Week2

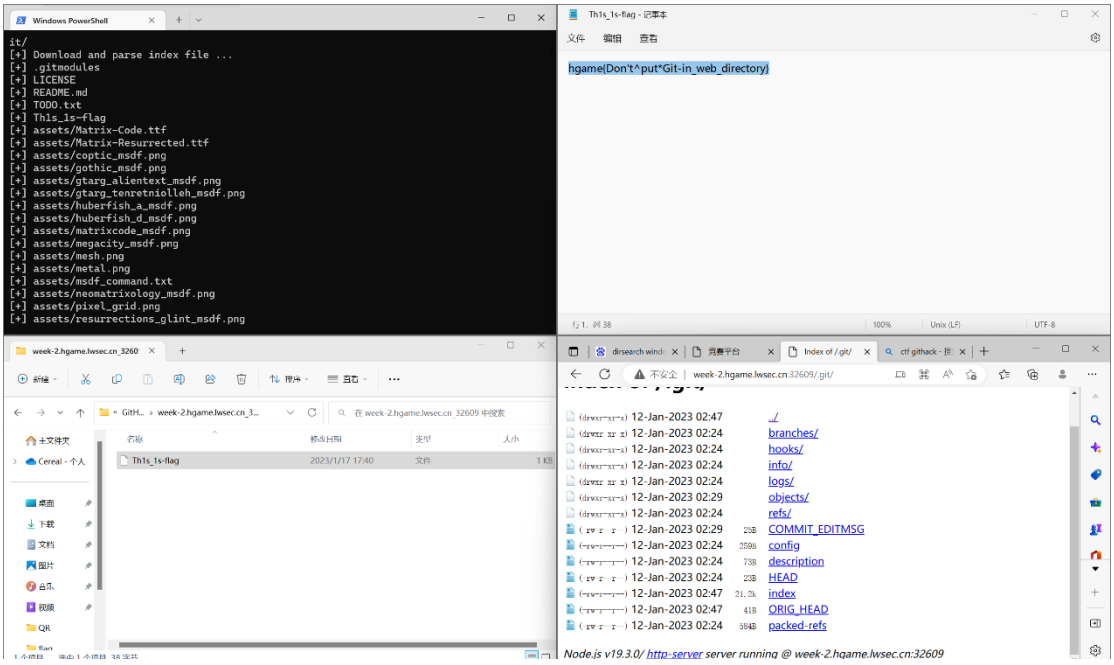
前言：年前年后家中事务很多，一周只有很少的时间拿来做题，没有能够尝试攻克难题（原计划是想办法完成 Web 和 Misc 的，结果只完成了一半），很可惜。许是选到了比较简单的题型，在第一周的学习后完成题目变得相当顺利（不过还是向学长们寻求了提点，感谢耐心的学长们）（寄希望于 week3 也能这样顺利完成）。

