

Document maintained by JTobcat  
Join us solving at Neon District Discord <https://discord.gg/aH69kbJ>

## **Original Reddit Post**

[https://www.reddit.com/r/Bitcoin/comments/9kq7it/introducing\\_the\\_310\\_btc\\_bitcoin\\_challenge/](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/1446C8HqMtvWtEgu1JnjwLcPESSruhkmV>

---

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

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

1AA 0FC 32D 5FF 78F 643 42C 5C7 490 2F4 36E 43B

To Decimal

426 252 813 1535 1935 1603 1068 1479 1168 756 878 1083

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

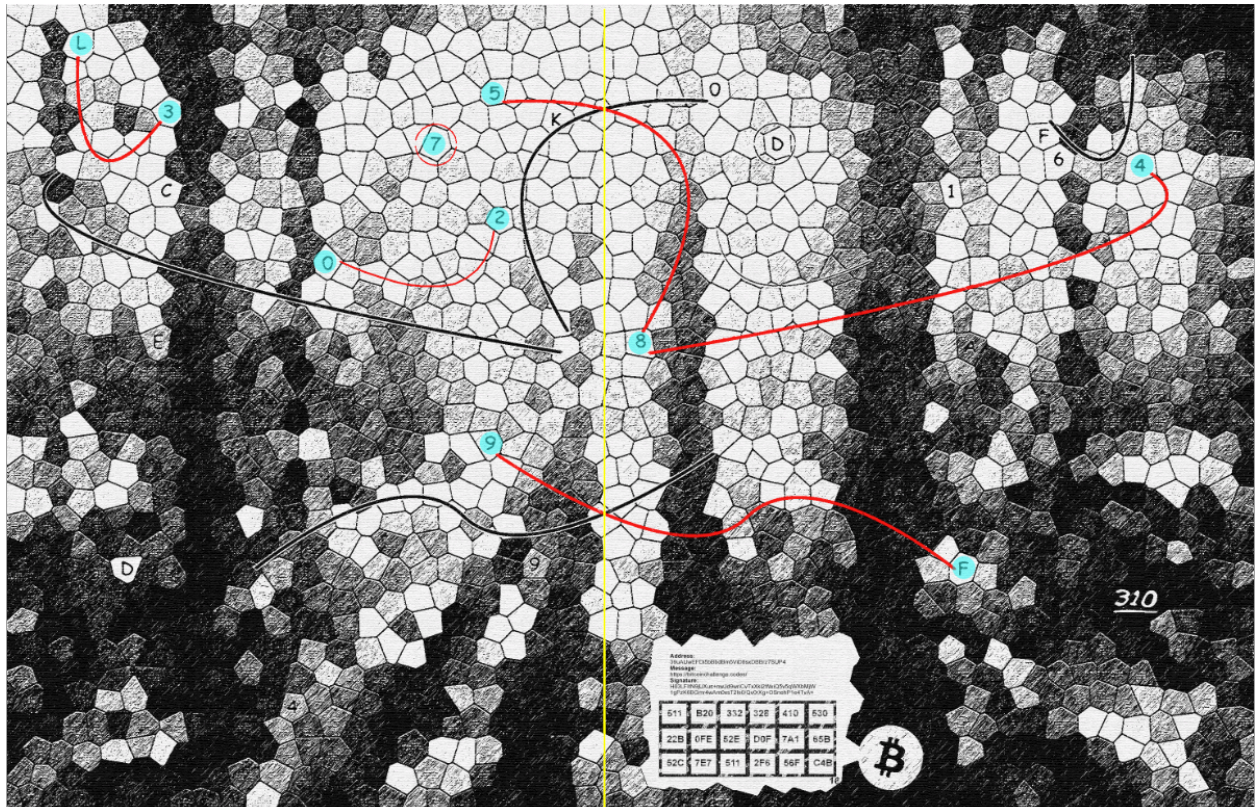
<https://github.com/bitcoin/bips/blob/master/bip-0039/english.txt>

**cry buyer grain save vault sign lyrics rhythm music fury horror mansion**

---

## 0.2btc Solution

Mirroring the 6 curves about the vertical axis  $x=1412$  connects them all to 10 characters.



