WEEK 4

ezECC

约等于送分吧 就是了解一下椭圆曲线 用sage很快就能做出来了 代码不长直接贴出来了

```
p = 1026347883361447
a = 499590297305427
b = 30115568120981
E = EllipticCurve(GF(p), [0, 0, 0, a, b])
G = E([817367249716330, 483834901818242])
k = 622849
pub = k * G
print pub
# pub (513848964032483 : 886359250407321 : 1)
# flag = hgame{1400208214439804}
```

CBC V0.3

套路和0.2真的是一模一样 同样是利用unpad的不合理

这次由于一定要是 `salt+'admin'` 才能拿到flag 所以这里要利用unpad的不合理去爆破salt

首先获取salt的长度 利用unpad我们可以截取不同长度的salt+ciphertext

通过逐渐缩小长度我们会遇到d41d8cd98f00b204e9800998ecf8427e不能绕过sig校验的情况

这个时候我们就截取到了salt的第一位 同时也就可以推算出salt的长度了

然后就是爆破salt了 因为我们可以利用unpad做到逐位截取salt 而每一位salt至多只有0x00-0xff种情况

只要通过了md5校验就是正确的 所以我们只要尝试至多256次就能获取salt的第一位 以此类推就能拿到salt了

为了减少复杂度这题salt只有9位 `ert#$%678` 也就是因为这个产生了非预期

题目没有限制空字符串 输入空串我们就能拿到salt的md5值 因为太短了所以可以直接在线查出来

至于检测ciphertext16位后的限制 直接注册两份就好了

解题脚本 鸽了

HellRSA

这题先从提示来说吧 观察 `13*12` 和`5*4` 二进制低位变化

```
13*12 --> 0b1101*0b1100 --> 0b10011100
 5* 4 -->  0b101* 0b100 -->    0b10100
```

我们可以发现5、4转化成二进制就是13、12二进制值的最低三位 同时它们乘积的最后三位是相同的

多换几组数据我们会发现结果是一样的 再尝试一下我们就能发现 这个规律能应用到几乎所有的基本计算中

也就是说 这题也符合这种规律 那么现在就明朗了 我们可以利用这个规律逐位爆破出p和q

具体如何操作可以看HellRSASolve.py

奇怪的SQLi

简单的浏览一下站点 可以发现是一个非常典型的ssrf利用 因为这题的需要我给出了源代码

扫一下就能发现 `.git` 目录 这里稍微多了点花样 我把config.php藏在了之前的版本中 所以要回溯一下

结合题目和ssrf 可以去百度或者谷歌搜搜ssrf攻击mysql 随便看看应该能看到这篇文章

接下来模仿文章里的操作就可以了