem solving while still providing structured support when students encounter difficulties.

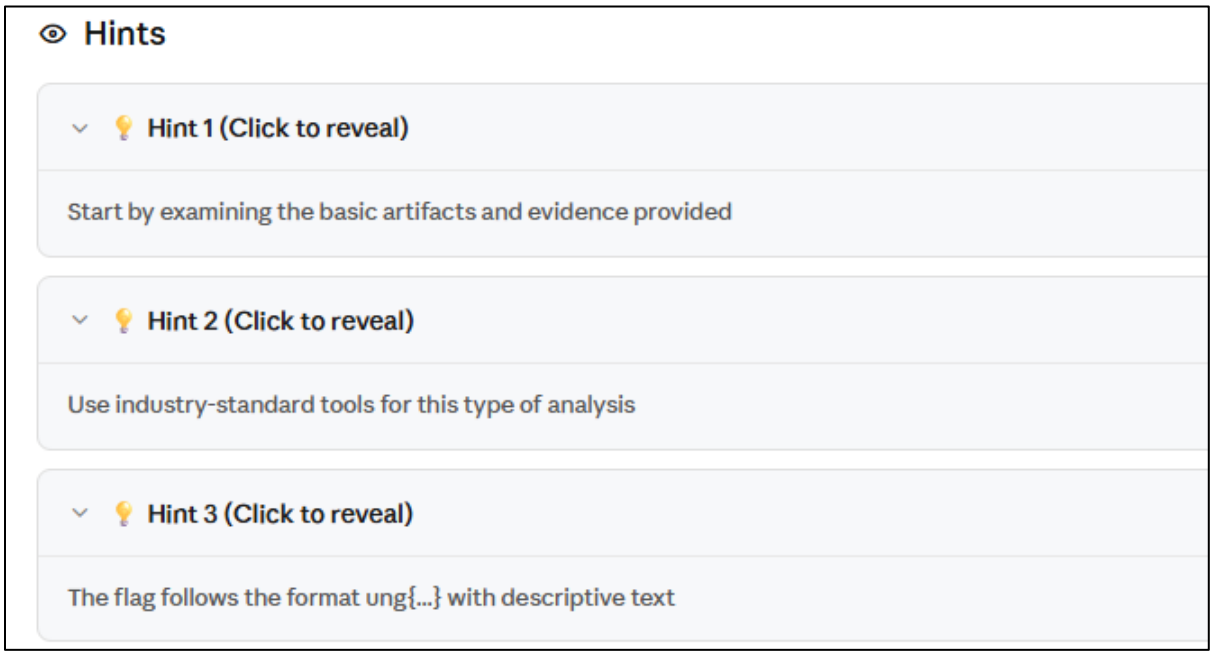

Figure 7. Progressive hint system with collapsible hint panels that provide increasingly specific guidance.

ISSN: 2473-4901
v12 n6745

Together, the scoring and hint mechanisms support a balanced learning experience by encouraging student progression, accommodating diverse skill levels, and promoting engagement without trivializing challenge content.

4.8 Development Workflow

Operation North Guard was developed using an iterative software engineering process emphasizing modularity, maintainability, and incremental feature integration. Development progressed from the core application framework to challenge delivery, dynamic flag validation, state management, and user-interface refinement. The platform was deployed through an automated GitHub Actions workflow using static-site generation and GitHub Pages, providing a lightweight, reproducible architecture that can be readily extended with future capabilities such as instructor analytics, centralized authentication, and backend services. The complete implementation is publicly available through the project repository (Esmaeili, 2026).

5. CHALLENGE DESIGN AND IMPLEMENTATION

A core component of Operation North Guard is its curated library of eight senior-level cybersecurity challenges. These challenges were engineered to reflect realistic investigative workflows and align with competencies expected of students enrolled in the Cybersecurity Capstone course. Each challenge is implemented as an MDX module containing structured metadata, downloadable artifacts, progressive hints, learning objectives, and support for dynamic flag validation.

The challenge library spans multiple cybersecurity domains, including network forensics, disk forensics, cryptography, reverse engineering, web security, cloud security, privilege escalation, and governance, risk, and compliance (GRC). Figure 8 illustrates the challenge catalog interface used by students to browse available challenges, while Table 1 summarizes the implemented challenge set.

