CTF Archive: Learning Beyond the Competition

Pratham Gupta
Arizona State University
Tempe, AZ, USA
prathamgupta@asu.edu

Aditya Maheshbhai Gabani
Arizona State University
Tempe, AZ, USA
agabani@asu.edu

Connor Nelson
Arizona State University
Tempe, AZ, USA
Connor.D.Nelson@asu.edu

Yan Shoshitaishvili
Arizona State University
Tempe, AZ, USA
yans@asu.edu

Abstract

Capture the Flag (CTF) competitions produce large numbers of authentic, hands-on cybersecurity exercises, but most competitions last only 24–48 hours and leave behind artifacts that are difficult for learners or instructors to reuse. We present CTF Archive, a public effort to preserve past CTF challenges as runnable, browser-accessible learning environments with validated deployments, preconfigured tools, and metadata for discovery. Unlike artifact archives that primarily preserve files, CTF Archive preserves the conditions needed to begin working: a running target, a compatible environment, configured tools, and evidence that the challenge remains solvable. As of June 2026, CTF Archive rehosted 834 challenges from more than a decade of competitions; 805 learners had solved 8,150 challenges through the public deployment. We describe the expert rehosting workflow needed to turn scattered challenge files, Dockerfiles, writeups, and solver scripts into usable exercises, and we report evidence from public usage, two learner surveys, open-source maintenance activity, and two course deployments at a large public R1 university. Our experience suggests that archived CTF challenges are most useful for education when preservation addresses both deployment and discovery: learners need to be able to start immediately, while instructors need metadata that helps them select challenges for cohesive lessons, labs, and demonstrations.

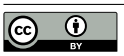
CCS Concepts

• **Applied computing** → **Education**; **Interactive learning environments**; • **Security and privacy** → **Systems security**; **Software and application security**.

Keywords

Cybersecurity Education, Capture The Flag, Challenge Preservation, Rehosting, Interactive Learning Environments

ACM Reference Format:
Pratham Gupta, Aditya Maheshbhai Gabani, Connor Nelson, and Yan Shoshitaishvili. 2027. CTF Archive: Learning Beyond the Competition. In *Proceedings of the 58th ACM Technical Symposium on Computer Science Education V.1 (SIGCSE TS 2027), February 17–20, 2027, Sacramento, CA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3857222.3860226



This work is licensed under a Creative Commons Attribution 4.0 International License.
SIGCSE TS 2027, Sacramento, CA, USA
© 2027 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-3075-7/2027/02
https://doi.org/10.1145/3857222.3860226

1 Introduction

Capture-the-Flag (CTF) competitions have become a prominent and effective form of hands-on cybersecurity education. Prior work has shown that CTF-style exercises can increase engagement, motivation, and student confidence while preserving learning outcomes relative to traditional exercises [8, 15, 33]. Other work has shown that CTF challenges exercise practical skills across topics such as cryptography, web security, reverse engineering, binary exploitation, forensics, and network security [4, 26]. Their educational value comes from immediacy: learners are placed in front of a working technical artifact, given a goal, and invited to investigate. Instead of only reading about vulnerabilities or protocols, they interact with live systems that behave in concrete and sometimes surprising ways. This makes CTF challenges valuable not only for competitors, but also for cybersecurity educators seeking realistic exercises for particular topics and independent learners seeking self-paced practice [14].

Creating these experiences requires substantial expert labor. A useful CTF challenge must be designed around a security concept, tuned to an appropriate difficulty, packaged with the right dependencies, deployed reliably, tested for solvability, and supported by infrastructure that lets participants begin working. Prior studies of CTF organizers describe challenge creation, preparation, testing, scoring, and infrastructure management as work that requires significant effort and expertise [6, 7, 28, 29, 32]. Each competition therefore represents a substantial investment of expert effort converted into a short-lived learning experience.

Yet the learning opportunity often disappears when the competition ends. Most CTFs run for only a weekend, after which services are shut down, infrastructure is dismantled, and the working environment becomes unavailable. Even when organizers publish challenge files, reusing those challenges for education requires overcoming two practical barriers: discovery and deployment.

The first barrier is discovery. During a live competition, discovery is partly supplied by the event itself: participants are expected to encounter a broad set of unfamiliar problems, and the challenge set defines the space of what to try. After the event, however, those challenges enter a much larger and less organized ecosystem. A single event may include around 20 challenges, and dozens of events run each year, meaning that more than a thousand new challenges may be created in a year. This abundance is valuable, but difficult to use

directly. Challenges are scattered across event pages, GitHub repositories, Docker images, writeups, and informal archives, often with inconsistent labels and incomplete descriptions. It is difficult for educators to know which challenge to reach for when building a cohesive curriculum around a particular topic and difficulty level. For independent learners, the same problem appears as navigation: knowing what to try next and whether it matches their current goals.

