Checkpoint: A Tool for Supporting Terminal-Based Capture-the-Flag Assessments

Connor Robert Bernard*
University of California, Berkeley
Berkeley, CA, USA
connorbernard@berkeley.edu

Melissa Fabros*
University of California, Berkeley
Berkeley, CA, USA
mfabros@berkeley.edu

Zhifei Li*
Renmin University of China
Beijing, CN
zhifei.li@berkeley.edu

Narges Norouzi
University of California, Berkeley
Berkeley, CA, USA
norouzi@berkeley.edu

Dan Garcia
University of California, Berkeley
Berkeley, CA, USA
ddgarcia@berkeley.edu

Armando Fox
University of California, Berkeley
Berkeley, CA, USA
fox@berkeley.edu

Abstract

Checkpoint is a responsive, terminal-based assignment system designed to support faculty in teaching and assessing a broad scope of computer science topics with Capture-the-Flag (CTF) style challenges. Unlike existing CTF frameworks, where binary completion of a given task is the only metric for evaluation, *Checkpoint* enables students to receive partial credit and dynamic feedback as they complete tasks. Not only does this allow students to confirm they are on the right track, but it simultaneously provides detailed analytics to instructors about each student's progress as they complete the assignment.

Presented by an in-browser terminal emulator neatly wrapped in a Docker image, *Checkpoint* is a versatile tool that can be implemented in a variety of Learning Management Systems (LMSs) with minimal effort. Instructors can easily define flags in a user-friendly configuration file with flag names mapped to optional feedback and regex patterns, which automatically mark flags as complete and render feedback to students within their terminal environment.

ACM Reference Format:
Bernard, et al.. 2025. *Checkpoint*: A Tool for Supporting Terminal-Based Capture-the-Flag Assessments. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 2 (SIGCSE TS 2025), February 26-March 1, 2025, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3641555.3705142

1 Introduction and Background

Capture-the-Flag challenges (CTFs) are a style of cybersecurity assessments that *gamify* learning by embedding a "password" somewhere in a vulnerable system. The student hacks the system and submits the password to a separate server that checks if the password matches an expected value. This assessment style has been shown to significantly increase student motivation without reducing proficiency as evaluated through summative assessments [3].

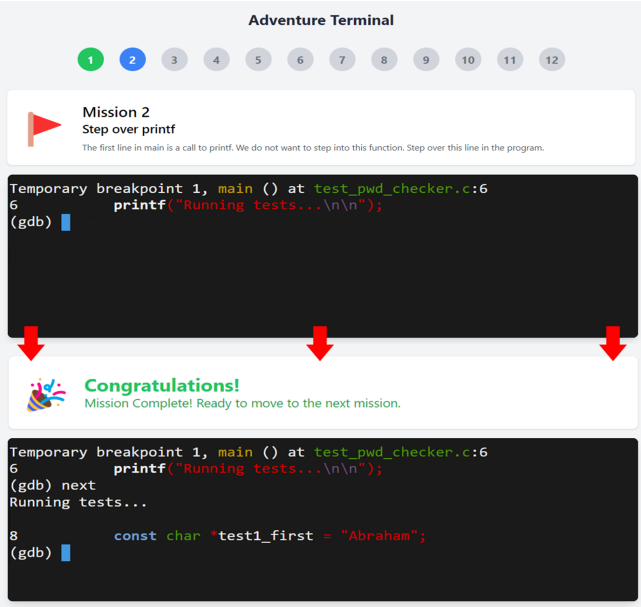

Figure 1: *Checkpoint* GDB capture-the-flag session both before and after flag completion.

Beyond cybersecurity, the pedagogical applications of CTFs are largely unprecedented. One platform, *uAssign*, provides a preliminary implementation and evaluation of such an expansion. It emulates command-line tooling in assessments following a typical CTF format. Researchers found this platform capable of increasing student-reported proficiency with related concepts by over 20% [1].

Despite the many benefits of CTFs existing formulations that are too challenging tend to deter students from engaging with the content rather than increase motivation and learning. One study solved this issue by starting with "level zero" questions and progressively introducing more challenging CTFs [6]. *Checkpoint* takes a different approach. In addition to mitigating this problem, it has the capability to provide instantaneous, responsive feedback as the students complete sub-parts of a given question.

When properly implemented, many studies have found that CTF-style assessments consistently increase student engagement,

Developed alongside Hasanov, et al. [4].
*equal contribution.

SIGCSE TS 2025, February 26-March 1, 2025, Pittsburgh, PA, USA

ACM ISBN 979-8-4007-0532-8/25/02
https://doi.org/10.1145/3641555.3705142

reinforce student knowledge, and increase student confidence when applying evaluated techniques to real-life applications [5]. These exercises leverage constructivist learning theory, which posits that learners construct new knowledge by actively engaging with content that builds on their previous experiences [2].

2 Methodology

Checkpoint introduces several novel features to support its interactive CTFs, including *dynamic flags* that grant partial credit and provide students incremental feedback. In addition to contributing to student experience and success, it provides instructors a window into individual and aggregate student performance on CTFs.