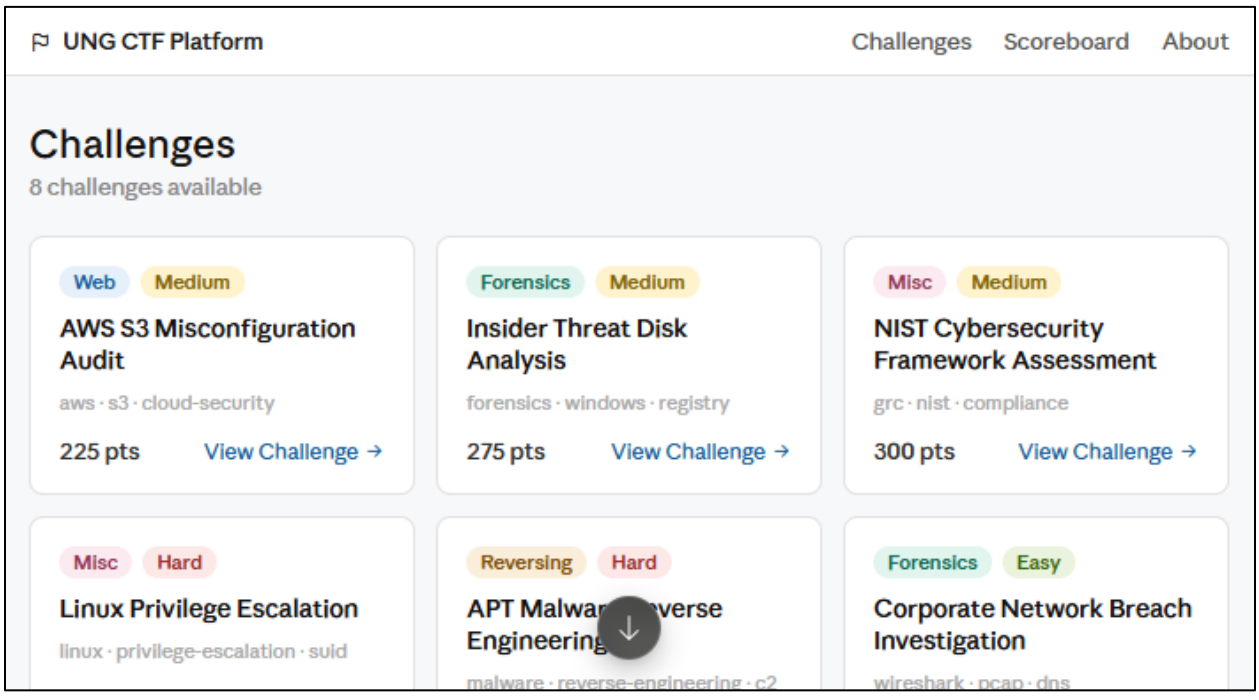

Figure 8. Challenge Catalog Displaying All Eight Cybersecurity Challenges Organized By Category, Difficulty, And Point Value.

Each challenge follows a consistent instructional structure consisting of a realistic scenario, mission briefing, supporting artifacts, defined learning outcomes, recommended tools, progressive hints, and

dynamic flag variants. Artifacts include packet captures, disk images, registry hives, binary executables, cloud resources, and compliance documentation, depending on the challenge domain. This standardized design framework promotes consistency across the platform while allowing challenges to target distinct technical competencies.

The challenge set provides a total of 2,250 available points distributed across multiple difficulty levels. Point values were assigned according to expected effort, technical complexity, and required analytical depth. Collectively, the challenge library exposes students to a broad range of cybersecurity concepts while emphasizing practical investigation, problem solving, and tool proficiency.

Several challenges also incorporate open-ended investigative elements inspired by research-oriented cybersecurity competitions such as CSAW. Rather than following a predetermined sequence of steps, students are required to analyze unfamiliar artifacts, formulate hypotheses, evaluate multiple possibilities, and justify their conclusions before arriving at a valid solution. This hybrid approach bridges traditional Capture-the-Flag exercises with the analytical reasoning and independent inquiry expected of senior-level cybersecurity students.

Challenge	Domain	Difficulty	Points
Network Breach Analysis	Forensics	Easy	150
Weak RSA Cryptosystem	Cryptography	Medium	250
Insider Disk Investigation	Forensics	Medium	275
Healthcare Portal SQLi	Web Security	Medium	200
APT Malware Analysis	Reverse Engineering	Hard	400
AWS S3 Misconfiguration	Cloud Security	Medium	225
Linux Privilege Escalation	Privilege Escalation	Hard	450
NIST CSF Assessment	GRC / Compliance	Medium	300

Table 1. Summary of implemented challenges.

6. RESULTS, EVALUATION, AND DISCUSSION

Operation North Guard successfully progressed from an architectural concept into a functional cybersecurity education platform. The system was evaluated through local testing, static-export validation, and deployment on GitHub Pages. Testing confirmed reliable operation of challenge content, navigation, state persistence, progress tracking, and dynamic flag validation across supported browsers and devices.

The completed platform includes eight cybersecurity challenges spanning network forensics, disk forensics, cryptography, reverse engineering, web security, cloud security, privilege escalation, and governance, risk, and compliance. Each challenge includes structured metadata, downloadable artifacts, learning objectives, progressive hints, and dynamic flag validation, providing 2,250 available points across multiple difficulty levels and cybersecurity domains.

