Writeup for "Radio" from WelcomeCTF 2021 by NUS Greyhats

Miscellaneous

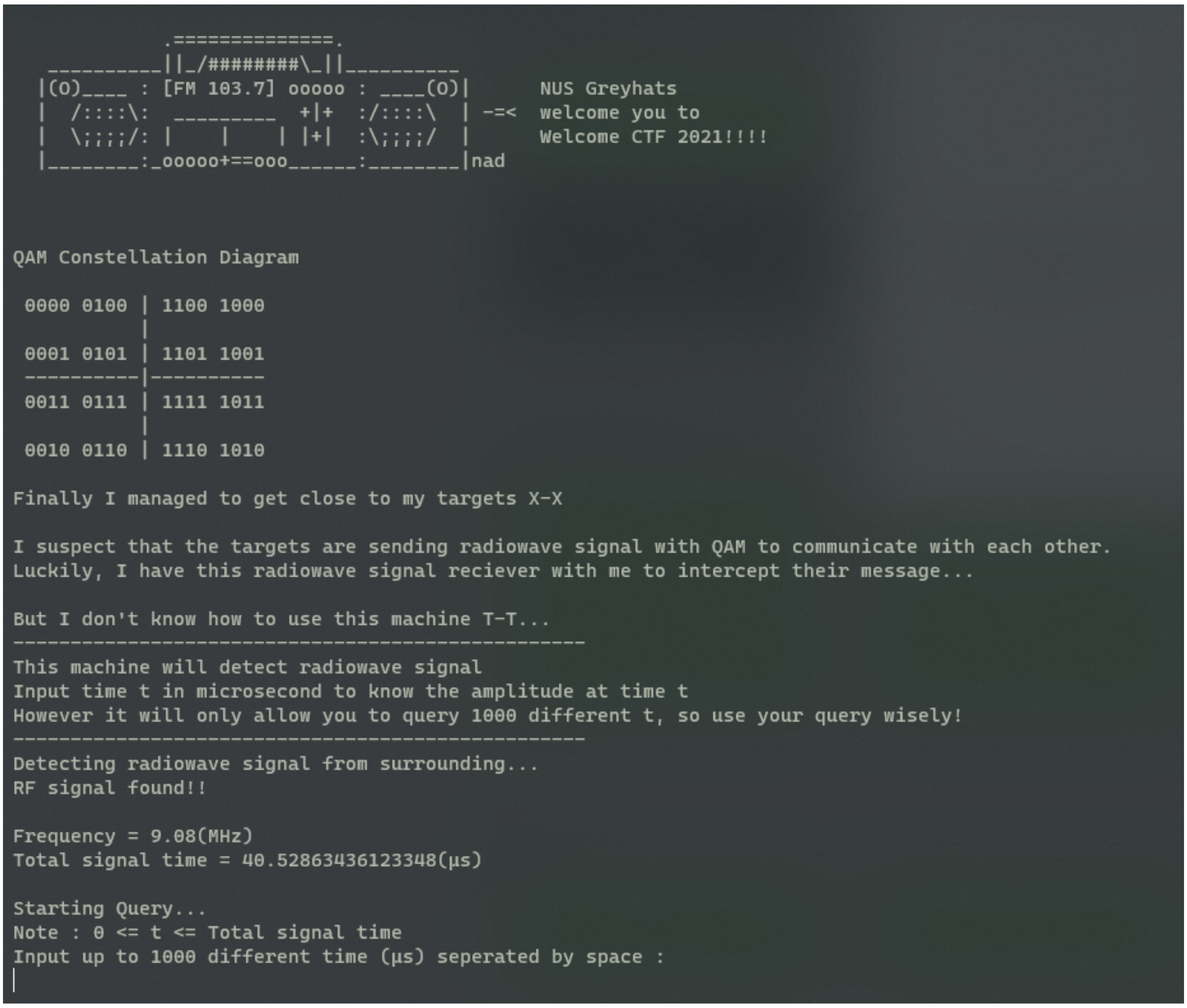
Radio

- mechfrog88

Bzzz...