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
N5ZCSkBynnVqBawl4p8wKO008zI6D0A1+VEVCUyEvEeNoUfGcS0El9d93vsPxbg7
D5avufQsScgsk3QEtq9/M4Do32OKFeq00/3NrxW0sMmh3AXmDzUUZ0qmZaI7re16
FcXIrmPPiQDOHrc7wt0ng6qLiNz7VqESRTdxPOahKFRkWT8sT+Ur2y+2iZ2LEaxN
M7UZqcPwYgm6FoKOVjndeg30R27jc6AoFPyRZ2g8+EJMp3n/pf94oSCLewk0os
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.

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

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

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 concatenating 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/---)

**273e2b95648fd3cbad0d7fe3ed820e783c0b12fdbe29b57bfb2d1f243d92b1a5**

---

## 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

```
U2FsGVkX1+WPMJQISUVUvGRg7p4zCX4jIODIGb6b6cAreXFxv0W0xgCeSw9K+im
THiWMkRq45FsPXHs3TjYqcJz7QzQ8HeM340EwWQWXAi0fVy+r6NPmiJRgMgMqLCu
4Q9o/WkNyHxvPScNgG9jf8gskggx10FiTcoyF1KE+nxjmRkEuj7uQQsPr1RP3sj
114KXhAzrGQZi5E4sajQOBGQfaJjei5fHXXO6sxeYsFcuxzo3JdMOF3JFYQtuUDY
```

