

Original Reddit Post

https://www.reddit.com/r/Bitcoin/comments/9kq7it/introducing_the_310_btc_bitcoin_challenge/

Prize Wallets

310btc (solved) RIP :'(

<https://blockexplorer.com/address/39uAUwEFDi5bBbdBm5ViD8sxDBBrz7SUP4>

0.31btc (unsolved)

<https://blockexplorer.com/address/3NPZiNWid7cCfXZa1D8tnEZBPgQ884cVw7>

0.2btc (solved)

<https://blockexplorer.com/address/1G7qsUy5x9bUd1pRfhVZ7cuB5cMUP4hsfR>

0.1btc (solved)

<https://blockexplorer.com/address/1446C8HqMtvWtEgu1JnjwLcPESSruhzhkmV>

0.1btc Solution

The grid at the bottom is hex characters

511	B20	332	328	410	530
22B	0FE	52E	D0F	7A1	65B
52C	7E7	511	2F6	56F	C4B

Above that on the image is OCT 2 2018 in small print.



Using 20181002 as a shift key, shift all the original hex values to new value as shown below.

Original Hex	511	B20	332	328	410	530
shift	201	810	022	018	100	220
decrypted hex	310	310	310	310	310	310
Original Hex	22B	0FE	52E	D0F	7A1	65B
shift	181	002	201	810	022	018
decrypted text	1AA	0FC	32D	5FF	78F	643
Original Hex	52C	7E7	511	2F6	56F	C4B
shift	100	220	181	002	201	810
decrypted text	42C	5C7	490	2F4	36E	43B

The first 310 repeating line is an indicator telling you you did it right so can be stripped, which leaves you with 12 sets of hex numbers, which when converted to decimal are all less than 2048.

Hex

The will give you the groupings of L3, 02, 7, 584, and 9F. No logic has been discerned but there are only 1920 combos of different groupings that string this together. In this case the correct string is L3+7+9F+485+02=**L379F48502**.

Now using salted string 1 from the alpha channel (details below), use this as a password to decrypt the string

```
U2FsdGVkX19Q3I//VCH0U3cVtITZ3ckILJnUcdPX3Gs5qjdF1UjZ3mAftGivtFYD
N5ZCSkBynnVqBaw14p8wK0008zI6D0A1+VEVCUyEvEeNoUfGcS0E19d93vsPxbg7
D5avufQsScgsk3QEtd9/M4Do320KFeq00/3NrxW0sMmh3AXmDzuuZ0qmZaI7re16
FcXIrmPPiQD0HRc7wt0ng6qLiNz7VqESRTdxP0ahKFRkWT8sT+Ur2y+2iZ2LEaxN
M7UZqcPwYgm6FoKOVjndeg30R27jc6AoFPyRZ2g8+EJMp3n/pf94oSCLEWkc0os
jH9DqbM6DUptu3HJbAVwXQ==
```

Which gives the following output

Bitcoin Challenge 310 BTC
<https://bitcoinchallenge.codes/>

Well done!
Now find something really interesting here:

```
511 B20 332 328 410 530
245 651 58F C2C 03A 717
401 9AC 36A 53F 4C6 B26
332 328 410 530 491 312
```

310 BTC

This gives a new grid similar to the 0.1btc solution (as indicated by the first row) and can be solved via the same solution.

511	B20	332	328	410	530
245	651	58F	C2C	03A	717
401	9AC	36A	53F	4C6	B26
332	328	410	530	491	312

Using 20181002 as a shift key, shift all the original hex values to new value as shown below.

Original Hex	511	B20	332	328	410	530
--------------	-----	-----	-----	-----	-----	-----

shift	201	810	022	018	100	220
decrypted hex	310	310	310	310	310	310
Original Hex	245	651	58F	C2C	03A	717
shift	181	002	201	810	022	018
decrypted text	1C4	65F	38E	41C	018	70F
Original Hex	401	9AC	36A	53F	4C6	B26
shift	100	220	181	002	201	810
decrypted text	301	78C	2E9	53D	2C5	316
Original Hex	332	328	410	530	491	312
shift	022	018	100	220	181	002
decrypted text	310	310	310	310	310	310

The first 310 repeating line and the last 310 repeating line is an indicator telling you you did it right so can be stripped, which leaves you with 12 sets of hex numbers, which when converted to decimal are all less than 2048 again.