> nc challs1.nusgreyhats.org 5213

An innocent looking challenge with not a lot of solves. (4 solves by the end of the ctf)



```
               .===============.
   ___________||_/#########\_||__________
  |(O)____ : [FM 103.7] ooooo : ____(O)|      NUS Greyhats
  |  /::::\:  __________  +|+  :/::::\  | -=<  welcome you to
  |  \;;;;/: |    |    | |+|  :\;;;;/  |      Welcome CTF 2021!!!!
  |_________:_ooooo+==ooo______:_________|nad



QAM Constellation Diagram

 0000 0100 | 1100 1000
           |
 0001 0101 | 1101 1001
 ----------|----------
 0011 0111 | 1111 1011
           |
 0010 0110 | 1110 1010

Finally I managed to get close to my targets X-X

I suspect that the targets are sending radiowave signal with QAM to communicate with each other.
Luckily, I have this radiowave signal reciever with me to intercept their message...

But I don't know how to use this machine T-T...
--------------------------------------------------
This machine will detect radiowave signal
Input time t in microsecond to know the amplitude at time t
However it will only allow you to query 1000 different t, so use your query wisely!
--------------------------------------------------
Detecting radiowave signal from surrounding...
RF signal found!!

Frequency = 9.08(MHz)
Total signal time = 40.52863436123348(μs)

Starting Query...
Note : 0 <= t <= Total signal time
Input up to 1000 different time (μs) seperated by space :
|
```

When you first netcat into the server, you will get the following response above.

What is this? QAM Constellation Diagram?!?!?

you can enter query in microseconds and it will return you the corresponding amplitude at based on the time you have entered and you can query up to 1000 times.

As this is a question under the Miscellaneous Section, the question basically test you on your general logic.

Doing a quick google search will tell you QAM is a kind of data to radio modulation techniques where data is encrypted into a radio wave and then transmitted out. And hence the title radio. Which means we need to extract out sample points from the question and demodulate the the coded script. Luckily for us the question already provided us with a QAM Constellation Diagram and the decrypted syntax is in bits.

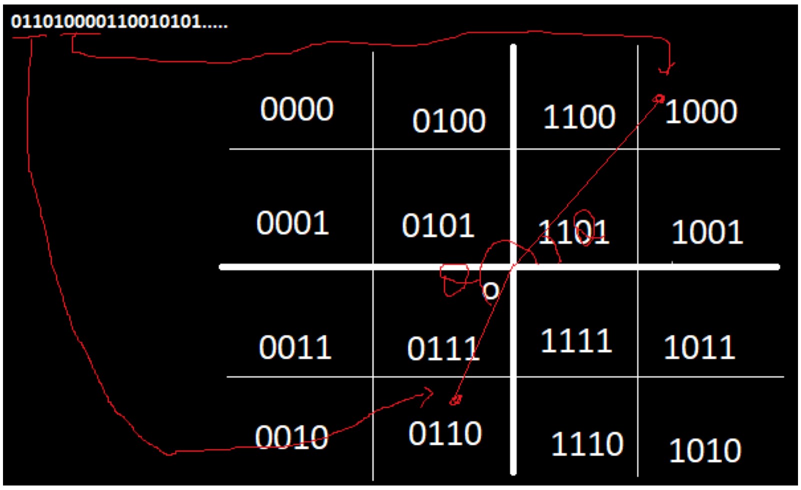
How Does QAM Work?

But how does QAM work you might ask? Welp I don't really know until I googled. As someone who is a freshman that is not from a math major, understanding how the process work took us quite a while. Essentially, data that you want to transmit is first represented into symbols available on the constellation diagram.

For example,

`"hello" -> 0110100001100101011011000110110001101111 // in binary`

and each of 4 bits in the binary is then mapped to the constellation provided



you should think of QAM diagram as a kind of graph with the origin at $(0,0)$ and every data mapped to the symbol should be thought of as a point within that region on the graph.

Now recall your A level Mathematics and every point (x,y) can be represented somewhat like as $(Acos(\theta), Asin(\theta))$, where θ is the angle the the point make with the x-axis and A is the length of the point to origin.

by converting the data to points represented by angle θ and A, we can then obtain a cosine wave function using:

$$signal = Acos(2\pi f \times t + \theta)$$

where f is the frequency, A is the amplitude of the wave and θ is the phase difference and t is the time parameter.

This allows the data to be transmitted in segments of radio waves for others to receive. The frequency is kept as a constant here as for an antenna can only be set to resonance with a specific frequency during usage. (not impt here).

Take note: in reality the 2 axis is regarded as real and imaginary axis and only the x-axis/real part will represent the Amplitude wave to origin. But I think this is easier to digest so I will use this here.

