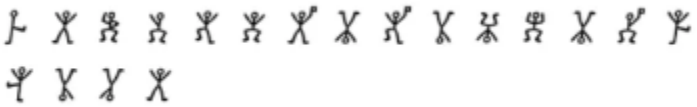
A series of mini CTF challenges created by other students at Northeastern!

- Challenge 1
    - “Yeehaw! You’re headed into town today because you’ve got a busy day ahead of you. The only problem is–where’s town? (Your flag will be the name of the city in the picture! Remember to wrap it with CTF{}, separate any spaces with underscores and put the name in all caps.)”
    - My Solution: Reverse Image Search and several peoples’ photos in that town will come
    - Image link
    - CTF{FORT_WORTH}
- Challenge 2
    - “You ride into town and see this sign. What does it say? Remember to wrap it with CTF{}, separate any spaces with underscores and put the text in all caps.”



-
    - I googled cipher with stick figures and the dancing men cipher appeared, then I went to dcode.fr and had to manually decipher it
    - CTF{WELCOME_TO_DUSTY_RIDGE}
- Challenge 3
    - “Your horse is tired and you want to buy them some food at the General store. You need to fill out this form to enter though…”
    - https://forms.gle/AXdyuCTW3MevtD4u8
    - I hit view page source of the form and quickly found the flag by command F CTF{
    - CTF{L3t_M3_1n}
- Challenge 4
    - “You walk into the general store. There is a carrot vending machine! How useful! However, money is tight for cowboys these days. You know of a special code that breaks the vending machine and gives the carrot for free! You take your notepad out and see that you wrote down the MD5 hash to the code which is “388fead5c64d1b4093bb74a3363fada1”, but you don’t remember the exact code itself… well, as far as you can remember, the code looks like CTF{CARROTX} where X is a number from 1 to 200. I.e. the real code looks like CTF{CARROT13} or CTF{CARROT194}, etc… If you hash all of those, eventually you’ll find a match! Maybe you can automate this…”

    - Solution: For this one I had to practice my python script writing, I never formally learned python so for this one I had to get the hint to use hashlib, from there i did a simple string comparison and brute forced each potential number.

```
import hashlib

def crack(target_hash) :
        for i in range(1, 201):
                code = f"CTF{{CARROT{i}}}"
                hashed = hashlib.md5(code.encode()).hexdigest()
                if hashed == target_hash:
                        return code
                else:
                        return None

traget_hash = "388fead5c64d1b4093bb74a3363fada1"
print crack(target_hash)
```

-
    - With this script I learned that code.encode essentially codes CTF{CARROT{i}} into a md5 hash. I also learned that you have to call hexdigest() to convert each byte of the hash to its two-digit hexadecimal representation. If you just do not call a digest, you will simply get a MD5 hash object, (tried this on my first scripting attempt and got <md5 _hashlib.HASH object @ 0x7f8b1c0b6f90>), simply doing .digest would return a hex format b'\x8f\xc3\xa3\xcd\xe7\x13\xd0\x70\x06\x98\x1f\xbe\xbb\xf0\xa3\x69', and hexdigest() gets a human readable string "8fc3a3cde713d070069818fbebbf0a369" which is the string we want.
- Challenge 5
    - "You bring your horse to water, but you can't seem to get them to drink! What's the problem?"
    - was given a why.docx file, I extracted with `binwalk -e why.docx`. When extracted it contains an unused .png asset which contains the flag easily visible.
    - CTF{they_only_drink_gatorade}
- Challenge 6 (HAS AN EASY AND HARD VERSION)
    - "Whoops! You were caught stealing and are taken to jail. Can you get out?"
    - Solution (EASY VERSION). I was given that there was a blocklist here.
    - I had to get guidance for this and was told that the blocklist was written in python. I did print(len(blocklist)) and found the blocklist was 16 items long. I popped each one (blocklist.pop(0 through 15)). From here I had to import os; os.system('sh')
    - I learned that this essentially invokes they system shell using Python's os module. Basically running the shell as a subprocess.

    - From here I used ls cd, and cat commands to find the flag.
    - CTF{p0p_0ut_0f_j41l}
- Challenge 7 (HAS EASY AND HARD VERSION, DIFF FLAGS)
    - “You enter the saloon. There’s some interesting music playing, and you record a snippet of it. What does it mean?”
    - Solution: This challenge has been used before in other CTF challenges I have done by these students in the past. So I knew to use https://morsecode.world/international/decoder/audio-decoder-adaptive.html to decode the morse code audio
    - CTF{C0TTON_EYE_JO3}
- Challenge 9
    - “A townsman slides a file your way. There’s probably something important if you read it…” Remember to wrap it with CTF{}, i received a zip file that had many other zip files inside of it.
    - Solution: manually unzipped the file 30 times, got a txt file at the end
    - CTF{Sh3riff’s_1n-t0wn_squ4r3}
-
- **Challenge 10 (HAS AN EASY AND HARD VERSION)**
    - “You head to the town square and see the Sheriff waiting for you, his hand on his pistol”, this challenge required me to type in bang after the shootoff begins but before the sherrif which was a 0.1 second window.
    - Wrote a script to connect, wait 3 seconds then send the input to the server. This helped me strengthen my TCP/UDP coding, I was able to become familiar with using socket in python through my Network Security Class.

```
import socket
import time

def exploit():
    HOST = 'challenges.neu-ctf.club'
    PORT = 1234

    # Create a socket object
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
        # Connect to the server
        s.connect((HOST, PORT))

        # Wait for 3 seconds
        time.sleep(3)

        # Send the "bang" input
```

```
        s.sendall(b'bang\\n')

        # Receive and print the response (flag)
        response = s.recv(1024)
        print(f"Received: {response.decode()}")

if __name__ == "__main__":
    exploit()
```

Reflection: Overall I really liked the range of challenges this GM offered. My favorite one was the blocklist, as I did not know you could execute a shell inside of a python script. CTF Club's Winter CTF in 2023 December was the first CTF I ever did, back then I did not even know how to use netcat to access a host. I also didn't know what a port was. It was fun to use this GM as a benchmark to see how far I have progressed since I first started this club.