文章目录

Web

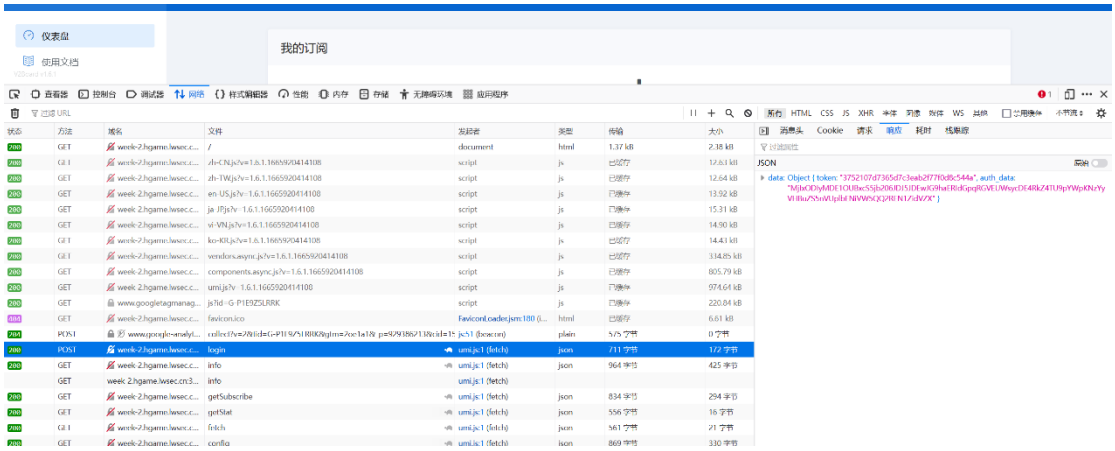
Git Leakage

由题目可知，本题与 Git 泄露相关，通过使用工具 Githack 成功取得 Flag。

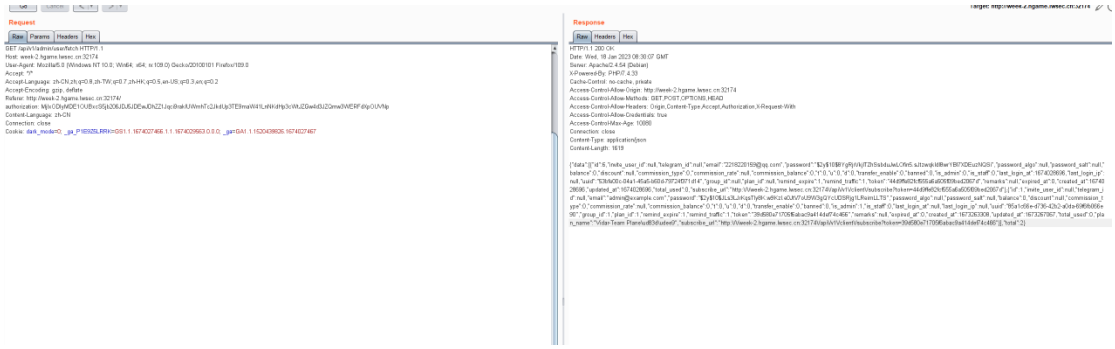


V2board

一开始毫无头绪，查找后发现 v2board1.6.1 具有代码漏洞，注册后在网页找到了 auth_data。



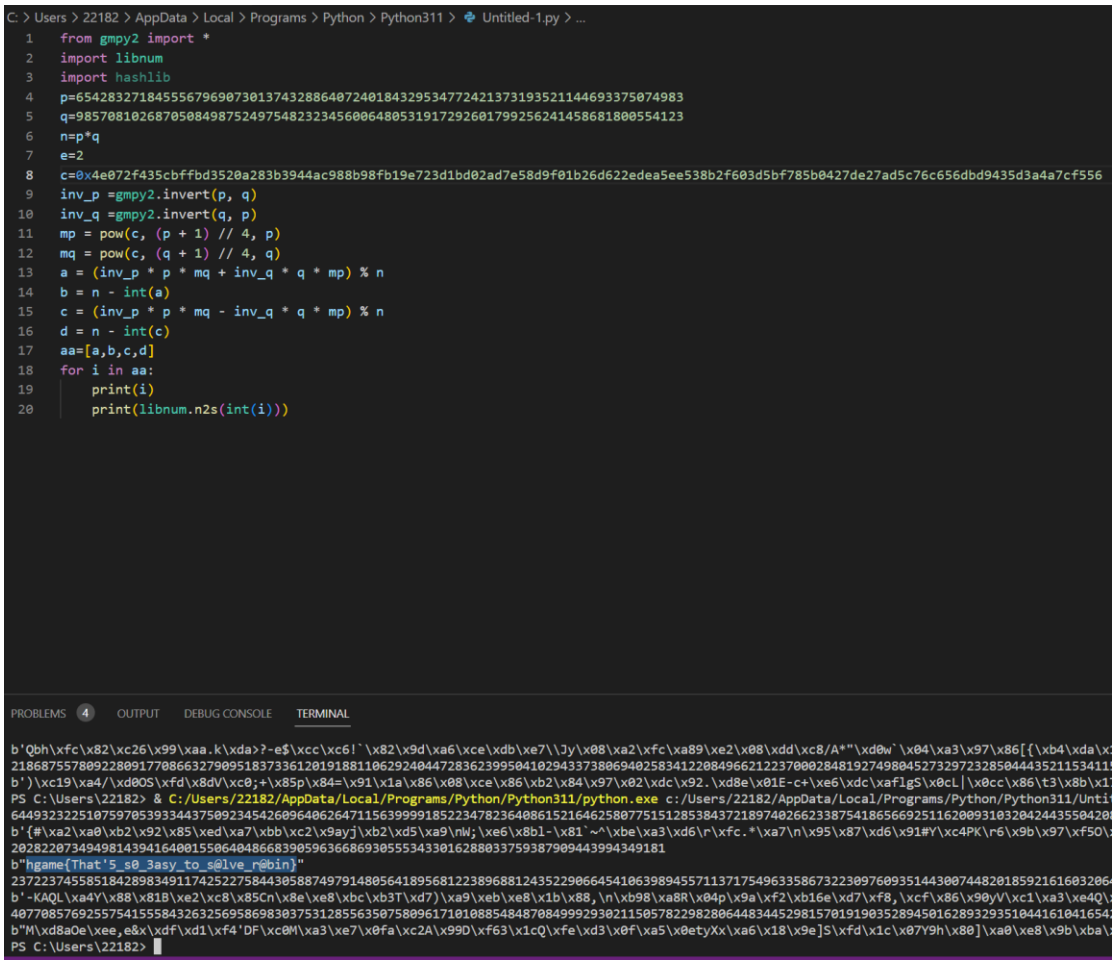
而后使用 burpsuite 抓包，将 Authorization 修改为刚刚得到的 auth_data 输入服务器，得到 Admin 访问权限，从而获得 Token。