Hex

1C4 65F 38E 41C 018 70F 301 78C 2E9 53D 2C5 316

To Decimal

452 1631 910 1052 24 1807 769 1932 745 1341 709 790

Which you can map to bip39 seed words for the wallet using 1-2048 index

<https://github.com/bitcoin/bips/blob/master/bip-0039/english.txt>

debris slim immune lock actual tide gas vapor fringe pole flat glance

This seed doesn't unlock the wallet and is an invalid mnemonic phrase, but after concating the new words with the words from the first wallet, you get a new 24 word seed that is valid and unlocks the second wallet.

**cry buyer grain save vault sign lyrics rhythm music fury horror mansion debris
slim immune lock actual tide gas vapor fringe pole flat glance**

Registration Hash

With both the salted strings decrypted, you can now submit a successful registration hash at <https://bitcoinchallenge.codes/register-310/>. The proper submission is to concat seed 1 + alpha salt decrypt + gzip salt decrypt with new lines removed as shown below.

sha256(cry buyer grain save vault sign lyrics rhythm music fury horror mansionBitcoin
Challenge 310 BTChttps://bitcoinchallenge.codes/---Well done!Now find something
really interesting here:511 B20 332 328 410 530245 651 58F C2C 03A 717401 9AC
36A 53F 4C6 B26332 328 410 530 491 312---310 BTCBitcoin Challenge 310
BTChttps://bitcoinchallenge.codes/---You're either very very close, or working in the
wrong direction :)Here you go : Z465/---

273e2b95648fd3cbad0d7fe3ed820e783c0b12fdb29b57bfb2d1f243d92b1a5

Red Channel String Decryption

Using the same process as before and the groupings of L3, 02, 7, 584, and 9F, a second sequence can be ordered to find **02L3F95847** which can be used as the decryption for the salted string obtained from the gzip.

Using salted string 2 from the red channel gzip (details below), use this as a password to decrypt the string

U2FsdGVkX1+WPMJQISUVUvGRg7p4zCX4jIODIGb6b6cAreXFxv0W0xgCeSw9K+im
THiWMkRq45FsPXHS3TjYqcJz7QzQ8HeM340EwWQWXAi0fVy+r6NPmiJRgMgMqLCu
4Q9o/WkNyHxvPScNgG9jf8gskggx10FiTcoyF1KE+nxjmRkEuj7uQQsPr1RP3sj
1l4KXhAzrGQZi5E4sajQ0BGQfaJjei5fHXX06sxeYsFcuxzo3JdM0F3JFYQtuUDY