The second barrier is deployment. During a live CTF, this barrier is, again, largely solved by the competition itself: the target is already running, and the challenge has been tested to ensure that it behaves correctly. Raw artifacts rarely preserve that state after the event ends, and in cybersecurity these missing details matter as even minor changes in configuration or environment can determine whether a challenge works at all. For experienced participants, reconstructing this environment may be tedious. For novice learners, it can be prohibitive: they no longer get to jump in and start reasoning about the security concept, but must first reverse-engineer the deployment. For educators, the same burden becomes an adoption barrier: before using a challenge as an exercise or demonstration, they must determine how it should be configured, what tools are needed, and whether the intended behavior is preserved.

We observe that making CTF challenges reusable for education requires both discovery and deployment. We present CTF Archive, a system for preserving CTF challenges as runnable, browser accessible learning environments. CTF Archive collects challenges from past competitions, reconstructs and validates their original environments, records metadata such as event, category, and difficulty, packages them in configurable containers, documents how they were rehosted, and makes them available through a browser-based workspace with preconfigured tools. With the rich indexing and embeddings, we created a discoverability system (https://pwn.college/ctf-archive/search) that helps educators search for challenges aligned with course topics, difficulty levels, and curriculum goals. In this way, CTF Archive goes beyond reproducing the original competition environment: it makes past challenges easier to discover, launch, and use for teaching or self-directed practice. The archive currently rehosts more than 800 real-world challenges spanning more than a decade of CTF competitions, available through a live browser-based platform (https://pwn.college/ctf-archive) and a public GitHub repository (https://github.com/pwncollege/ctf-archive) with Dockerfiles and rehosting documentation. We describe the preservation workflow, public deployment, learner surveys, course uses, and community maintenance practices that make this archive usable for both classroom adoption and self-directed learning, changing the shelf life of CTF challenges from a weekend-long competition experience into a persistent, accessible learning resource for courses, demonstrations, and independent practice.

2 Related Work

Organizer-specific CTF archives. After their CTF events have concluded, some organizers preserve their own challenges for later use. CyLab Security Academy [11] (formerly picoCTF) preserves its CTF challenges in a playable form, with flag submission, live services, and a browser-accessible shell. Several other CTF organizers preserve their challenges through organized archives, including Ångstrom CTF [1], Imaginary CTF [13], and OOO's Archive [2],

Project	Scope	Artifacts					Playability				
	Original Challenges	Challenge Assets	Deployment Files	Writeups	Topic Metadata		Deployed Service	Flag Submission	Validated Solvable	Browser Workspace	Tools for Solving
Organizer-specific CTF archives											
CyLab Security Academy	●	◐	-	-	●		●	●	●	●	◐
Ångstrom CTF	●	●	●	-	◐		-	◐	-	-	-
ImaginaryCTF	●	●	●	●	◐		-	◐	-	-	-
archive.ooo	●	●	●	-	◐		◐	◐	-	-	-
CTF artifact aggregators											
CTFtime	-	●	-	-	◐	◐	-	-	-	-	-
Sajjadium's GitHub archive	-	●	●	◐	◐	◐	-	-	-	-	-
r3kapig's Notion archive	-	●	●	◐	◐	◐	-	-	-	-	-
ShellStorm	-	●	●	-	-	◐	-	-	-	-	-
Persistent wargames and training platforms											
pwn.college	●	-	●	●	-	●	●	●	●	●	●
CryptoHack	●	-	◐	-	●	●	●	●	●	-	◐
OverTheWire	●	-	-	-	-	●	●	●	●	-	-
pwnable.kr	●	-	-	-	-	◐	●	●	●	-	-
Hack The Box	●	-	-	-	◐	●	●	●	●	◐	◐
TryHackMe	●	-	-	-	◐	●	●	●	●	◐	◐
RET2 Wargames	●	-	-	-	-	◐	●	●	●	-	◐
wargame.nexus	-	●	-	-	-	◐	-	-	-	-	-
This work											
`CTF Archive`	-	●	●	●	◐	●	●	●	●	●	●

● provides property;
◐ partially provides property;
- does not provide property.

Table 1: Related CTF archives and training platforms.

often with source files, descriptions, flags, writeups, or setup instructions. Similarly, many individual CTF teams release challenge repositories on GitHub after a competition. These archives are valuable, but with the exception of CyLab, they generally preserve artifacts rather than a complete playable environment: a learner or educator may still need to rebuild, configure, and verify challenge behavior is intended. CTF Archive differs by targeting the broader CTF ecosystem: we do not run a CTF of our own and then archive it; instead, we reconstruct challenges from many independent competitions.

