- (MISC) Tag: Let Me Play
    - Description:
        - You go to play tag with the cool kids, but when you start to run around with them, they ask you if you're part of the CTF Club Discord. Sheepishly, you say that you aren't, and they kick you out. Recess ends soon… hurry, get the full invite link to the CTF Club discord so you can join them before you go back to class!
          The flag is in the format: CTF{https://discord.com/invite/TEXTHERE}
    - Flag: CTF{https://discord.com/invite/4B2CmPZvC7}
    - Flag is in CTF Club linktree found in Instagram!
- (Misc) Minesweeper
    - Description:
        - What is the coordinate of where the last flag should be planted? Format the flag as CTF{(x,y)}
    - Solution:
        - Flags are where mines are suspected to be in the game minesweeper. The last flag is in (9,5) because there must be 3 flags next to the 3 and 2 flags next to the 2 that is immediately below the 3.
    - Flag:
        - CTF{(9,5)}
- (Forensic) Where's Waldo?
    - Description:
        - Waldo just posted this pic, and it looks so pretty! Where are they? Wrap the name of the beach, in all caps and with no spaces, in CTF{}
    - Solution:
        - Reverse image search the photo
    - Flag:
        - CTF{SOUTHBEACH}
- (Crypto) Memory
    - Description:
        - Let's play a game of memory! Match the plaintext words to its corresponding ciphertext. There is an order to this, since the plaintext words can be organized in a certain order to spell out a sensible phrase.

          The flag is four numbers (wrapped in CTF{}), derived from the position and order of the ciphertext cards, if they were on a keypad. For example, if the order of the plaintext phrase is "codes flags crack seize" and the ciphertext for 'codes' is in the upper lefthand corner, the first character of the flag would be '1'. If the ciphertext for 'flags' is in the middle of the second row, the second character of the flag would be '5'.
    - Solution:
        - The plaintext phrase is "Crack codes seize flags". The ciphertext for "crack" is at position 5 on a keypad; the ciphertext for "codes" is at position 9 on a keypad, etc..

    - Flag:
        - CTF{5934}
- (Crypto) Passing Notes
    - Description:
        - My friend and I are in coding class, working on assignment. They sent me this file ciphertext.txt, which I think they created using our code. What does it say?
    - Solution:
        - The code is simply xoring the input and outputting it to a file. Thus, the reverse can be done where we xor the output to get the original input. Change the ‘plaintext.txt’ in the first line to be ‘ciphertext.txt’, and run the code.
        - You can run a powershell script on mac by downloading Homebrew and then ‘brew install powershell` and the command `pwsh assignment1.ps1’
    - Flag:
        - CTF{g1ve_m3_ur_c0d3!}

- (WEB) Hide n Seek: Hiding Monster
    - Hide n Seek is so fun! I swear, I just saw the monster around here somewhere … did he hide in this website somehow?
    - Flag: CTF{C00KIEM0NST3R}
        - Flag is in the cookies. Check local storage in your browser’s Inspect Element/Devtools
- (Pwn) Mother, May I?
    - Description:
        - Ask mother for the flag, nicely.
    - Solution:
        - Read the source code to see you just need to ask a question with ‘flag’ and ‘please’ before mother goes back to sleep. Can copy / paste ‘flag please’ when the program prompts you and you’ll get the flag
    - Flag:
        - CTF{m00oooo00000oooooommmmm}
- (CRYPTO) Four-Square
    - Hmmm… I was told there would be four square at recess, but instead I got this and a message. Since when do students do this instead of play outside?
    - NOTE: WRAP THE FLAG **IN ALL CAPS** IN CTF{}, i.e. CTF{FLAGHERE}
    - CIPHERTEXT: KLDUBSBEEBGTVNAOLGMIABMI
    - Solution: Look up “four square cipher” and use the dcode.fr solver
    - Flag: CTF{FOURSQUAREISCOOLWITHMATH}
- (WEB) Tic tac toe:
    - Description:
        - Can you win this game of tic-tac-toe?
    - Solution:

        - You can't win the game by playing. Intercept requests (with like Burp Suite) and on the move just before you win, change the placement of the responding O to somewhere else on the board so it doesn't block your win.
    - Flag:
        - CTF{t1c_t4c_f33t}
- (RE) 20 Questions
    - Description:
        - It's my favorite game - 20 questions!! Except the questions look a little bit funky…
    - Solution:
        - The file *is* stripped so I just opened it in Ghidra, clicked the entry function, and then clicked the function there to find the main function / code. From there, you can see that the obfuscated byte array flag is being XOR'ed with 0x29 to check if the user input matches the flag. To reverse it, copy the byte array and XOR it with 0x29 again to get the flag. Alternatively, you can brute-force the program
    - Flag:
        - CTF{20_w4s_t00_f3w!}
- (WEB) Hopscotch
    - Description:
        - Did you know I am actually The World's Best Hopscotcher™? No way you beat me in it! I'll even give you a prize if you do.
    - Solution:
        - The idea is that you get clues from one page to find the URL for the next page.
        - /: Inspect Element, the next page is a comment
        - /swishswoosh: Inspect Element or highlight, the next page is written in white font
        - /hippityhoppity: There's a hint in the page that the <p> tag is related to the flag. The next page is in the CSS, CTRL + F for super-secret, which is the class of the <p> tag
        - /potionofleaping: In the linked js page, there is a simple javascript function. Run it in the console or reverse-it to simply reverse the string and get the next page
        - /upintheclouds: There are 4 linked js, deobfuscate them by googling "deobfuscate javascript" and clicking on first link (https://deobfuscate.io/). Three js files are useless, the hop is in cumulus.js if you successfully deobfuscate and reverse
        - /wowholycowyouregoodatthis - Flag is on this page
    - Flag:
        - CTF{y0u_rule_th3_playgr0und_now}
- Rock paper scissor: (Medium)
    - Description:

        - Rock Paper Scissors: Deathmatch Edition!
        - Win 100 blind rounds of Rock-Paper-Scissors in a row against a mystery opponent. No ties—they reset your streak to zero. It's all or nothing. Think you've got what it takes?
    - Solution: Check solution script at the end of this document.
    - Flag:
        - CTF{rps_victory_1snt_gu4rant33d}
- (WEB) Simon Says
    - Description:
        - Simon asks for different encryptions/hashes/encodings to be done, Caesar, Vigenère, etc in a minute, progress resets if not completed in time
    - Solution:
        - Perform the following in under a minute (just a time trial)
            - Encrypt the plaintext you're given with Caesar shifted by the key you're given
                - https://www.dcode.fr/caesar-cipher
            - Encrypt the plaintext you're given with Vigenère using the key you're given
                - https://www.dcode.fr/vigenere-cipher
            - Hash the given plaintext with SHA1
                - https://www.dcode.fr/sha1-hash
            - Base64 encode the given plaintext
                - https://www.dcode.fr/base-64-encoding
            - Encrypt the plaintext you're given with Atbash
                - https://www.dcode.fr/atbash-cipher
    - Flag:
        - CTF{S1m0N_s4Yz_@_L0t}
- (RE) Hangman
    - Description:
        - I was a little too good at hangman, so I made a harder version.
    - Solution:
        - Annoying Go reversing binary which asks for a user to guess a 16 character alphanumeric string with no incorrect answers permitted. Decompilation looks gross because it is Go, but the challenge can be easily solved by patching the executable to always branch/jump to the win code.
    - Flag:
        - CTF{this_isnt_fun}

RPS SOLVESCRIPT

```
import subprocess
import time

EXECUTABLE_PATH = "./rps_ctf_challenge"

def rps_bot():
    win_streak = 0

    try:
        print(f"DEBUG: Using executable path: {EXECUTABLE_PATH}")

        process = subprocess.Popen(
            [EXECUTABLE_PATH],
            stdin=subprocess.PIPE,
            stdout=subprocess.PIPE,
            stderr=subprocess.PIPE,
            text=True
        )

        while win_streak < 100:

            print("DEBUG: Sending 'debug_mode' command.")
            process.stdin.write("debug\n")
            process.stdin.flush()

            opponent_hand = None
            for _ in range(10):
                game_output = process.stdout.readline().strip()
                print(f"DEBUG: {game_output}")  # Log game output
                if "DEBUG: Opponent's next hand is" in game_output:
                    try:
                        opponent_hand = game_output.split("is
")[1].strip("'")
                        print(f"DEBUG: Extracted opponent's hand:
{opponent_hand}")
                    except IndexError:
                        print(f"ERROR: Failed to parse opponent's hand
from: {game_output}")
                    break
                time.sleep(0.1)

            if not opponent_hand:
                print("Error: Opponent's hand not detected after
'debug_mode'.")
                break
            if opponent_hand == "rock":
                player_choice = "paper"
            elif opponent_hand == "paper":
                player_choice = "scissors"
            elif opponent_hand == "scissors":
                player_choice = "rock"
            else:
                print(f"Error: Unexpected opponent hand! Raw data:
{opponent_hand}")
                break

            print(f"Bot chooses: {player_choice}")

            process.stdin.write(player_choice + "\n")
```