Which gives the following output

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

---

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

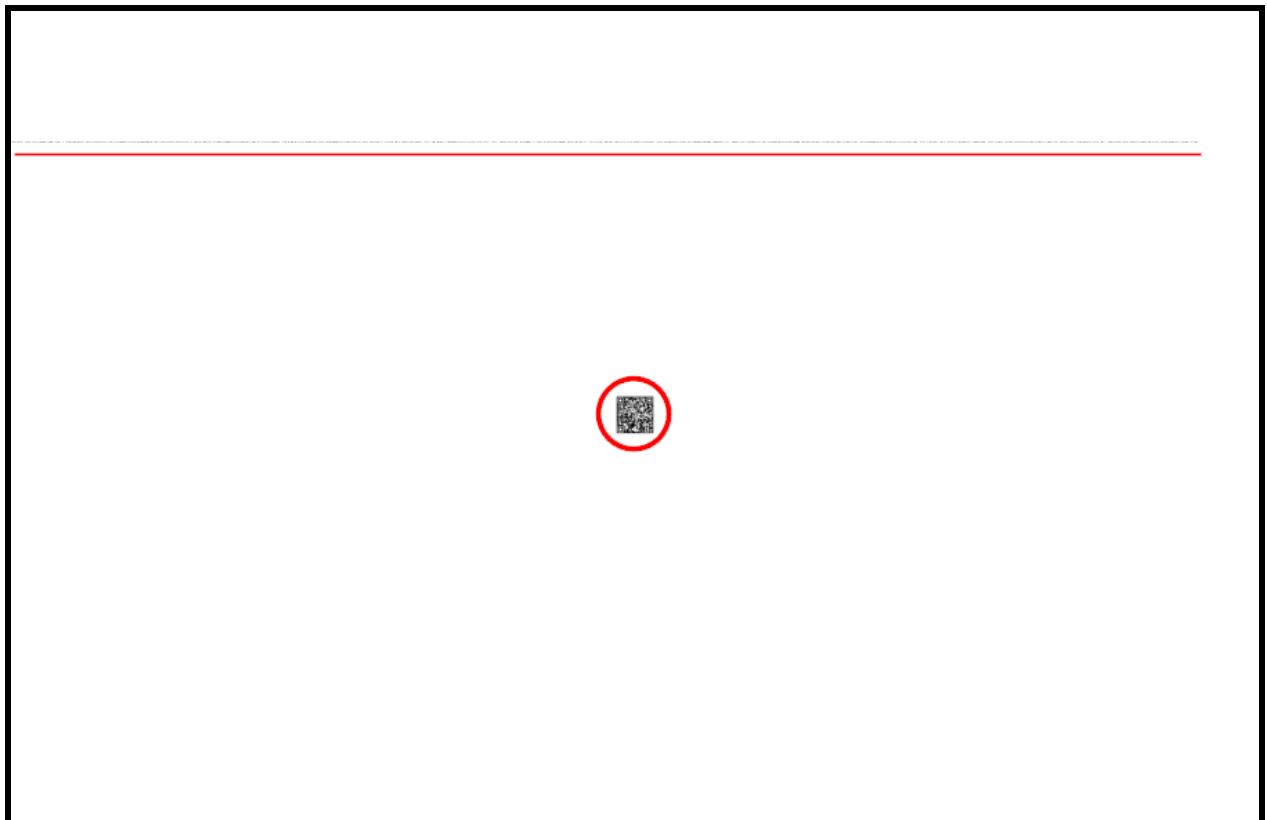
Here you go : Z465/

---

-----

## 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.

```
010101010011001001000110011100110110010001000111010101100110101101011000001100010011001010100010011001101001001011100101110
10101100100001101001000001100000101010100110011011000110101011000100100101010001011010001100110110001101010110100100101
0011000100101001101110010101011000110110010001010000010110000010011010001101110011001101010110001011010100110010001000110001
1000101010101011010100110101000110011011010101000001011001100111010001000111011010010111011001110100010001001011001010001000000
10100100111000110101010110100100001101010011010101101000010011110010101110011011100110110011011001100110001010001001100001011101101101
10000110100011100000111000011101110100101101001111001100000100111100111000011110100100100100110110010001000011000001000001001100
0100101011010101100100010101011001000011010101011100101000101011101100100010101100101010011100110111010101011001100100011
1011000110101001100110000010001010110110000110010110000011001001000110011011011001100110101000001110000110001001100110111
0000101001000100001101010110000101110110011101010110011001010001011100110101001101100011011001101100110110101100110011010100010
1000101011101000111000100111001001011101001101001101000100010011011100110011001100100100111101001101101000110011001010111000100
11000000110000001011110011001101001110011001001110000101011101001111011001101001101011010110101101000001100110100000101011000011
01101010001000111101001110101011101010110100011000001110001011011010110100110000010100100100110111011001001100101001100010011
0110000010100100011001100011010110000100100101110010011011010101000001010000010100010100010001001111010010000101001001100
011001101110111011101101000011000001101110011001100110110011000101001100011010011010011100111010001101110101011001110001010001
010101001101010010010101000110010001111000010100000100111101100001011010000100101101000110010100100110101101010111010101000011100
001110011010101000010101101010101110010001100100111100100101011001100100101010100011001001001100010001010110000101111000
010011100000101001001101001101110101010101101001110001011000101010000011011101011001011001101101011001101100100011001101110
10010110100111101010110011010100110111001110001011001000110010101100111001100110011000001010010001100100011011101010100110001100
11011001000001011011101000110010100000111001010100100101100001100100110011100110000010101100001010101001010010101110000001
1001101011100010111011100000110011000111001001101000110111101010011010000110100110001000101010101110101011010101100011001100000110
11110110011000010100110101001001000001110010100010001110001011000100100110100110110010001000101010111000000111010001110100111010100110
011010010000100101001100010010000010101011001110110101100001010001001111010011110100001010
```

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

```
U2FsdGVkX19Q3I//VCH0U3cVtITZ3ckILJnUcdPX3Gs5qjdf1UjZ3mAftGivtFYD
N5ZCSkBynnVqBawl4p8wK008zI6D0A1+VEVCUYEvEeNoUfGcS0El9d93vsPxbg7
D5avufQsScgsk3QEtq9/M4Do32OKFeq00/3NrxwOsMmh3AXmDzUUZ0qmZaI7re16
FcXIrmPPiQDOHRc7wt0ng6qLiNz7VqESRTdxPOahKFRkWT8sT+Ur2y+2iZ2LEaxN
M7UZqcPwYgm6FoKOVjndeg30R27jc6AoFPyRZ2g8+EJMp3n/pf94oSCLewkC0os
jH9DqbM6DUptu3HJbAVwXQ==
```