CTF artifact aggregators. There are various efforts that aggregate CTF activity and artifacts across many competitions. CTFtime [10] is the central community index for upcoming and past CTF events; as part of its archival role, it has also preserved many community writeups for prior challenges, although this has become less common in recent years. Archives maintained by Sajjadium [23], r3kapig [21], and Shell-Storm [25] perform related preservation work by aggregating challenge files from many events. These collections make otherwise scattered assets easier to find, and we have drawn from them as part of this work. Their primary unit of preservation, however, is the raw artifact: source files, binaries, challenge descriptions, and sometimes writeups or setup files. It is becoming increasingly common for CTF authors to release Dockerfiles, which dramatically helps reproduce challenge environments, but this is still not the default across most CTFs and does not by itself guarantee a working live service. Archived artifacts generally do not guarantee that a challenge still builds, runs with the intended dependencies and permissions, exposes the intended service, or remains solvable in a known environment. CTF Archive

builds on this aggregation layer by performing the additional expert labor needed to turn archived assets back into usable exercises, and by attaching metadata that helps learners and educators discover challenges focused on specific topics.

Persistent wargames and training platforms. Whereas most CTF competitions run for a 24–48 hour weekend window, other security challenge sites are designed to remain available indefinitely. These systems are often called wargames. Examples include pwn.college [17, 18], CryptoHack [9], OverTheWire [19], and pwnable.kr [20], as well as commercial training platforms such as Hack The Box [12], TryHackMe [30], and RET2 Wargames [22]. Indexes such as wargame.nexus [34] help learners discover these persistent challenge sites. These platforms solve availability by design: their challenges are intended to remain playable, and many provide live targets, scoring, or guided learning paths. Most, however, are not trying to preserve past CTF competitions, and many still expect learners to provide their own local tooling or working environment. CTF Archive instead takes challenges created for ephemeral competitions, performs the rehosting work needed to make them persist as educational resources, and attaches metadata that helps learners and educators find challenges aligned with particular topics.

3 Design

CTF Archive is designed around two requirements that determine whether past CTF challenges can become reusable educational resources: *deployability* and *discoverability*. Deployability means that a challenge is not merely available as a collection of files, but can be started in a known-good environment, interacted with as intended, and solved through the expected security concept. Discoverability means that learners and educators can understand what a challenge is about, what concepts it exercises, how difficult it is likely to be, and where it fits in a broader learning path. Figure 1 summarizes this workflow. The central design challenge is that neither property is reliably preserved by raw CTF artifacts alone.

Recovering Challenge Materials. CTF Archive begins by gathering challenge materials from many sources. Existing aggregators and archives provide an important foundation: they collect large numbers of challenge files, descriptions, binaries, source code, writeups, and occasionally deployment files such as Dockerfiles. We draw heavily from these efforts, but they are only one layer of the preservation process. Many challenges require searching beyond aggregators into organizer pages, team GitHub repositories, personal writeup repositories, Docker image references, and blog posts written by participants. These sources often contain partial but crucial evidence: a missing binary, the original flag, a command used to start a service, a dependency version, or a solver script that reveals what behavior the challenge was meant to expose.

This broader recovery work is necessary because public artifacts are uneven. Some challenges include a Dockerfile, but the Dockerfile may refer to a private registry, depend on an image that no longer exists, omit runtime configuration, or fail because of dependency drift. Other challenges have no deployment files at all, or only appear in a writeup where the author saved the files needed for their own solution. In practice, preserving the challenge requires reconstructing a scattered record of the original competition environment from whatever pieces remain.

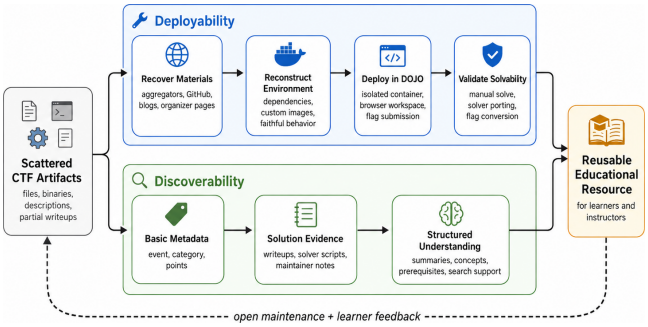


Figure 1: CTF Archive preservation workflow from scattered artifacts to runnable, discoverable educational resources.

Reconstructing Runnable Environments. Once the challenge materials are collected, the next task is to turn them into a working deployment. CTF Archive maintains a broad preconfigured environment with common cybersecurity tools, dependencies, and libraries. We first test whether the recovered artifacts build and run correctly in that environment. When dependencies are missing, we add them when doing so is compatible with the rest of the archive. When dependencies conflict, when the challenge requires unusual system behavior, or when the vulnerability depends on a particular runtime, we create a custom image for that challenge.