Which gives the following output

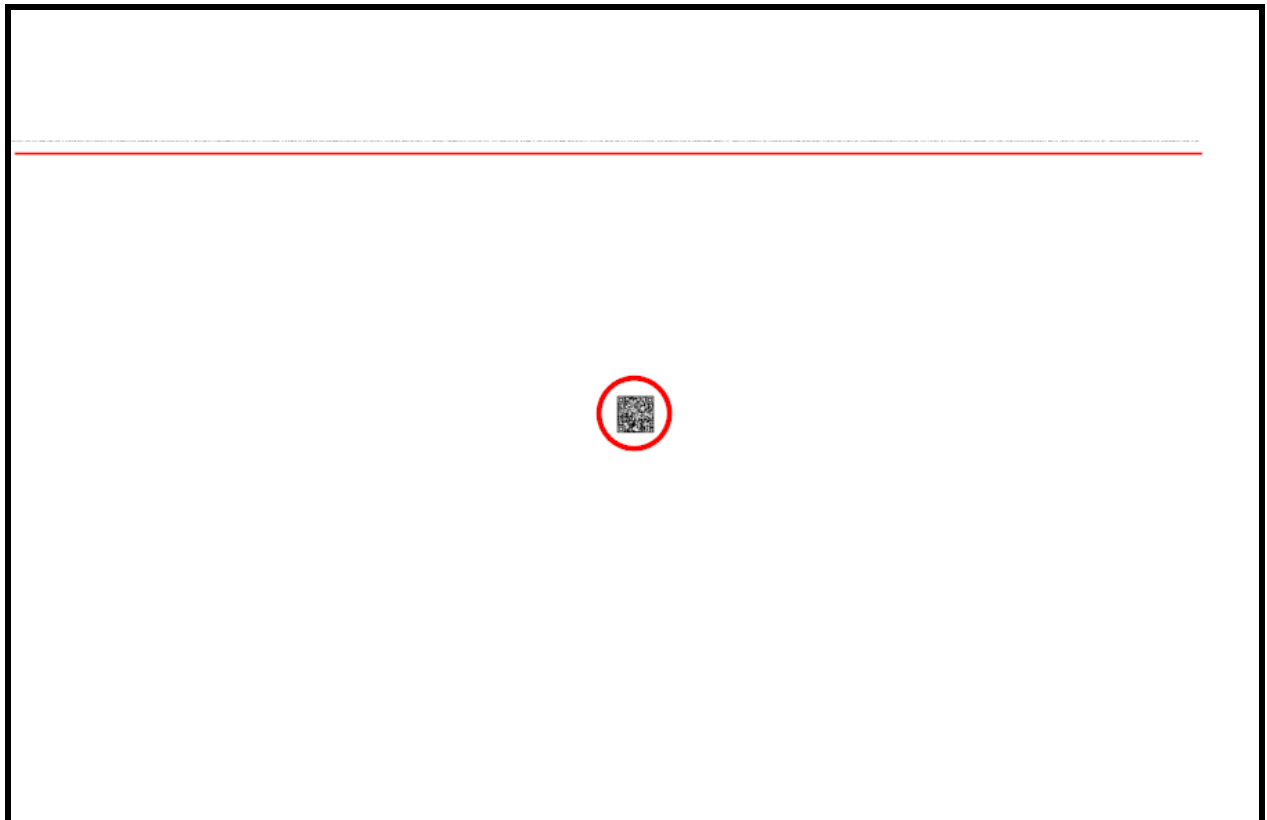
Bitcoin Challenge 310 BTC
<https://bitcoinchallenge.codes/>

You're either very very close, or working in the wrong direction :)

Here you go : Z465/

Alpha Channel Extraction for Salted String 1

Retrieve the alpha channel color code from the image via any chosen method (python, gimp, photoshop). Almost all pixels will have a value of 255 minus a few areas. A row of pixels on line 310 and 2800 pixels long has some modified values along with a square in the middle of the image.



The square in the middle of the image is a QR and when scanned directs you to <https://bitcoinchallenge.codes/register-310/>



The remaining line at the top is a binary sequence. The first 2800 pixels will be either alpha value 253 or 254, pixels 2801+ are 255 and can be discarded. Convert 253=1 and 254=0 to get a binary string.

```
01010101001100100100011001110011011001000100011101010100110101101011000001100010011100101010001001100110100101111001011110
101011001000011010010000011000001010101011001101100011010101100111010001001001010100011001101100011011010110100100101
001100010010011011100101010110001101100100010100000101100000110011010001110111001100110101110001011010100110010001000110001
100010101010101101010010110100011001101101010000010110011001110100010001110110100101110110011101000100011001011001010001000000
1010010011100011010101101001000011010100110101101000010011110010110111001101100101100111000101000010011000010111011101101
10000110100011100000111000011011101001011010011110011000010011110011100001111010010010010110110010001000011000001000001001100
0100101011010101001000101010110010000110101010111100101000101011101100100010101100101010011100110111010101011001100100011
1011000110101001100110000010001010110110000110010110010000111001001100110110110011001101010000011100001100010011001100110111
00001010010001000011010101100001011101100111010101100110010100010110011010100110110011101100110110101100110011010100010
1000101011101001110001001110010010111010011010001000100111100110011001100100100111101001101000110011001010111000100
1100000011000000101111001100110100111001100111000010111010011110111001101001101011010101000001100110100000101011000011
0110101000100011110100111010101101010110100011000001110001011011010110100110000101001001001101110010011001100010011
01100000101001000110011000110101100001001001110010011011010101000001010000101001010001010001000111010010000101001001100
01100110111011101110110100001100000110111001100110011011001100011000110100110100111010011010011011101001000101001001100
0101010011010100101010001100100011110000101000001001111011000010110100001001101000110010100110101101010111010101000011100
00111001101010000101011010101011100100011001001111001001010110011001001101000110010011000110010001010110000101111000
0100111000001010010011010011011101010101010111000101100011010100000111011101010010110011101101001101100100011001101110
100101101001111101010110011010100110111001100010110010011001011001100110000010100100011001101101010100110001100
11011001000001011011110100011001010000011100101010010011010001100100110011100000101011010001010100100110101110000001
10011010111000101110111000001100110001110010011010001101111010100110100001101001100010001010101110110101101100011001100000110
1111011100110000101001101010010000011001010001000111000101100010010011010011011001000100010101011100000111010001110100110
0110100100001001001100010010000010101011001110110101100001010001001111010011110100001010
```