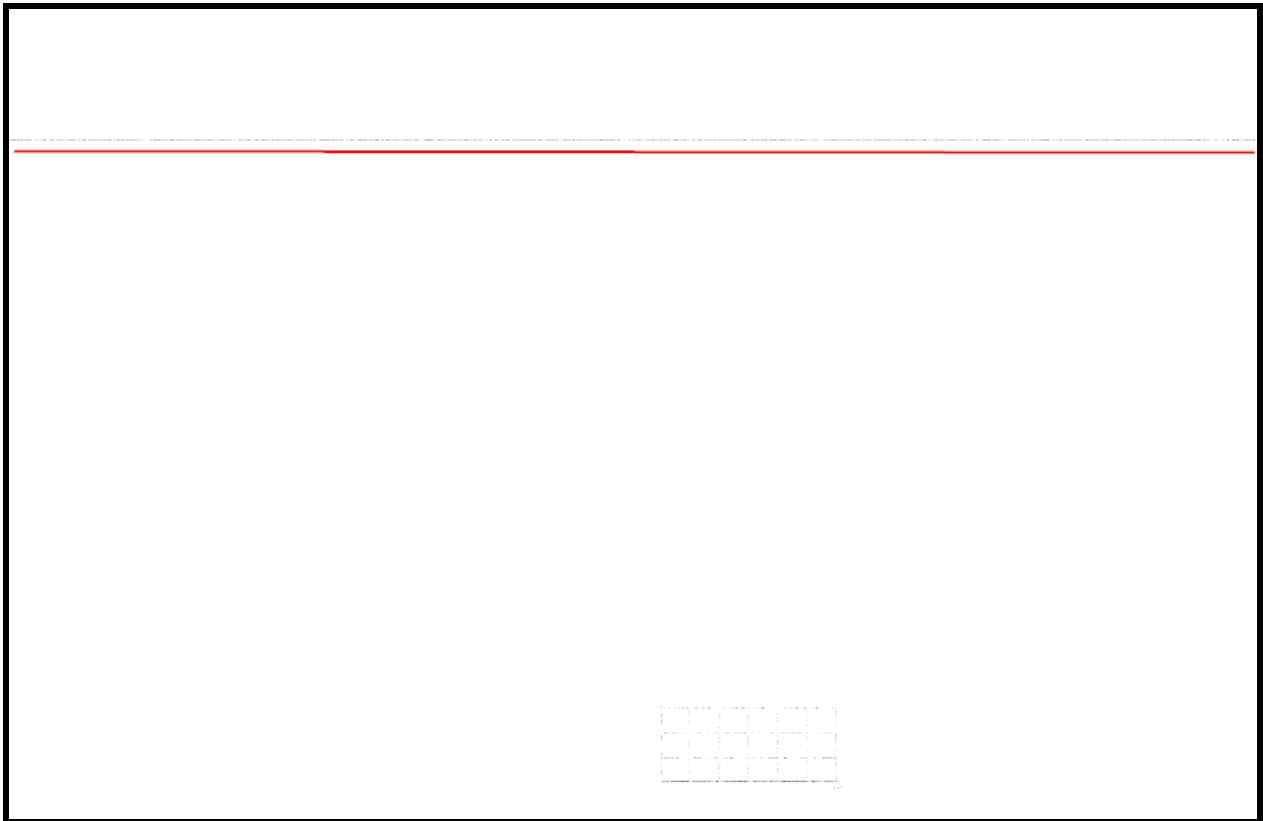
The start of this string is "U2FsdGVkX1" which converts to "Salted\_\_" in b64 to ASCII which indicates openssl was used to make it.