This stage is the most labor-intensive part of the design. Rehosting often requires fighting compilation errors, recovering old library versions, debugging crashes, resolving architecture assumptions, and understanding whether surprising behavior is a bug in the rehosted environment or the intended vulnerability. This distinction is especially difficult in cybersecurity: a crash, permission failure, memory error, or strange network behavior may be exactly what the learner is supposed to investigate. Some older challenges depend on obsolete tooling such as Python 2 or outdated system libraries. When possible, we lightly port such challenges to newer dependencies, but only when doing so does not change the core security concept. Writeups and solver scripts are therefore often necessary evidence for understanding the intended solve path and deciding whether a rehosted challenge is faithful.

Manual solving remains the strongest validation method. In many cases, existing solver scripts must be modified because subtle changes in the rehosted environment alter paths, hostnames, ports, timing, library versions, or interaction details. In other cases, no usable solve script exists, so the maintainer must solve the challenge manually to determine whether the environment is correct. More recently, large language models have also become useful assistants for interpreting old code, diagnosing build errors, summarizing writeups, and suggesting likely dependency fixes, but the final decision about whether a challenge preserves the intended behavior remains a human expert judgment.

DOJO-Based Deployment. The live archive is deployed using DOJO, the open-source infrastructure behind pwn.college [16]. DOJO was designed to make hands-on cybersecurity challenges easier for students to access and easier for instructors to deploy: it provides isolated containerized challenge environments, browser-accessible workspaces, flag submission, progress tracking, persistent storage, and instructor-managed challenge collections. The DOJO paper reports that the platform has supported more than

Category	Challenges	Solves
Cryptography	295	5,490
Reverse Engineering	146	1,011
Binary Exploitation	180	645
Miscellaneous	70	419
Web Security	47	286
Forensics	64	195
OSINT	23	91
Hardware/RF	2	11
Blockchain	2	2
Malware	5	0
Total	834	8,150

Table 2: Challenges and solves by category.

10,000 learners across more than 100 countries and more than 100 academic institutions, making it a mature foundation for hosting long-running cybersecurity exercises rather than one-off competition infrastructure [16].

This infrastructure matters for CTF Archive because it lets the archive preserve not only the target, but also the workspace needed to work on it. Each archived challenge can be started as an isolated container that learners access through a web terminal, web-based VS Code, web-based graphical desktop, or SSH. DOJO's browser-based interfaces and preconfigured tooling allow learners to perform discovery, implementation, and debugging inside the hosted environment, reducing the need to install local virtual machines, compilers, debuggers, reverse-engineering tools, or challenge-specific dependencies before beginning. For instructors, the same infrastructure provides a mechanism for making a challenge immediately available to a class without asking each student to reproduce the environment on their own machine.

Adapting a challenge to DOJO is itself a deployment step. In CTF Archive, challenge files are placed in a root-owned `/challenge` directory, learners work from a persistent `/home/hacker` directory, and the challenge must ultimately produce a DOJO-submittable flag. The DOJO environment also changes the security context in which a challenge runs: the platform uses a root-owned `/flag` file and isolates learners inside containers. Rehosted challenges must therefore be tested inside the final DOJO environment, not only in a local build container, to ensure that the intended solve still works and that the new environment has not introduced an unintended way to read the flag or bypass the challenge.

Flags and Solvability. Flag handling is a recurring compatibility problem because many CTF challenges embed the original flag directly in an artifact. In DOJO, however, learners submit platform flags, so CTF Archive uses a generic flag converter: the challenge stores a hash of the original CTF flag, and when the learner provides that flag, the converter verifies it and returns the DOJO flag.

Every challenge is manually tested after being adapted to the hosted environment. This final test matters because the learner ultimately interacts with the DOJO deployment, not with the maintainer's intermediate reconstruction. A challenge may build successfully but still fail under the final permissions, filesystem layout, network configuration, or flag-handling mechanism. Solvability testing is therefore both a quality-control step and a pedagogical guarantee: learners and instructors should be able to trust that the exercise is broken only in the way the challenge author intended.

Metadata for Discoverability. Deployability makes a challenge runnable; discoverability makes it useful at scale. CTF challenges usually come with basic metadata such as a name, event, category, and point value. Aggregators, especially CTFtime [10], are useful for recovering this basic context. Categories such as binary exploitation, reverse engineering, cryptography, web security, forensics, and miscellaneous give learners and instructors a first approximation of topic, while point values often provide a rough signal of difficulty within the original competition.