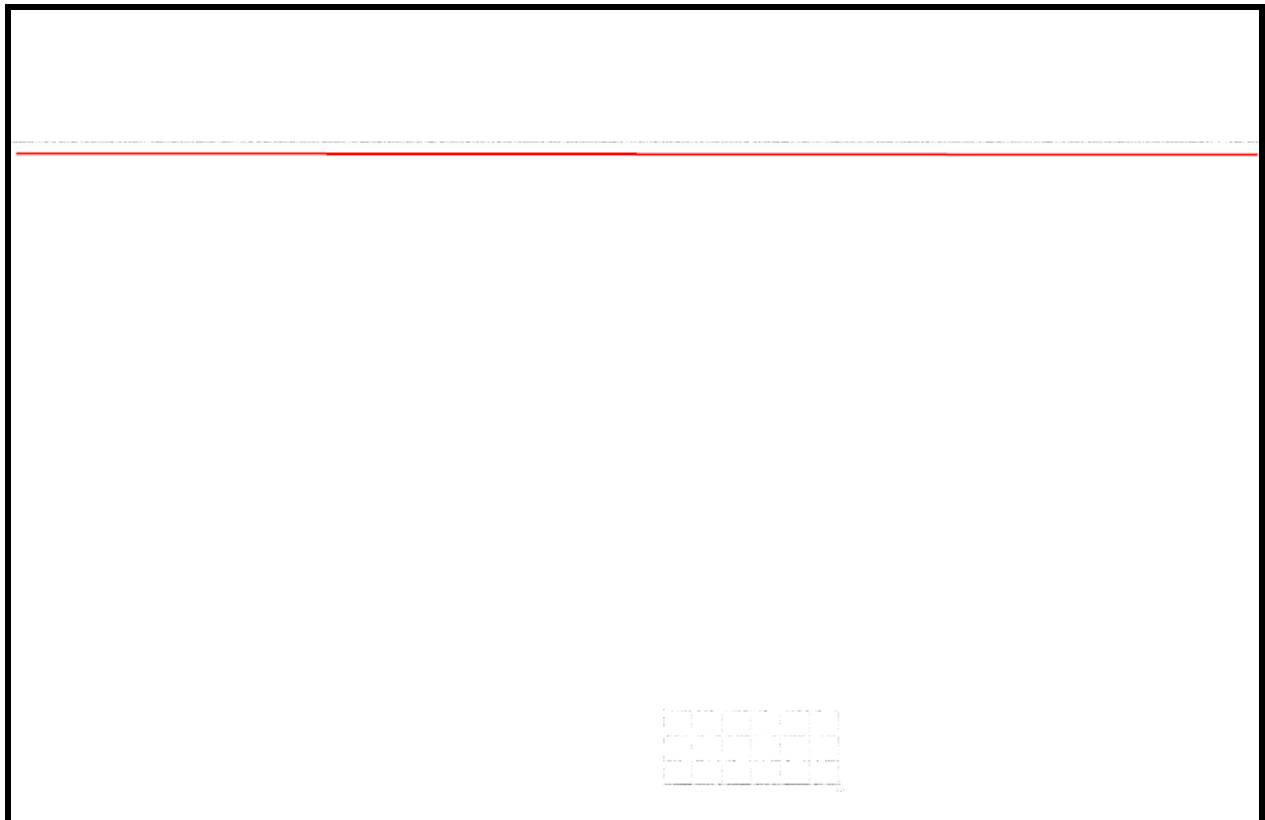
and then convert this to ASCII to get the text below.

```
U2FsdGVkX19Q3I//VCH0U3cVtITZ3ckILJnUcdPX3Gs5qjdF1UjZ3mAftGivtFYD
N5ZCSkBynnVqBaw14p8wK0008zI6D0A1+VEVCUyEvEeNoUfGcS0E19d93vsPxbg7
D5avufQsScgsk3QEtq9/M4Do320KFeq00/3Nrxw0sMmh3AXmDzuuZ0qmZaI7re16
FcXIrmPPiQDOHrc7wt0ng6qLiNz7VqESRTdxPOahKFRkWT8sT+Ur2y+2iZ2LEaxN
M7UZqcPwYgm6FoK0Vjnnqdeg30R27jc6AoFPyRZ2g8+EJMp3n/pf94oSCLEWkc0os
jH9DqbM6DUptu3HJbAVwXQ==
```

