

CTF^2 (CTF Task Fight)

Overview (Pitch)

Welcome to CTF^2! AppSec Village is proud to present our official DEF CON CTF Contest. This competition's goal is to reward the best CTF Developers in the global community. This is a competition where the winners are not the players, *but the creators* of the challenges.

We will ask you to submit your task in an appropriate format for CTFd deployment, under one of the 4 CTF categories (web, reversing, secure coding practices, crypto). All approved challenges will be played as part of the AppSec Village DEF CON CTF for judging.

Winners are selected by judges and players.

*The finalists will get to compete for **Best Overall Level** and win prizes according to the following*

Prizes:

1st Place- \$2,000 USD

2nd Place- \$1,000 USD

3rd Place- \$500 USD

CFL Conditions:

Each CTF submission must be under one of the four categories (see below) and must be completely new, which means they cannot have been used in other competitions, or have any writeup presently available on the internet. This will be verified by the judges. It must fit other specifications mentioned below, including the uniform flag format, in a CTFd format, and have a write-up in English.

Categories:

1. WEB:

This category should consist of web (HTTP) vulnerabilities (CSRF, XSS, SQLi, SSRF, etc.). The players should NOT be able to find the vulnerability through the use of scanners. If a vulnerability could be found by a scanner, it should be provided in the challenge description. Challenges should be self-contained by infrastructure.

2. SECURE CODING PRACTICES:

This category should be based on either teaching or implementing secure coding practices. Developers should create insecure pieces of code for players to either correct, or learn about a specific secure coding practice, and then best select an answer that best fits that definition. This can be done in many ways, and this new category will leave the creative freedom of that up to the developers. CTFd allows for users to submit code in a language of the CTF designer's choice OR select multiple-choice options. This allows this new category the freedom to have users implement a CTF that can either

1. Teach about secure coding practices in a way you find both unique, and viable.
2. Correct insecure code in order to retrieve a flag. For example, you have a piece of PHP that allows for SQL injection, and the player must correct that code to prevent an SQL injection.

3. CRYPTO:

This category should consist of either a file or of one or more online services (TCP or HTTP) implementing a crypto protocol or weakness that players must exploit by solving a cryptographic challenge. If players just need to reverse engineer the protocol or algorithm used, the challenge must be submitted to reversing instead.

4. REVERSING:

This category should consist of either a file or one or more online services on which players must reverse engineer a system or binary to discover a flag. This category is not steganography related and does not exploit a vulnerability. Players should be able to extract or deduce the flag without depending on luck. Challenges must test skills related to application security. Reverse Engineering is a crucial component of most vulnerability research, and tasks in this category should test skills that could be useful in application security. Examples of this include, but are not limited to the type of deobfuscation needed when looking for vulnerabilities in closed-source code, working on top of niche or old architectures when doing security research, hardware security research, or even debugging the communication protocol between a JavaScript web application and the server. Note that challenges in this category should not exploit a vulnerability themselves, and are just meant to test the skills of the reverse engineering stage of vulnerability research.

Challenge Submission

- Third-party dependencies should be clearly marked!
- Must be completely novel and never seen before
- Must follow the infrastructure specifications
- Must meet the criteria for one of the 4 categories (web, reversing, secure coding practices, and crypto)

- Must include a README.md file that includes (in English) three sections:
 - Public Challenge Description. What players will see, including hints (if any).
 - Secret Design Specification. Like a slide deck, design document or similar.
 - This will be what the judging panel will use to score the challenge.
- Full Solution Explanation. Like a CTF challenge write-up
 - This will be what the judging panel will use to score the challenge.
 - Must include an exploit and solver that works out of the box
 - Challenge must be solvable in less than 2 days (the CTF lasts longer, but challenges shouldn't take longer than 2 days to solve)

Flag format is **ASV{ }**. The format within should be strong, often including a mix of letters, numbers, and special characters. An example could be: **ASV{Fr0m_3D3N}** or **ASV{4PP_53C_1s_fun!}**, however, the strings can also be random. **ASV{452qdT5f^j80pG6}**, what is absolutely mandatory is the prefix **ASV** and the curly brackets **{ }** at the beginning and end.