CTF Archive extends this metadata with knowledge recovered during rehosting. We collect writeups and solver scripts when available, and when manual solving is necessary, we create or adapt solution notes and scripts as part of the validation process. These materials serve deployment by proving that the challenge works, but they also serve discovery by revealing what concepts, prerequisites, tools, and techniques the challenge actually involves. Because many original writeups and scripts assume the competition environment, they often need to be ported to the rehosted environment; the updated solver becomes part of the preserved record of the challenge.

To make this information more useful, we use large language models to synthesize a standardized understanding of each challenge from the available metadata, artifacts, writeups, and solution material. This includes summaries of what the challenge is about, what concepts it involves, what prerequisites it assumes, and how the solve proceeds. Human review is still important, but we have found this synthesis useful for standardizing challenge descriptions across many competitions and categories. The resulting structured understanding supports richer indexing, search, keyword matching, and embedding-based discovery so that educators can find challenges aligned with particular lessons and learners can find appropriate next problems. Instructors can visit the platform to query our metadata for a curated list of challenges.

Open Maintenance. CTF Archive is intentionally maintained in public. The repository stores recovered artifacts, Dockerfiles or deployment materials when available, solver notes or scripts when possible, and `REHOST.md` documentation describing how a challenge was reconstructed. This makes the archive easier to repair when dependencies drift, platform assumptions change, or a challenge turns out to have been incompletely preserved.

The live platform also creates a feedback loop. More than 800 learners have interacted with the archive, and this community helps discover issues that survive initial validation. Learners and contributors report issues and submit fixes for challenges through GitHub. As a result, the archive improves over time: use reveals defects, public documentation makes repair possible, and community contributions help preserve more of the educational value produced by CTF competitions.

4 Evaluation

We evaluate CTF Archive as a preservation and education tool using three sources of evidence: public usage analytics from the global deployment, learner surveys, and course deployments at a large public R1 university. CTF Archive was opened to a public audience in May 2024. As of June 2026, 805 learners had solved

Reason for using CTF Archive	Respondents
Improve cybersecurity skills or learn techniques	30
Fun and enjoyment of solving puzzles	29
Explore a large variety of archived challenges	21
Practice for future CTF competitions	15
Compete or compare performance against others	9
Course or training requirement/recommendation	7

Table 3: Primary reasons for using CTF Archive.

Solve range	Learners	Learners (%)	Solves	Solves (%)
1 solve	193	24.0%	193	2.4%
2–4 solves	140	17.4%	390	4.8%
5–9 solves	252	31.3%	1,825	22.4%
10–24 solves	180	22.4%	2,456	30.1%
25–49 solves	29	3.6%	847	10.4%
50+ solves	11	1.4%	2,439	29.9%
Total	805	100.0%	8,150	100.0%

Table 4: Learner distribution by number of solved challenges.

8,150 challenges across 834 archived challenges. These learners include a global public audience, a cohort of third-year undergraduate cybersecurity students, and a GitHub community that contributes fixes, reports issues, and adds archived challenges.

Public usage. The public deployment shows that archived CTF challenges can continue to attract and sustain learner activity after competitions ended. Table 4 shows that 805 learners completed at least one challenge, with an average of 10.1 solves per learner. Most learners solved fewer than 10 challenges, consistent with lightweight exploration or occasional practice. At the same time, 220 learners solved at least 10 challenges, and the 40 learners who solved at least 25 challenges account for 40.3% of all solves, suggesting that the archive also supports deeper repeated engagement.

Table 2 breaks current archive and solve activity down by normalized challenge category. Cryptography accounts for the largest share of solves and archived challenges because cryptography and reverse-engineering challenges are often easier to rehost than service-heavy challenges because they more often depend on static files rather than fragile networked deployments. The distribution reinforces one central design pressure of CTF Archive: categories that depend on precise services or operating-system behavior require substantially more rehosting labor.

The archive also functions as an open maintenance project. The GitHub repository receives roughly 1,600 clones and 2,000 views per month, has 100 stars, and has closed more than 100 issues. Community members have submitted more than 70 pull requests to fix existing challenges, improve metadata, or add newly archived material. This contributor activity is part of the evaluation because long-term preservation depends not only on an initial rehosting effort, but also on continued repair as dependencies, infrastructure, and learner expectations change.

Learner survey. To better understand learner perceptions, we conducted two optional surveys: one in 2025 with 18 respondents and one in 2026 with 36 respondents. Approval for the study was sought from our institution's Institutional Review Board before undertaking it, and the IRB designated the study as exempt. All survey questions were optional. Across the 54 total respondents, the highest completed academic level was high school for 19 respondents, associate degree for 1, bachelor's degree for 21, master's degree for 9, and doctorate for 4. In the 2026 survey, respondents self-rated their cybersecurity experience as beginner (12), intermediate (20), or advanced (4), indicating that the archive is used by learners across a range of preparation levels.

