11a) Naughty/Nice List with Blockchain Investigation Part 1

Difficulty: 5/5

Difficulty

Even though the chunk of the blockchain that you have ends with block 129996, can you predict the nonce for block 130000? Talk to Tangle Coalbox in the Speaker UNpreparedness Room for tips on prediction and Tinsel Upatree for more tips and tools(https://download.holidayhackchallenge.com/2020/OfficialNaughtyNiceBlockchainEducationPack.z
(Enter just the 16-character hex value of the nonce)

ANSWER: 57066318f32f729d

Access

Naughty list

The blockchain naughty list `blockchain.dat` file can download Santa's office.

Block chain file

Pseudo random number predictor

There is a tool from Tom Liston to help predict pseudo-random numbers using MT19937 at https://github.com/tliston/mt19937 An easier to addpt tool can be downloaded from https://github.com/kmyk/mersenne-twister-predictor.git

Extracting the nonces of the block

The function `load_a_block` of the provided python script can be modified to print the nonces of all blocks on the chain.

Modified naugthy_nice.py script

```
def load_a_block(self, fh):
    self.index = int(fh.read(16), 16)
    print (self.index) #MLR
    self.nonce = int(fh.read(16), 16)
    #print([self.nonce, self.index]) #MLR
    print(str('%016.016x' % (self.nonce))) #MLR
```

We save the extracted nonces to `nonceshex.txt`.

Nonces

Predictor script

The script will recreate a random state with the same parameters as the one that generated the nonces, and will feed them in order so that it recreates the state and then we call our custom random number generator to generate the next four nonces.

```
predictor = MT19937Predictor()
    with open('nonceshex.txt') as nonces:
        for nonce in nonces.readlines():
            knownnonce=int(nonce,16)
            predictor.setrandbits(knownnonce, 64)
    for i in range(4):
        print(str('%016.016x' % (predictor.getrandbits(64))))
```

Predicting script

Predicting the next four nonces

Once the nonces are stored in the `nonceshex.txt` file we run the `mt19937predictorBlockchain11a.py` predict the following values on the sequence.

```
b744baba65ed6fce
01866abd00f13aed
844f6b07bd9403e4
57066318f32f729d
```

ANSWER: 57066318f32f729d