The start of this string is "U2FsdGVkX1" which converts to "Salted__" in b64 to ASCII which indicates openssl was used to make it.

Red Channel Extraction for Salted String 2

Reviewing the grey bits channel of the image shows that not every pixel follows normal grayscale of R=G=B. In a few cases, specifically on line 310 where alpha was changed we can see R=GB is not valid and spans the full 2944 pixels of the image.



This time you will extract the LSB (least significant bit) of the Red channel for row 310 via any preferred method. This will give you the binary string below.

```
101101010100011010110001100001001000010011110011000110101100111110100111100110110000010111011110011000111101111101000011010101010100001101011101010110
001011100110101100110010011100011110010011000011101001110110101000110111011001000100000011011110101100000001001100000100000111100000101101000101011100
11001110111110111111011001100010100000111000010100011101001110100110000100001101110010001011000110101010011000111000110000001111101100111011100111000
011000101000101011001010110001000101001100010100010001110001010100000011100101100110111001000010110001011110111001000110101000011111101001001001010011
0011000000100101011001010010101001000000000000110011011110000100111101111101111001010011100000010100100100100110011110001000101001001001001001001001001
0110111110101011000100010110001001100011100011001010111000100011000011111100011100110100001001000011110011110111001000100100010101000011111010100
011010010100111010111011001111100001000011010111101110101101000000111000101001101111010000000011010110011100101101011111000001111101011000111011000100
1100011111001011100000101111001010010000011100110101101100010111110010000111100101110111011101001011010001001001010001010001010001010000101
100100011000111110001100001111110110100100011010100110010001000100011010101000110011011100011101011111000101100011011000101011001001011011010000011111
101000110010000010010101010001000011011001100001110000110011101010010110000101111001110001100001001010101111010010010101100001000001011101111001101
11100010100010011010010000100100000010011011010000101100100011100111011100001011111000110111011010101001000110111001100000010101110001110000000111100
1000100001110000001010111101011110011000000100111111011001001001110000110101110100100101001011110100100000111001011100101110001100110001100
```

You will already notice you are onto something as all the bits after the 2800th pixel are zeros which is when the alpha channel binary ends so you can remove those. XOR these 2800 bits with the 2800 bits from the alpha channel to get the binary below.

Again you will notice that after pixel 1960ish all the values become the same again indicating something is going on. Flip the bits (swap 1 and 0) to make the trailing string all zeros and convert the string to hex (you can also remove extra trailing zeros). You will notice it starts with "1F 8B 08" which is the header for a gzip file.