Survey responses were strongly positive about both usability and learning value, as shown in Figure 2. In the 2026 survey, 88.9% of respondents agreed or strongly agreed that CTF Archive is user friendly and easy to navigate, and 88.9% agreed or strongly agreed that having tools and dependencies already configured made challenges easier to start. Similarly, 83.3% agreed or strongly agreed that solving rehosted, preconfigured challenges was easier than manually downloading and configuring past CTF challenges. These responses directly support the deployment argument in this paper: learners value being able to start working on the security concept without first reconstructing the challenge environment.

Learners also reported educational value. In the 2026 survey, 86.1% agreed or strongly agreed that CTF Archive helped them practice or deepen their understanding of cybersecurity concepts, 86.1% agreed or strongly agreed that the lack of live CTF time pressure helped them learn on their own time, and 86.1% agreed or strongly agreed that they learned at least one new concept. A smaller but still substantial 77.8% agreed or strongly agreed that CTF Archive helped bridge core security concepts and practical application, and 80.6% agreed or strongly agreed that the variety of challenges helped them understand different kinds of problems that appear in live CTFs.

Table 3 shows why respondents reported using the archive. The most common reasons were personal learning, enjoyment, and curiosity about the variety of archived challenges. This matters for an archive intended to serve both structured education and self-directed practice: many learners arrive not because a course requires them to, but because the preserved challenges create an appealing space for exploration. At the same time, 15 respondents used the archive to prepare for future CTF competitions, and 7 encountered it through a course or training context.

Responses about public writeups were more mixed. In the 2026 survey, 12 respondents agreed or strongly agreed that they use publicly available writeups while solving CTF Archive challenges, 15 disagreed or strongly disagreed, and 9 were neutral. Free responses suggest that learners who do use writeups often treat them as hints or post-solve explanations rather than as direct substitutes for solving. When asked what they gained from solving CTF Archive challenges, respondents named a wide range of technical topics, including reverse engineering, binary exploitation, cryptography, format-string exploitation, fuzzing, enumeration, Linux concepts, and tool familiarity. Several also emphasized process-level lessons such as persistence, patience, and becoming more systematic when approaching unfamiliar challenges.

Course deployments. CTF Archive was used in two deployments within a required cybersecurity course at a large public R1 university. In Fall 2024, cumulative learner activity increased around the course deployment. In that deployment, students could solve any

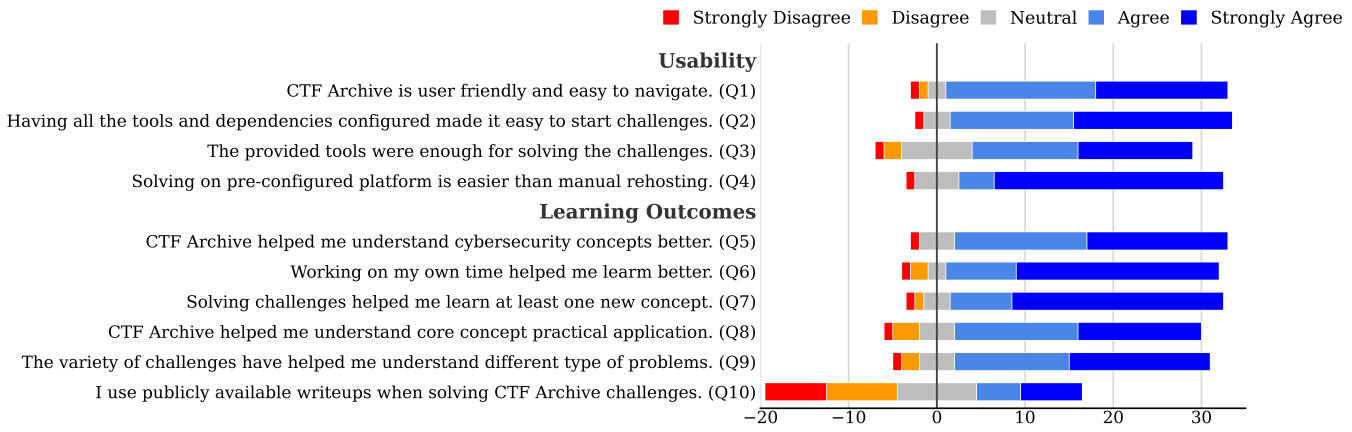


Figure 2: Learner responses to Likert survey questions about CTF Archive. There were 54 total respondents across two surveys.

archived challenges they wanted, and the course awarded credit based on the number of solved challenges. This open-ended deployment demonstrated that CTF Archive can absorb course-scale use and give students choice across preserved challenge material.

