https://github.com/awedxz/hgame2018-.git
这里就简略的说一下，具体的可以看每一题对应的文件夹下带`solve`字样的文件

easy rsa

phi = (p-1)*(q-1) = p*q-(p+q)+1

The same simple RSA

看`solve.py`

xasr

分析一下脚本我们可以知道msg中每一位的加密都是独立的 针对msg中每一位的加密过程如下
假设x是msg中的第一位 那么x的加密顺序为 `h}**********{emag`

```
hgame{**************************}hgame{**********************
*******}
 |<--到此为止                    <--x
```

也就是说msg中的每一位只是和不同顺序的flag进行计算 而且计算结果互不干扰
然后我们可以发现这种加密有一个特点（以 `msg = 'a' * 32` 举例）

```
msg = "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
c1 = "bd5595fd55f57d5d356d059dbda5fd250da525a55d95b5b58d3dddddd5ed85ed"
我们可以知道flag的第二位是g 我们对a进行一次加密可以得到
hex(((ord('g')^ord('a'))+ord('g'))%256) == 0x6d 即'm'
我们再对 msg = m + 'a' * 31 进行加密得到
c2 = "995595fd55f57d5d356d059dbda5fd250da525a55d95b5b58d3dddddd5ed85ed"
取c1的第二位与'g'进行一次加密
chr(((ord('g')^0x55)+ord('g'))%256) == '\x99'
我们可以发现我们计算出来的值恰好等于c2的第一位的值
所以可以总结出这样的规律 每一位msg与相应位上的flag先加密一次再代入前一位进行完整的加密的结果和先进行完整的加密再进行一次独立的加密的结果是一样的
```

我们可以依靠这个规律来编写爆破脚本 具体参见`xarsSolve.py`

Caesar&&Caesar

维吉尼亚有在线网站可解 不过这里我确实为难了一下校内的同学
https://www.guballa.de/vigenere-solver
http://www.mygeocachingprofile.com/codebreaker.vigenerecipher.aspx
具体怎么字频分析 鸽了
~~去年分析到凌晨三点，不想再受一次折磨了~~

violence

一共就`12*26 == 312`种可能 全部爆破一遍 然后根据描述找出唯一一个有意义的句子
看312遍就出来了 当然这样会很痛苦
分析一下flag格式可以看出 有一位是一个字母的 在句子中一个字母的可能性大概只有`a, i`然后稍微过滤一下就没几种可能性了

Random

一个非常简单的反序列化的漏洞 唯一难的地方在于可能很多新生不知道`&` 详细构造方式看`solve.php`

这里再说一下非预期（感谢MitAh师傅告知）

一开始没有检查是否是对象 所以可以传入`s:1:"a";` 然后$emmm就变成了字符串 $emmm->public = xxxxx$会报warning然后`emmm->public`变成NULL了 `secret`同理 `NULL==NULL` 成功绕过