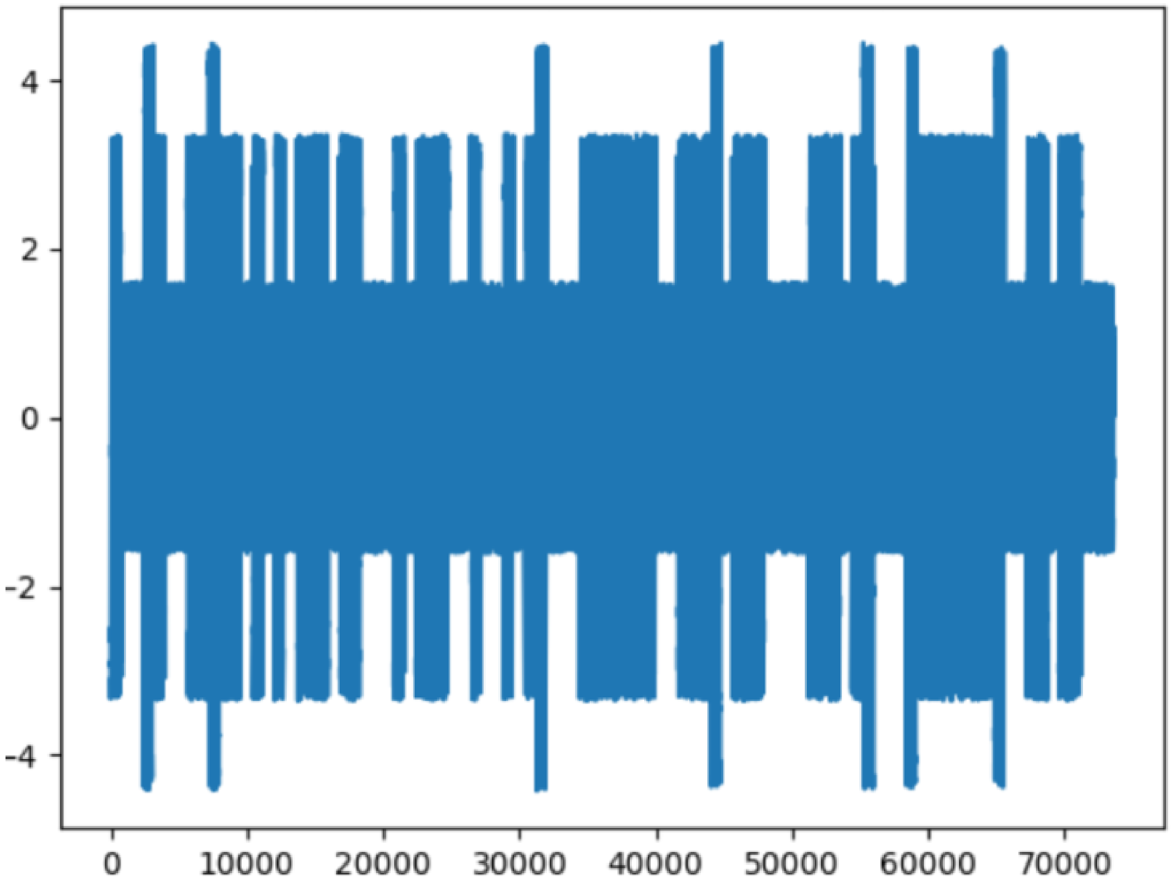
Steps Taken:

The challenge starts off with a netcat `nc challs1.nusgreyhats.org 5213`
Revealing this:
The instructions were as follows:

```
Finally I managed to get close to my targets X-X

I suspect that the targets are sending radiowave signal with QAM to communicate
with each other.
Luckily, I have this radiowave signal reciever with me to intercept their
message...

But I don't know how to use this machine T-T...
--------------------------------------------------
This machine will detect radiowave signal
Input time t in microsecond to know the amplitude at time t
However it will only allow you to query 1000 different t, so use your query
wisely!
--------------------------------------------------
Detecting radiowave signal from surrounding...
RF signal found!!

Frequency = 9.5(MHz)
Total signal time = 38.73684210526316(μs)

Starting Query...
Note : 0 <= t <= Total signal time
Input up to 1000 different time (μs) seperated by space :
```