In Summer 2026, we piloted a more curated deployment. Using metadata collected during preservation and rehosting, we selected three challenges of varying difficulty to complement concepts already present in the course curriculum. Students could complete the challenges for extra credit and were invited to complete an optional survey afterward. Among 13 responses, 8 agreed that the selected challenges helped them understand classroom concepts better and 5 were neutral. Most responses agreed that the challenges were educational and that students knew what to do when stuck. This smaller deployment suggests that the metadata collected by CTF Archive can help instructors choose archived challenges as targeted exercises or demonstrations rather than treating the archive only as a large undifferentiated pool of practice problems.

5 Discussion

Our deployment experience suggests that preserving CTF challenges is only the first step toward educational reuse. The archive becomes most useful when educators can curate challenges for specific learning goals, while the same runnable and documented collection can also support research uses beyond the classroom.

Course adoption. The two course deployments exposed an important distinction for educators adopting a large archive. In the first deployment, students were allowed to solve any challenge they wanted. This model preserved learner agency and made the archive easy to integrate at course scale, but it also encouraged some students to optimize for readily solvable challenges rather than challenges aligned with a particular lesson. This is not a failure of the archive; it is a reminder that a large collection of authentic CTF problems behaves differently from a designed curriculum.

The second deployment used the archive more selectively. Instructors chose a small set of challenges using preservation metadata and aligned those challenges with course concepts. This model supports cohesive lessons, labs, and demonstrations because the instructor can use the archive as a library of deployable exercises rather than as an undifferentiated pool of extra-credit work. Our experience suggests that CTF Archive is most useful in courses when it supports both modes: open-ended exploration for practice and curiosity, and curated selections for targeted reinforcement of recently encountered concepts.

Research reuse. CTF Archive has also begun to support research beyond its immediate educational use. Because the archive combines more than 800 rehosted challenges with documentation, rehosting notes, solvability checks, and preserved challenge artifacts, it can serve as a practical dataset for evaluating and training systems that reason about cybersecurity tasks. The CAISI Cyber Evaluations framework developed by the National Institute of Standards and Technology (NIST) incorporated CTF Archive as an externally tracked benchmark using the UK AI Security Institute's AISI framework for large language model evaluations, with rehosted real-world CTF challenges used as tasks for evaluated agents [5, 31]. Another research project used CTF Archive as part of a training dataset for large language models [35]. These early uses suggest that preserving CTF challenges as runnable, documented environments can make them useful not only for learners and instructors, but also for researchers studying automated vulnerability discovery, exploit generation, and cyber-reasoning systems.

6 Conclusion

CTF Archive preserves the educational value of CTF competitions by rehosting more than 800 real-world challenges as browser-accessible, preconfigured learning environments. By addressing both deployment and discovery, the archive turns short-lived competition problems into reusable exercises for independent learners, instructors, and researchers. Our public deployment, learner survey, and course experiences suggest that this preservation work can make expert-created cybersecurity challenges easier to find, start, teach with, and learn from long after the original event ends.

Acknowledgments

This work would not have been possible without the vibrant enthusiasm of the CTF community, pwn.college, and the generous support of the Department of Defense.

References

[1] ÅngstromCTF. 2026. https://angstromctf.com/.

[2] Archive.ooo. 2026. https://archive.ooo/.

[3] Carl Boettiger. 2015. An Introduction to Docker for Reproducible Research. *SIGOPS Operating Systems Review* 49, 1 (2015), 71–79.

[4] Tanner J. Burns, Samuel C. Rios, Thomas K. Jordan, Qijun Gu, and Trevor Underwood. 2017. Analysis and Exercises for Engaging Beginners in Online CTF Competitions for Security Education. In *2017 USENIX Workshop on Advances in Security Education (ASE '17)*. USENIX Association. https://www.usenix.org/conference/ase17/workshop-program/presentation/burns

[5] CAISI Cyber Evals. 2026. https://github.com/usnistgov/caisi-cyber-evals.

[6] Benjamin Carlisle, Michael Reininger, Dylan Fox, Daniel Votipka, and Michelle L. Mazurek. 2020. On the Other Side of the Table: Hosting Capture-the-Flag (CTF) Competitions, An Investigation from the CTF Organizer's Perspective. In *Proceedings of the 6th Workshop on Security Information Workers (WSIW)*.

[7] Ericka Chickowski. 2024. 7 Lessons Learned From Designing a DEF CON CTF. *Dark Reading*. Online; accessed July 2026.

[8] 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 (ITiCSE '22)*. 470–476. doi:10.1145/3502718.3524806

[9] CryptoHack. 2026. https://cryptohack.org/.