Convert this string to b64 and then decompress the file however you want. (online works too <http://www.txtwizard.net/compression>)

H4slCB9Ls1sAA0RBVEEABcHJYkQwAADQu19xKBWMoxqx1RJr9DZjSBNUSc2Er+971TvKD68esSo3W
RyioKjq6unlxPwFp4MBC9Jr4N2Nu9HZW4+heCpNKOjTFy8rkukslT5t4jFfgQ55hn2ulaxdu/A00Ykufh9rQH
FfDWqwTZWhPuTNSLKZhmJSbx+OrsEkLW8NWNy+OKZFV1CPIsNF8JHqoSqQFp2ywHVyJV/BJvz0d
2ZuSPes22b8kztJomEOFv+9w89EV1F/AbQ+mHh4ZbyHqqDz7GqcFF33LY7eJctPARp1ALYyv+9uraSv
8UWVO5BAEAAA==

U2FsdGVkX1+WPMJQISUVUvGRg7p4zCX4jIODIGb6b6cAreXFxv0W0xgCeSw9K+im
THiWMkRq45FsPXHS3TjYqcJz7QzQ8HeM340EwwQWXAi0fVy+r6NPmiJRgMgMqLCu

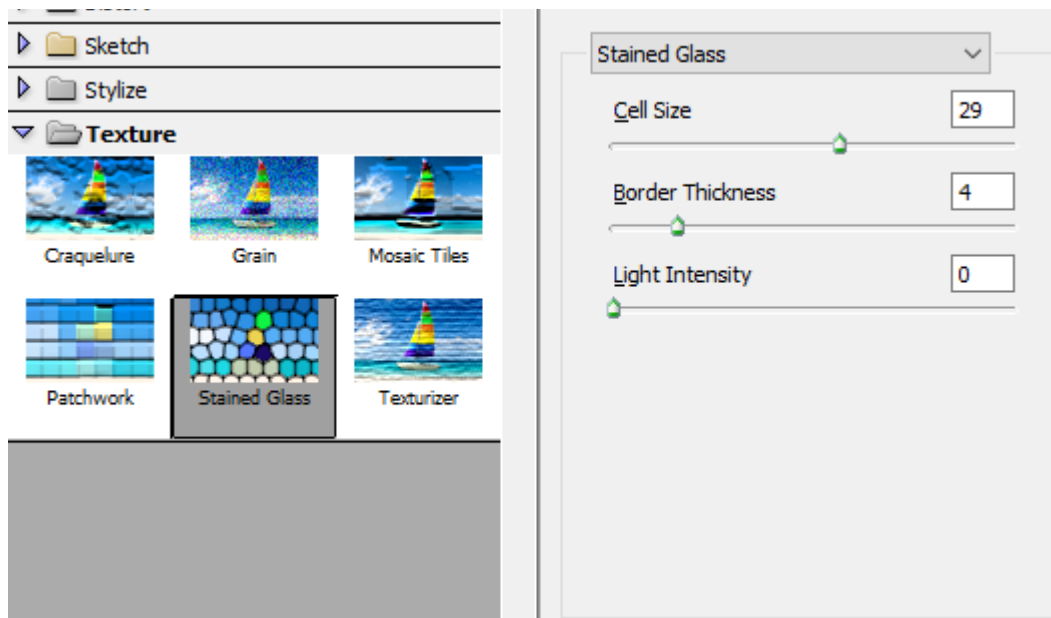
4Q9o/WkNyHxvPScNgG9jf8gskggx10FiTcoyF1KE+nxjmRkEuj7uQQsPrrlRP3sj
1l4KXhAzrGQZi5E4sajQ0BGQfaJjei5fHXX06sxeYsFcuxzo3JdM0F3JFYQtuUDY

The start of this string is "U2FsdGVkX1" which converts to "Salted__" in b64 to ASCII which indicates openssl was used to make it.

