

sCTF 2016 Q1 Guide

- Advice from previous quarters:
 - In general, have more of a range of problems regarding difficulty; many said they were much too difficult, others said they were good. It depends on how good a team is, but if we have a greater number of problems, we should be able to appease both sides.
 - This does not mean make the problems easier. It means make **more** problems to fill in the difficulty gap. People liked the hard problems; lots said they were rewarding to solve in comparison to some other CTFs with just one or two actually difficult problems.
 - More pwnables and forensics.
 - Especially more pwnables. If you need servers, just ask and I'll get them.
 - Some people liked recons and some didn't. Focus on not making recons and, at the end, if we want more problems, we'll do collaborative recons.
 - If a problem has the possibility for multiple answers, keep that in mind. For example, with Aaron's recon, accept "ave" as well as "avenue".
 - Keep problems compatible across operating systems universally. If you can only compile for a specific OS, modify it or rewrite in a different language to keep compatibility.
 - Be clever in problems. A lot of people liked the Punchagles series because they were new and interesting, attempt to make problems that are both fun and require skill.
 - One of my favorite responses: "They were really hard. I go home every night and cry about how difficult the problems are. I lose sleep and I take my anger out on my loved ones. This competition changed me from an expert coder to a monster."
- Staff advice:
 - Try to be on IRC in the background a bit more if possible. Hopefully this competition has better timing as to not be during finals and others.
 - Someone gave a shoutout to Aaron for using the banhammer on annoying teams.
 - If you are going to be AFK in the IRC, make sure to mark yourself as so.
- Problems should be written with the following in mind:
 - Problem Name
 - Description
 - Hint (not necessary)
 - Flag - **in format: sctf{FLAG}**
 - Points
 - Problem Type
 - Required Files (not necessary)
- Problems should be put in the repository as **problem.json** with the following structure:

```
[
  {
    "title" : "Problem Title",
    "description" : "Problem description",
    "hint" : "Problem hint (leave blank if not necessary)",
    "solution" : ["sctf{flag1}", "sctf{flag2}", "sctf{etcflag}", "sctf{all flags are case insensitive}"],
    "points" : 50,
    "problemtype" : 5,
    "filez" : ["file1.zip", "file2.zip", "etcfile.zip", "leave empty array if no files.jpg"],
    "author" : "Problem Author"
  }
]
```

- *Description* should contain the description of the problem.
- *Hint* should contain the problem hint, if there is one. If not, leave an empty string.
- *Solution* should contain the solution(s) to the problem as an array. If there is one solution, just use one string in the array. If there are more than one, add more strings. Simple.
- *Files* should be an array of the files. If there are no files, leave the array empty. Place the files in the same directory as the problem's JSON file.
- *Points* should simply be a number which is the point value of the problem.
 - Point values are to be done in increments of 5 or 10, with the introduction problem or survey being worth 5 points. This is an estimated point value, and problems will be reassessed after all are turned in to determine the actual point values based on the comparative difficulty to other problems.
- *Problemtype* should be a number denoting the problem type number.
 - Reference of problem type numbers:
 - 1 – Web
 - 2 – Cryptography
 - 3 – Forensics
 - 4 – Pwning
 - 5 – Recon
 - 6 – Reversing
 - 7 – Game
 - 8 – Algorithmic
- *Author* is a string containing the name of the author.

- We are in need of all problem types except for **game**.
- When a problem is completed, upload it to the GitHub repository. Make a new folder within the quarter's folder, and title the folder with the problem's name.
- Within that folder, put your JSON file and any other necessary problem files, as well as a **readme.md** file containing instructions on how to solve the problem and advice for anyone asking questions.