Crypto

Rabin

由题目可知本题使用 Rabin 算法（即特殊的 RSA 题型），使用与上次 RSA 相近的脚本获取 Flag,同时因 Rabin 算法会得出四个值，都算了一遍枚举于下。



Misc

Sign In Pro Max

第一部分一眼丁真，鉴定为 base64，使用 base64 进行解码，得到了奇怪的东西，第一反应是可能就这样了，暂且搁置。

> Part1, is seems like baseXX:QVl5Y3BNQjE1ektibnU3SnN6M0tGaQ==
> (AYycpMB15zKbnu7Jsz3KFi)

第二部分、第三部分和第四部分在经历网上的学习后得知分别需要爆破 MD5、Sha1、Sha256 的 Hash 函数，照猫画虎写了半天爆破脚本总是无法运转，最后有幸在网上找到了爆破工具成功化解（上网爆破的效率比自身写脚本高了数十倍，编程能力的信心又一次遭到打击）。

> Part2, a hash function with 128bit digest size and 512bit block size: c629d83ff9804fb62202e90b0945a323
> (f91c)
>
> Part3, a hash function with 160bit digest size and 512bit block size: 99f3b3ada2b4675c518ff23cbd9539da05e2f1f8
> (4952)
>
> Part4, the next generation hash function of part3 with 256bit block size and 64 rounds: 1838f8d5b547c012404e53a9d8c76c56399507a2b017058ec7f27428fda5e7db
> (a3ed)

第五部分就是常见的凯撒密码（开头的 Part5 有点明显）

```
Ufwy5 nx 0gh0jf61i21h, stb uzy fqq ymj ufwyx ytljymjw, its'y ktwljy ymj ktwrfy.
 (Part5 is 0bc0ea61d21c, now put all the parts together, don't forget the format.)
```

然后粗略结合了五个部分直接填进去未果，感到很奇怪，结果从学长处得知 don't forget the format 不是虚言（0 基础萌新这才知道有个叫 UUID 的东西），而后发现似乎和 UUID 要求的 8-4-4-4-12 不大一致后开始尝试破解 Part1 的神秘字符串，在使用 base32、base16 轮番测试后无果感到奇怪，而后核实了依旧使用 base 编解码，之后在轮番测试后使用 base58 编码得到。