---

## 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.

```
1011010101000110101100011000010010000100111100110001101011001111010011110011011000001011101111001100011110111101000011010101011010000110101110101011
0001011100110101100110010011100011110010011000011101001110110101000110111011001000100000110111101011000000010011000001000001111000001011010001010110111
001100111011110111111011001100010100000111000010100011101001110100110000100001101110010001011100010101010001100011000000111101100110111100111
0000110001010001010110010101100010001010011000101000100011100010101000001110011001100110010000101100010111011100100010101000011111010010010010101
001100110000001001010110010100101010010000000000001100110111000010011110111101110010100110000001010010010010011001110001000101001001001010010
010010110111101010110001000101100010011000111000110010101110001000110000111110001110011010000100100001111001110111001000100100010101000011111
01010001101001010011101011101100111100001000011010111011101011010000011100010100110111101000000011010110100111001011010111100000111110101100011101
10001001100011111001011100000101111001010010000011100110101101100010111110010000011110000011100110111011101101001011010001001000101000100100010
100001011001000110001111100011000011111101101001000110101001100100010001000110101010001100110111000111010111110001011000110101111100010110001101110100
00001111101000110010000010010101010001000011011001100001110000110011101010010110000101111001110001100010010101111010010010101100001000001011110
11110011011110001010001001101001000010010000001001101101000010110010001110011101110000101111100010111101101010100100011011100110000001010111000111000
0000011110010001000011100000010101111010111001100000100111111101100100101101001011101001000001110010111001011110000011001101
```



4Q9o/WkNyHxvPScNgG9jf8gskggx10FiTcoyF1KE+nxjmRkEuj7uQQsPr1RP3sj  
114KXhAzrGQZi5E4sajQOBGQfaJjei5fHXXO6sxeYsFcuxzo3JdMOF3JFYQtuUDY

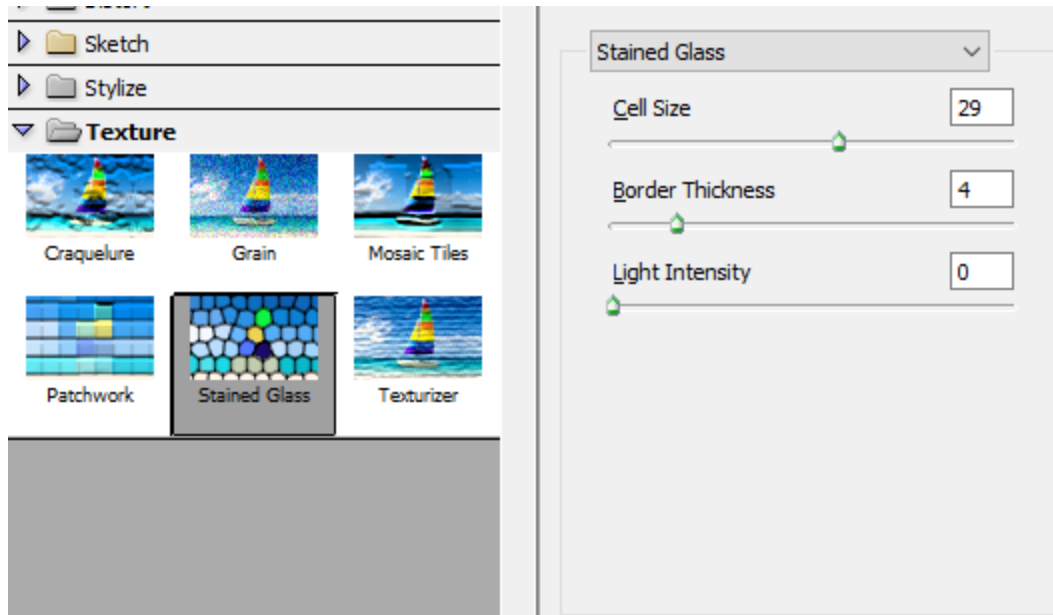
The start of this string is "U2FsdGVkX1" which converts to "Salted\_\_" in b64 to ASCII which indicates openssl was used to make it.