[10] CTFtime. 2026. https://ctftime.org/.

[11] CyLab Security Academy. 2026. https://cylabacademy.org/.

[12] Hack The Box. 2026. https://www.hackthebox.com/.

[13] Imaginary CTF. 2026. https://imaginaryctf.org/ArchivedChallenges.

[14] Stylianos Karagiannis and Emmanouil Magkos. 2021. Adapting CTF Challenges into Virtual Cybersecurity Learning Environments. *Information and Computer Security* 29, 1 (2021), 105–132. doi:10.1108/ICS-04-2019-0050 Online; accessed July 2026.

[15] Kees Leune and Salvatore J. Petrilli. 2017. Using Capture-the-Flag to Enhance the Effectiveness of Cybersecurity Education. In *Proceedings of the 18th Annual Conference on Information Technology Education (SIGITE '17)*. ACM. doi:10.1145/3125659.3125686 Online; accessed July 2026.

[16] Connor Nelson and Yan Shoshitaishvili. 2024. DOJO: Applied Cybersecurity Education in the Browser. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education (SIGCSE '24)*. doi:10.1145/3626252.3630836

[17] 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 '24)*. 937–943. doi:10.1145/3626252.3630912

[18] Connor Nelson, Robert Wasinger, Adam Doupé, and Yan Shoshitaishvili. 2026. Open Cybersecurity Education: Five Years of pwn.college. In *Proceedings of the 57th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE TS 2026)*. ACM, 743–749. doi:10.1145/3770762.3772636

[19] OverTheWire. 2026. https://overthewire.org/.

[20] pwnable.kr. 2026. https://pwnable.kr/.

[21] R3kapig. 2026. https://r3kapig-not1on.notion.site/Index-docs-format-09007cb92ef649838d8057a64f0d99dc.

[22] Ret2 Wargames. 2026. https://wargames.ret2.systems/.

[23] Sajjadium. 2026. https://github.com/sajjadium/ctf-archives.

[24] Tao Sauvage. 2025. Behind the Scenes at Hammercon 2025: The CTF Challenge. *Anvil Secure Blog*. Online; accessed July 2026.

[25] Shell-Storm. 2026. https://web.archive.org/web/20260227051950/http://shell-storm.org/repo/CTF/.

[26] Valdemar Švábenský, Pavel Čeleda, Jan Vykopal, and Silvia Brišáková. 2021. Cybersecurity Knowledge and Skills Taught in Capture the Flag Challenges. *Computers & Security* 102 (2021), 102154. doi:10.1016/j.cose.2020.102154

[27] Valdemar Švábenský, Jan Vykopal, and Pavel Čeleda. 2019. Towards Learning Analytics in Cybersecurity Capture the Flag Games. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*. 1255. doi:10.1145/3287324.3293816

[28] Clark Taylor, Pablo Arias, Jim Klopchic, Celeste Matarazzo, and Evi Dube. 2017. CTF: State-of-the-Art and Building the Next Generation. In *2017 USENIX Workshop on Advances in Security Education (ASE '17)*. USENIX Association. https://www.usenix.org/system/files/conference/ase17/ase17_paper_taylor.pdf

[29] Erik Trickel, Francesco Disperati, Eric Gustafson, Faezeh Kalantari, Mike Mabey, Naveen Tiwari, Yeganeh Safaei, Adam Doupé, and Giovanni Vigna. 2017. Shell We Play A Game? CTF-as-a-service for Security Education. In *Proceedings of the USENIX Workshop on Advances in Security Education (ASE 2017)*. USENIX Association. https://www.usenix.org/system/files/conference/ase17/ase17_paper_trickel.pdf

[30] TryHackMe. 2026. https://tryhackme.com/.

[31] UK AI Security Institute. 2026. https://github.com/UKGovernmentBEIS/inspect_ai/.

[32] Jan Vykopal and Miloš Barták. 2016. On the Design of Security Games: From Frustrating to Engaging Learning. In *2016 USENIX Workshop on Advances in Security Education (ASE '16)*. USENIX Association. https://www.usenix.org/conference/ase16/workshop-program/presentation/vykopal

[33] Jan Vykopal, Valdemar Švábenský, and Ee-Chien Chang. 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)*. ACM, 752–758. doi:10.1145/3328778.3366893

[34] wargame.nexus. 2026. https://wargame.nexus/.

[35] Terry Yue Zhuo, Dingmin Wang, Hantian Ding, Varun Kumar, and Zijian Wang. 2025. Training Language Model Agents to Find Vulnerabilities with CTF-Dojo. arXiv preprint arXiv:2508.18370. https://arxiv.org/abs/2508.18370