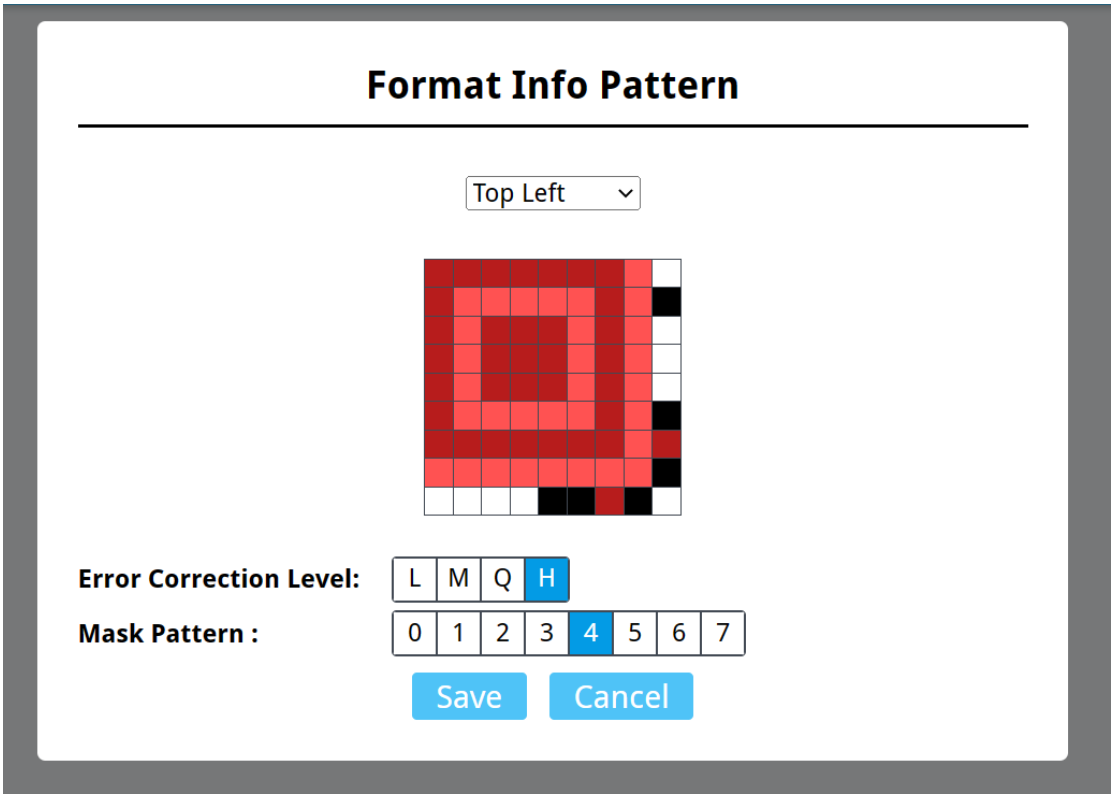
```
(MY2TCZBTMEYTQ===)
```

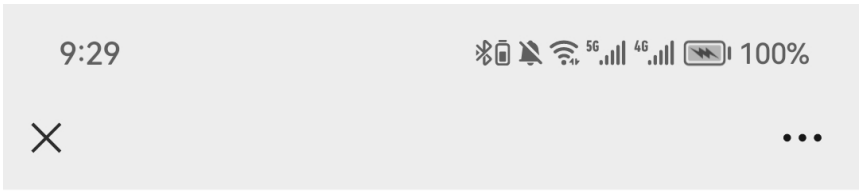
再根据刚刚学习到的 base 知识识破这是 base32，破解结合后得到 Flag。

```
Part1, is seems like baseXX:QVl5Y3BNQjE1ektibnU3SnN6M0tGaQ==
 (AYycpMB15zKbnu7Jsz3KFi)
(MY2TCZBTMEYTQ===)
(f51d3a18)

Part2, a hash function with 128bit digest size and 512bit block size: c629d83ff9804fb62202e90b0945a323
 (f91c)

Part3, a hash function with 160bit digest size and 512bit block size: 99f3b3ada2b4675c518ff23cbd9539da05e2f1f8
 (4952)

Part4, the next generation hash function of part3 with 256bit block size and 64 rounds: 1838f8d5b547c012404e53a9d8c76c56399507a2b017058ec7f27428fda5e7db
 (a3ed)

Ufwy5 nx 0gh0jf61i21h, stb uzy fqq ymj ufwyx ytljymjw, its'y ktwljy ymj ktwrfy.
 (Part5 is 0bc0ea61d21c, now put all the parts together, don't forget the format.)

f51d3a18-f91c-4952-a3ed-0bc0ea61d21c
```