We aim to answer the following Research Questions (RQs):

RQ1 How does providing incremental feedback to students as they solve CTF-style questions affect their ability to demonstrate mastery of the core concepts being evaluated?

RQ2 To what extent does smaller, flag-style prompting affect the ability of students to efficiently apply evaluated topics?

RQ3 How do fine-grained question analytics for CTFs aid instructors in creating CTFs that accurately reflect student mastery?

The following subsections outline the main contributions of the *Checkpoint* system.

2.1 Partial Credit

Rather than assessing a question as "solved" or "not solved," *Checkpoint* introduces the ability to have multiple flags that can be automatically triggered while the student interacts with the terminal session. These intermediary flags enable instructors to award partial credit for completing a subset of flags rather than solely evaluating based on the completion state. When point weights are assigned to flags, *Checkpoint* serves as an all-in-one question generator and auto-grader, making integration with various Learning Management Systems (LMSs) seamless.

2.2 Dynamic Flags

Checkpoint diversifies the nature of a "flag" from the typical approach that uses unique strings to identify completion. Instead, it uses regex to automatically parse and match text in a specified file or within the terminal, awarding completion when a flag's value is found. Moreover, since flag identification and validation are done automatically, students no longer need to take extra steps to receive credit, such as copying the flag's value to a separate tool. This enables instantaneous and responsive feedback as students complete the assigned task.

2.3 Incremental Feedback

Another contribution unique to *Checkpoint* is its ability to leverage its multi-flag architecture and automated evaluation technique to provide incremental feedback as students complete assessments. Feedback can be scoped to assessment style depending on whether the assessment is formative or summative.

2.4 Question Analytics

By nature, traditional CTFs typically do not provide detailed analytics on the state of the question. This means that, even if the total amount of time spent on a flag is logged, given the single-objective nature of the assessment, such data typically does not provide meaningful insight into the students' approach or struggles. Conversely, *Checkpoint* offers detailed analytics data for each flag, logging not only the amount of time spent on the flag but also the different commands used when solving a problem.

3 Applications and Example Use Case

Since *Checkpoint* leverages a learning environment that simulates a live terminal, it affords instructors the ability to objectively evaluate the practical application of course concepts. Moreover, by including multiple flags in their assessments, instructors may choose to utilize question analytics to influence point weight for specific flags, or iterate on question design to be used in future offerings.

One example use case for *Checkpoint* is to evaluate a student's competency with a command line interface tool introduced in the class. If an instructor wishes to test students' ability to use GDB (a debugger for the C programming language) to identify a bug in some provided sample code, the most representative evaluation would require students to actively demonstrate the use of GDB.

Figure 1 illustrates an assessment in progress through *Checkpoint*'s interactive interface. To configure the flag shown in the initial state of this figure, the instructor may set the flag using regex as follows: "`\d\s+const char \*test1_first = ".*";`." This regex matches the expected next line of code regardless of the value of `test1_first`.

4 Future Work

Checkpoint is currently a proof-of-concept implementation. We plan to develop example questions as drop-in solutions for instructors and as a reference when creating personalized questions. We also hope to measure application load to predict system requirements for concurrent evaluations to ensure reliability at scale.

Additionally, we aim to build native integrations of *Checkpoint* with common LMSs, such as Canvas. The end goal of this project is to perform user studies after deploying *Checkpoint* in courses both as a formative and summative assessment. These user studies will inform us on how *Checkpoint* **(RQ1)** enables students to better demonstrate conceptual mastery by completing the entire assessment, and **(RQ2)** increases student fluency with course concepts as they solve tasks more efficiently.

References

[1] Jacob Bailey and Craig Zilles. 2019. uAssign: Scalable Interactive Activities for Teaching the Unix Terminal. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. ACM. https://doi.org/10.1145/3287324.3287458

[2] Jerome S. Bruner. 1966. *Toward a Theory of Instruction* (nachdr. ed.). Belknap Press of Harvard University Press, Cambridge, Mass.

[3] Stephen V. Cole. 2022. Impact of Capture The Flag (CTF)-Style vs. Traditional Exercises in an Introductory Computer Security Class. In *Proceedings of the 27th ACM Conference on Innovation and Technology in Computer Science Education*. ACM, Dublin Ireland, 470–476. https://doi.org/10.1145/3502718.3524806

[4] Hasanov et al. 2025. An Interactive Tool for Randomized Autogradable Graph Assessments. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education*. ACM, 2. https://doi.org/10.1145/3641555.3705123

[5] Juho-Heikki Holmi. 2020. *Advantages and Challenges of Using Capture the Flag Games in Cyber Security Education*. Bachelor's. University of Oulu, Oulu, Finland.

[6] Connor Nelson and Yan Shoshitaishvili. 2024. PWN The Learning Curve: Education-First CTF Challenges. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education (SIGCSE 2024)*. ACM, New York, NY, USA, 937–943. https://doi.org/10.1145/3626252.3630912