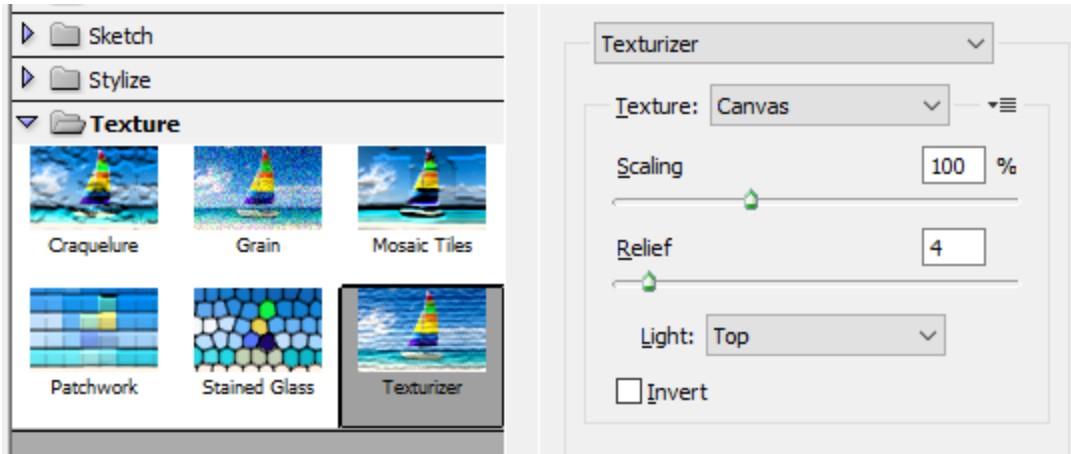
## Original Image

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.

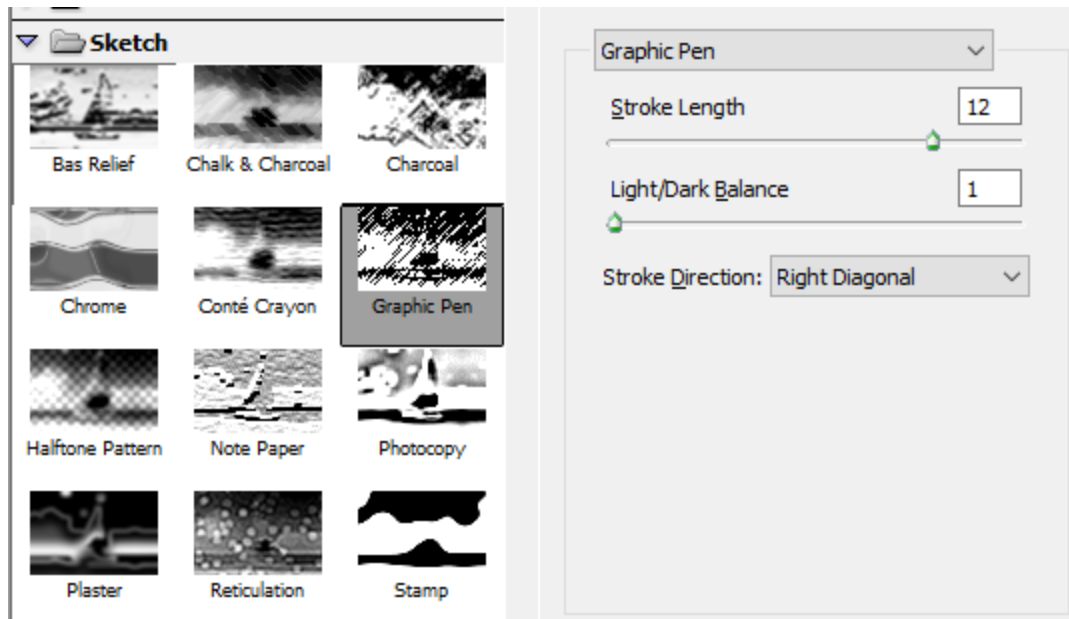
Stained Glass (Close, but appearance varies drastically based off slight changes in white/black distribution of base image)



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>

