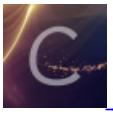
Python3对Base64隐写解密

原创

Cony_14 于 2019-10-24 16:19:28 发布 1041 收藏 1
分类专栏： Python 文章标签： Base64




```
import base64


def get_base64_diff_value(s1, s2):
    """get base64 diff value"""
    base64chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
    for i in range(len(s2)):
        if s1[i] != s2[i]:
            return abs(base64chars.index(s1[i]) - base64chars.index(s2[i]))
    return 0

def solve_stego(stego_str_list):
    '''stego_str_list str列表'''
    bin_str = ''
    for stego_str in stego_str_list:
        stego_str = stego_str.replace('\n', '')
        norm_str = base64.b64encode(base64.b64decode(stego_str)).decode()
        diff = get_base64_diff_value(stego_str, norm_str)
        pads_num = stego_str.count('=')
        bin_str += bin(diff)[2:].zfill(pads_num * 2)


    ret = b''
    for i in range(0, len(bin_str), 8):
        ret += int(bin_str[i:i + 8], 2).to_bytes(1, 'little')
    return ret

def to_stego(normal_data, stego_data, stego_bit_len=4):
    base64chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
    bin_str = ''
    for each in stego_data:
        bin_str += bin(each)[2:].zfill(8)
    line_len = len(normal_data) // (len(bin_str) // stego_bit_len)
    if stego_bit_len == 4:
        line_len = line_len - ((line_len % 3 - 1) + 3) % 3 # 令 line_len % 3 = 1 右边建立了一个 0 1 2 -> 2 0
    else:
        line_len = line_len - ((line_len % 3 - 2) + 3) % 3 # 令 line_len % 3 = 2 右边建立了一个 0 1 2 -> 1 2
    if line_len <= 0:
        return []
    ret = []
    index = 0
    while True:
        if index >= len(normal_data):
```

```
            break
        encode_str = base64.b64encode(normal_data[index:index + line_len]).decode()
        # print('index=%d bin_str=%s' % (index, bin_str))
        index += line_len
        # print(encode_str)
        if bin_str != '':
            if stego_bit_len == 4:
                now_encode = bin_str[:4]
                bin_str = bin_str[4:]
            tmp_list = encode_str.rpartition(encode_str[-1 * (stego_bit_len // 2 + 1)])
            encode_str = tmp_list[0] + base64chars[base64chars.index(tmp_list[1]) + int(now_encode, 2)] + t
        ret.append(encode_str)
    return ret


def main():
    with open('2.txt', 'rb') as fp:
        file_lines = fp.readlines()
    stego_str_list = []
    for each in file_lines:
        stego_str_list.append(each.decode().replace('\n', ''))
    print(solve_stego(stego_str_list))

    l = to_stego(b'12345678901234567890123456789012345678', b'cnss', 4)
    print(l)
    print(solve_stego(l))

if __name__ == '__main__':
    main()
```

隐写原理参照：

https://www.tr0y.wang/2017/06/14/Base64steg/