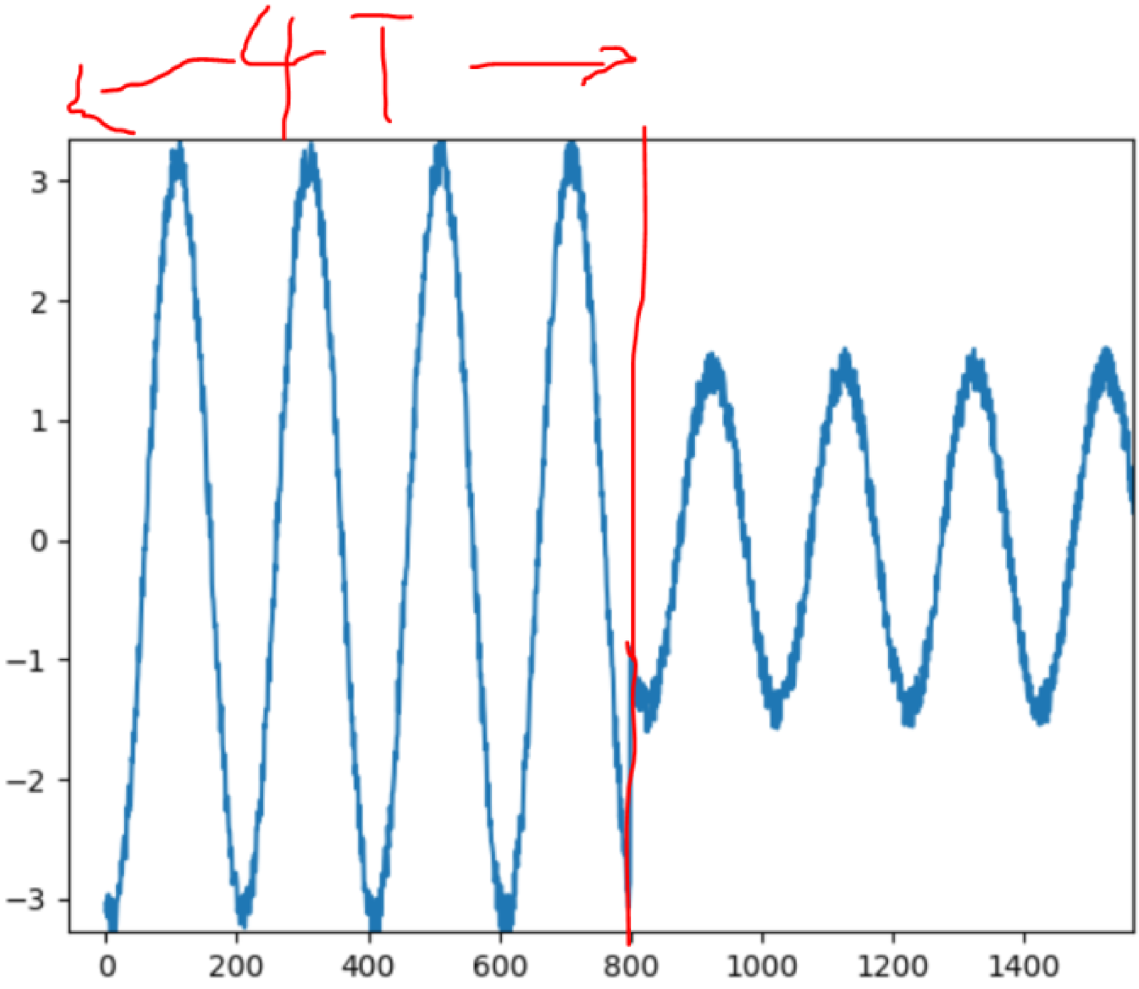
Entering time values (time in milliseconds or 10^{-6} seconds) produced the respective signal values at those times. Even though the frequency changes each time, the waveform is the same (the total signal time is effectively the same as well). We ended up using a Python script to automate obtaining all the signal values using 800 points per signal. (1000 query limit? No problem, just query again and adjust for the freq each time :P) The pwn library comes in handy here. Here is the code we used:

```python
def getData(blk_num):  # extract data in 76 chunks
    conn = remote("challs1.nusgreyhats.org", 5213)
    text = conn.recvuntil(b'seperated by space :', drop=True).decode('utf-8')
    text = text.split("\n")
    for line in text:

        if line.startswith("Frequency"):
            frequency = float(line.split("=")[1].strip()[:-5])
        elif line.startswith("Total signal time "):
            time_total = float(line.split("=")[1].strip()[:-4])

    blk = 1 / frequency * 5 # This was originally 20
    interval = blk / 1000
    timestamp = []
    for i in range(1000):
        timestamp.append(blk_num * blk + i * interval)
    timestamp = [k for k in timestamp if k < time_total]
    queryString = " ".join([f"{k}" for k in timestamp])
    queryString += "\n"
    conn.send(bytes(queryString, "utf-8"))
    text = conn.recvuntil(b'~~', drop=True).decode('utf-8')
    conn.close()
    return timestamp, [float(d) for d in text.split("\n")[2].split()]
```