The dynamic multi-variant flag framework functioned as intended throughout development and testing. Salted SHA-256 hashing protected stored flag values while enabling reliable validation of legitimate submissions. Deterministic assignment of predefined flag variants supports challenge reuse and reduces opportunities for answer sharing in academic environments.

Informal peer testing with cybersecurity students produced positive feedback regarding usability, navigation, challenge organization, and overall platform design. Participants reported that the interface was intuitive and that the challenge workflow provided a clear progression from scenario analysis through investigation and flag submission. Feedback also resulted in refinements to spacing, typography, mobile responsiveness, and dark-mode contrast.

The project demonstrates that Capture-the-Flag activities can be adapted for graded academic environments when supported by appropriate integrity mechanisms. It also illustrates that MDX-based challenge authoring and static-site deployment provide a practical foundation for developing and maintaining educational cybersecurity platforms.

Several limitations remain. The current implementation stores progress locally rather than through a centralized database, preventing instructors from monitoring student activity remotely. Instructor-facing analytics, including submission histories, hint-utilization patterns, and engagement metrics, have not yet been implemented. Dynamic flag variants currently rely on predefined challenge configurations, and future versions may benefit from automated flag-generation mechanisms. Additionally, the platform presently focuses on static challenge content and does not include containerized laboratory environments or live cyber-range functionality.

Future work will focus on instructor analytics, centralized authentication and user management, cloud-synchronized progress tracking, automated flag generation, and support for dynamic challenge environments.

7. CONCLUSION AND FUTURE WORK

Operation North Guard demonstrates that CTF methodologies can be successfully adapted for formal cybersecurity education while supporting academic integrity, usability, and instructional value. Developed for the Cybersecurity Capstone course at the University of North Georgia, the platform combines modern web technologies, structured challenge design, and a multi-variant dynamic flag framework to create a scalable educational environment.

The completed system provides eight cybersecurity challenges spanning multiple technical domains, including forensics, cryptography, reverse engineering, web security, cloud security, privilege escalation, and governance, risk, and compliance. Through progressive hints, standardized challenge structures, and individualized flag validation, the platform supports student learning while improving assessment authenticity.

Operation North Guard provides a foundation that can be extended by future cybersecurity programs seeking to incorporate experiential learning into their curricula. The proposed multi-variant dynamic flag framework also demonstrates a practical approach to reducing answer sharing while preserving fairness and challenge reusability in educational CTF platforms.

Future work will focus on instructor analytics, centralized authentication and user management, cloud-synchronized progress tracking, automated flag generation, dynamic containerized laboratory environments, and additional challenge content covering emerging cybersecurity topics.

8. REFERENCES

Chapman, G., Burket, M., & Brumley, D. (2014). PicoCTF: A game-based computer security competition for high school students. In *Proceedings of the USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE)*. USENIX.

CTFd LLC. (n.d.). *CTFd*. https://ctfd.io

Elgibreen, H., & Almazroi, A. (2019). Improving OCR text recognition using preprocessing techniques. *International Journal of Advanced Computer Science and Applications, 10*(5), 126–133.

Esmaeili, K. (2026). *Operation North Guard: A multi-variant dynamic flag framework for cybersecurity education* [GitHub repository]. https://github.com/Kian11228/ung-final-ctf

Meta Open Source. (n.d.). *Facebook Capture the Flag (FBCTF)*. GitHub. https://github.com/facebookarchive/fbctf

Pardo, A., & Siemens, G. (2014). Ethical and privacy principles for learning analytics. *British Journal of Educational Technology, 45*(3), 438–450. https://doi.org/10.1111/bjet.12152

RootTheBox Development Team. (n.d.). *RootTheBox*. GitHub. https://github.com/moloch--/RootTheBox

Siemens, G., & Baker, R. S. J. d. (2012). Learning analytics and educational data mining: Towards communication and collaboration. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge* (pp. 252–254). ACM. https://doi.org/10.1145/2330601.2330661

Smith, R. (2007). An overview of the Tesseract OCR engine. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR)* (pp. 629–633). IEEE. https://doi.org/10.1109/ICDAR.2007.4376991

Švábenský, D., Vykopal, J., & Ošlejšek, R. (2018). Cybersecurity knowledge and skills taught in Capture the Flag challenges. In *Proceedings of the 2018 USENIX Workshop on Advances in Security Education (ASE)*. USENIX.

Vykopal, J., Švábenský, V., & Chang, E.-C. (2020). Benefits and Pitfalls of Using Capture the Flag Games in University Courses. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)* (pp. 752–758). ACM. https://doi.org/10.1145/3328778.3366893

Zauner, C. (2010). *Implementation and benchmarking of perceptual image hash functions* (Master's thesis, University of Applied Sciences Upper Austria).