Must be submitted before the **HARD DEADLINE of 30 JULY 2024!**

For tips and suggestions on how to make great CTF tasks check [The CTF Design Guidelines](#), [The Many Maxims of Maximally Effective CTFs](#), and [PPP Suggestions for Running a CTF](#).

Judge Responsibilities

The judges' responsibilities are as follows:

1. Play all the CTFs as intended. Attempt them as you would if you were a player without the walkthrough.
2. Once completing the task, read the walkthrough to understand the intended solution(s)
3. For each level score on a 1 to 5 scale, with 5 being the highest and 1 being the lowest. You will have to justify this score later to other judges for the best categories.
 - a. Taking notes might be useful, especially for selecting the best of each category.
4. Once all the CTFs have been played, submit your score through the available form, and all of the points will be tallied up, and the top 3 challenges with the highest scores will win.
5. You will have to meet with the other judges to pick challenges that fit the best in each category, the notes will be useful here as they will have to be elected by all the judges. The justification will be given at the awards ceremony as to why each task was picked.

Rubric

Judges will grade this using a 1 to 5-point-based system grading on several separate categories. 5 is fantastic, 1 needs lots of improvement. The score will be added up from all the judges, and later players to determine the best challenge(s) in

WEB/REVERSING/CRYPTO/SECURE CODE.

Overall categories for judging a challenge:

Write up: Score based on 1 to 5. You'll want to base your score on ideas such as:

- Is this easy to follow?
 - Is it step by step?
- How easy is this to read?
 - Does this cover the entire challenge?
- Could a non-technical person follow this?
 - Would someone else be able to replicate this using just this write-up as a guide?

Challenge (Presentation): Score based on 1 to 5. You'll want to base your score on concepts such as:

- Is the challenge neat?
 - If there's a webpage, does it look tidy, professional and realistic?
- Is the code readable?
 - If there's code involved, is it neat, readable and runs with little errors or bugs?
- Does this challenge look like it was made by a professional?
 - Is there time, and effort visibly added to the challenge's presentation?
- Are hints needed?
 - Is the description straightforward and easy to follow or are supplement hints necessary?

Challenge (Functionality): Score based on 1 to 5. You'll want to base your score on concepts such as:

- Does the challenge work?
 - Are you able to complete the challenge (and in a timely manner)?
- Is the challenge tidy, with little rabbit holes or points of confusion?
 - Is it direct without needless dead ends to distract players?
- Does the challenge have any bugs?
 - Is there any specific thing the user could (accidentally) do to cause the entire challenge to break?
- Does it work well in CTFd?

- Is it formatted to CTFd correctly?
- Is it easy to connect to?
 - Is the challenge easy to access?

These are concepts you'll want to consider but can also score it on other things that you notice/want to bring attention to. All points for the categories are then added up for a total of 15 if it's a perfect challenge. After that, all of the points amongst the judges for each challenge will be added up overall and used to determine the best challenge. Each judge will then select a Best x Challenge for a special challenge coin. See below for category info.

Best Challenge Categories

Best Implemented Challenge

Great attention to detail. Great User Interface. Superb documentation. Overall high quality. Superb stability (if applicable). Abuse protection. Great monitoring or troubleshooting.

Best (educational) Beginners Challenge

Complete beginners (CS students) could solve the challenge and learn something from it. Can be used as a "gateway" to CTFs. Required knowledge is presented to players naturally. Dead-ends and next steps are clearly defined.

Most fun/engaging Challenge

Challenges that are fun and engaging to solve. Either because of the mechanics or because of the overall problem-solving experience designed by the author. Must be fun for most players, so if a challenge is very hard, it might qualify instead for Best Implementation or Most innovative instead.

Best (solvable) Difficult Challenge

The best challenge for experienced players. Could be a 500 points task in top-tier CTFs. The solution is reasonable. Can't be solved by pure luck. An acceptable amount of work.

Most Innovative/Original Challenge

The solution requires a novel exploitation idea, a new vulnerability class, a never-seen-before primitive, or is otherwise unique in some positive way.

Awards:

For CTF Players:

- 1st, 2nd and 3rd place get various Swag. (Tshirts, stickers etc)

For challenge creators:

- 1st place: \$2,000 USD You will also receive swag!
- 2nd place: \$1,000 USD, and various swag.
- 3rd place: \$500 USD and swag.