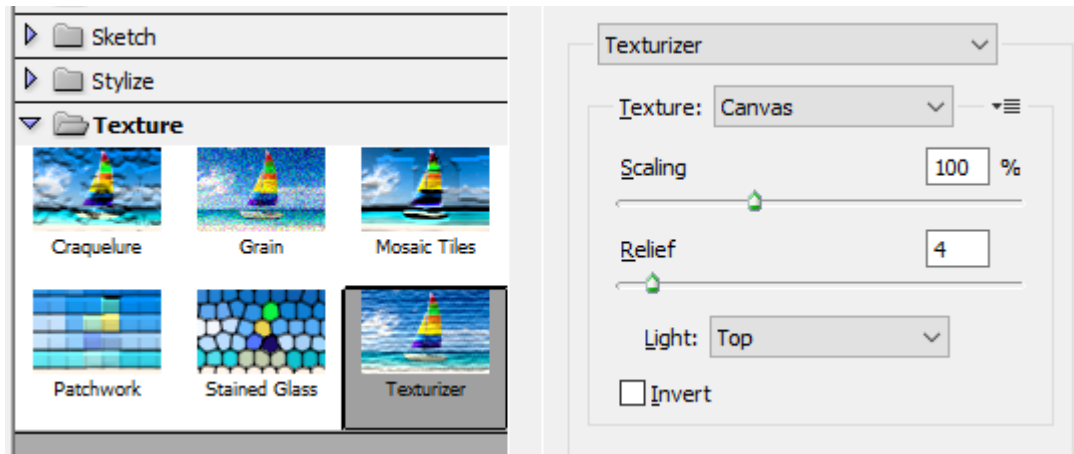
Original Image

The original image found to have generated this puzzle is located below. It has been converted to grayscale and had the following photoshop filters applied.

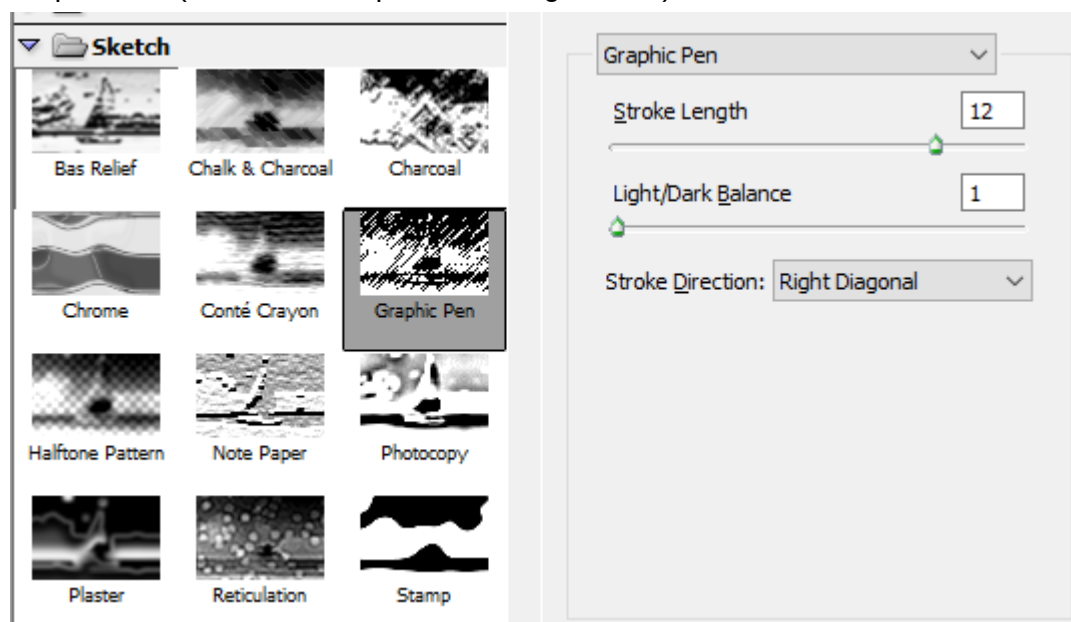
Stained Glass (Close, but appearance varies drastically based off slight changes in white/black distribution of base image)



Texture (Exact settings below)



Graphic Pen (Close, but not perfect settings match)



<https://www.goodfreephotos.com/albums/other-landscapes/light-shining-through-the-trees-in-the-forest.jpg>