Crazy_Qrcode

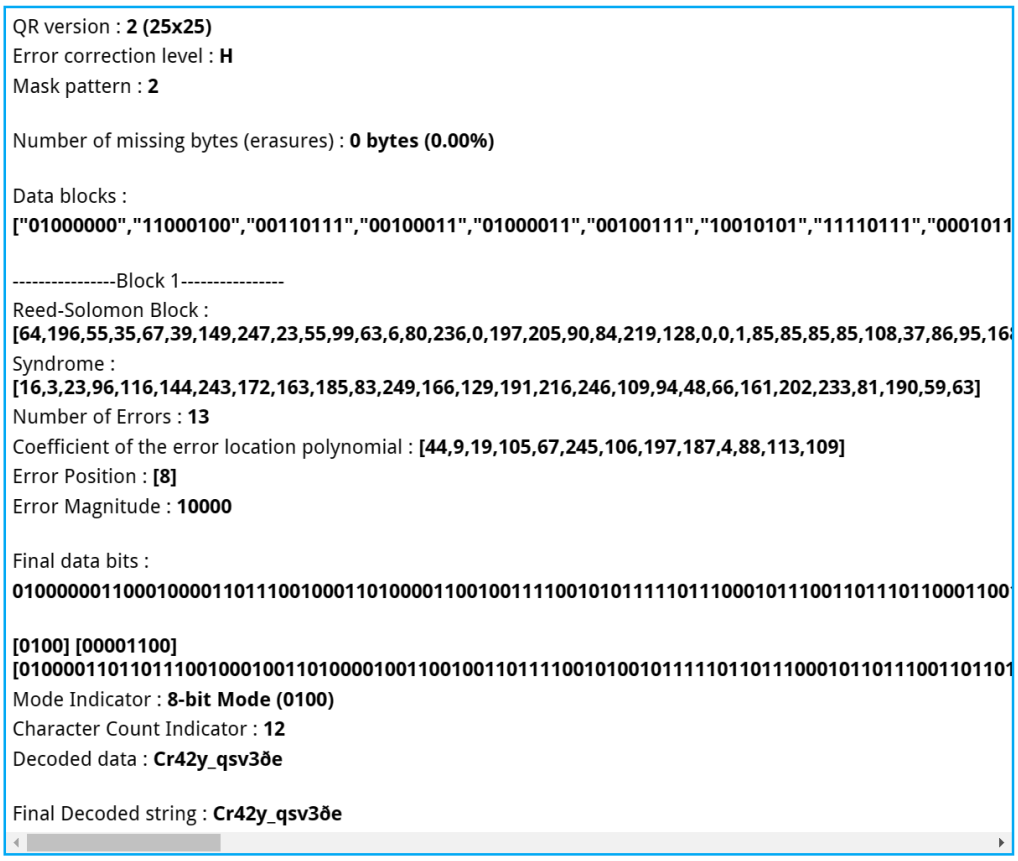
二维码扫不出来，好怪。在学长提点下去学习了二维码的构成并且在资料的收集过程中找到了神奇的网站（强力），想了半天感觉内容完整想要破坏二维码只能通过调整纠错等级和掩膜达成（一开始以为只有纠错等级，傻乎乎调了半天弄不出来），最后在这个设置下得到了扫得出来的二维码和内容（其实同时密码通过 Ziperello 爆破出来了）。





QDjkXkpM0BHNXujs

打开文件夹发现是 24 张二维码的碎片，据饭卡前辈说是要写个脚本爆破，但是那样感觉我根据在网上得到的知识拼了一个下午的 15 个正确部分有点可惜（实际上还是因为编出的脚本启动不了，编程能力的信心被打击*2），想了一下把剩下的碎片填了几个进去放到网站上扫扫看（反正二维码有容错，保不齐就出了），结果很幸运的得到了前一半部分的正确内容：



那这样后半部分的内容就能够推断出了（原本要枚举 26 万次，工作量大幅减少了），填入 Flag：Cr42y_qrc0de。