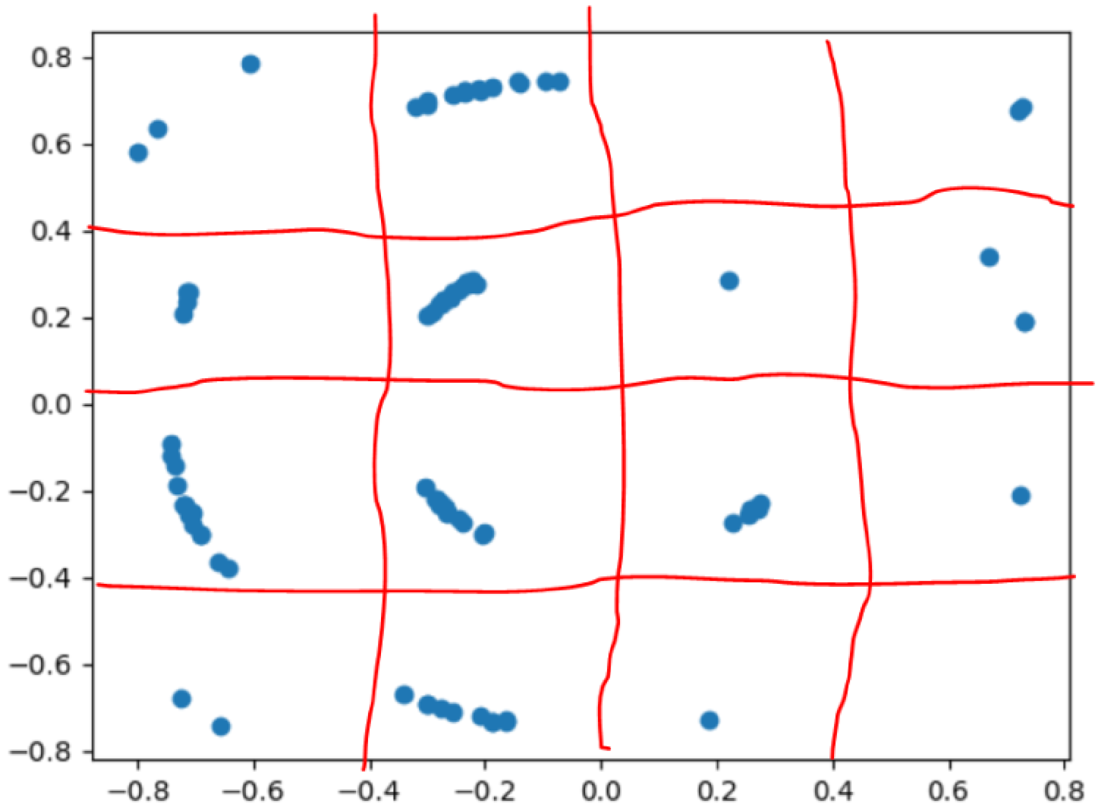
After digging around QAM and how it worked, we now knew that to decipher the signals, we only needed

1. Amplitude. The signals had 2 amplitudes at ~3 and ~1 respectively. Simply get the `max(signal)`.
2. Phase offset. By finding the first peak and measuring the distance from the start (each signal was 800 points containing 4 waves, so the phase is simply $\frac{\text{index of the first peak point}}{200} * 2\pi$.



While the idea was fine, in practice there was some **mix up across signals.**
The peak point/amplitude occasionally measured that of the previous/next signals (ie we **should not assume** that points 0, 800, 1600... are the start of a new signal/end of the current signal). Since each signal has 4 waves, we simply shaved off the first and last periods (200 points of data each). In doing so, we ensure that the signal we are measuring is indeed the correct signal and not the next or previous one.

So all that's left is to plot the points in the quadrature graph. It was this graph that motivated us to increase the number of points queried in an attempt to make them look 'nicer' ie fit the 16-QAM diagram better. (though a better would most definitely be denoising the graph but we got lazy :P) After all the steps above (including the optimisation), here is how it looked like:



It looked slightly suspicious and we didn't know if the data was going to be alright. We implemented a simple QAM mapping, following the diagram provided in the challenge. After obtaining the binary data, we translated that into ASCII, and got this:
7r?|8>{vt¼ûoõ?koüõgjõöf1»õ»°ûkõ÷ng8õò>»njwks?}
We nearly got a heart attack. But with the curly braces there, *surely* there was something we did right. After looking up the ascii codes, we found that '7', was possibly replaced by 'g'. By some stroke of luck, the curly braces were correct! So we simply reflected the graph in the line $y = x$ and tadah! Flag obtained.

```
greyhats{IT5_e45Y_70_S3nD_D@T4_W17h_RaD10w4ve}
```

Credit:

Team: SUN egelloC lairepmI
Doomkiller for carry
SLICC for moral support