hackover18 writeup

threst / 2018-11-08 06:55:00 / 浏览数 1838 安全技术 CTF 顶(0) 踩(0)

I AM MANY

直接foremost分离即可

flag:hackover18{different_Fl4g_for_3arly_ch33tahz}

ez web

> Easy web challenge in the slimmest possible design.... namely none.
> http://ez-web.ctf.hackover.de:8080

发现有robots.txt文件，提示/flag/,进入文件夹,有个falg.txt,点击提示

> You do not have permission to enter this Area. A mail has been sent to our Admins.
> You shall be arrested shortly.

抓包修改Cookie: isAllowed=true

flag:hackover18{W3llD0n3,K1d.Th4tSh0tw4s1InAM1ll10n}

i-love-heddha

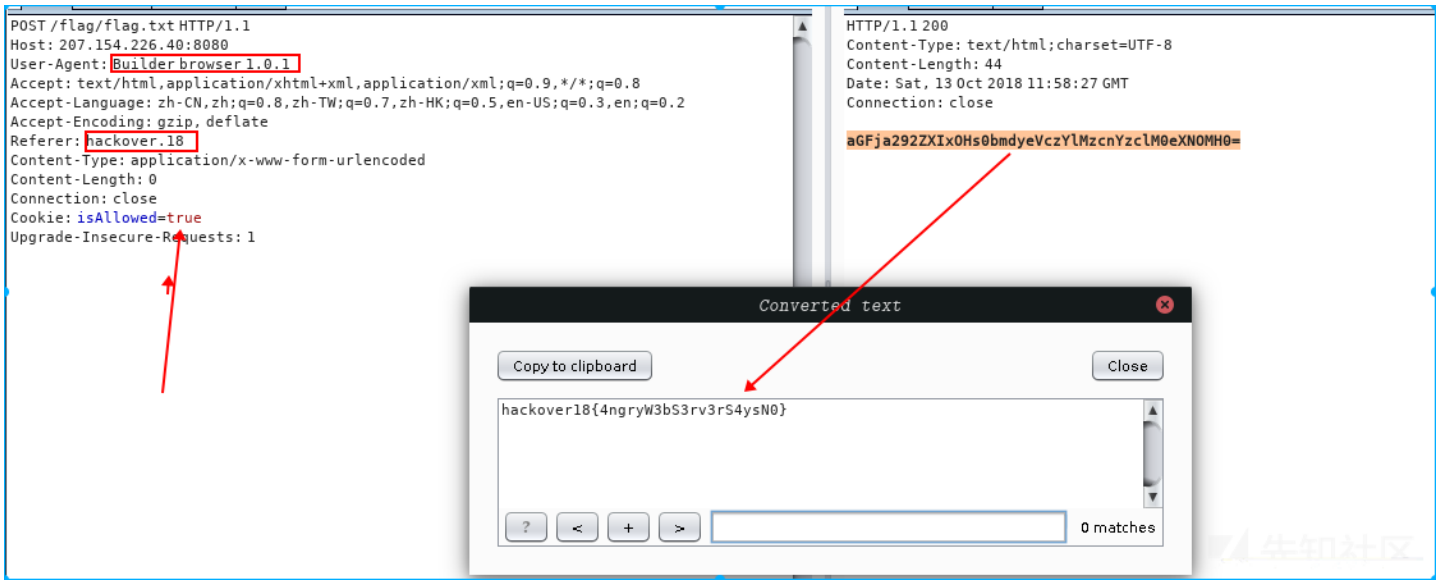
> A continuation of the Ez-Web challenge. enjoy
> 207.154.226.40:8080

是刚才那个的升级版,一样的找到/flag/flag.txt,设置isAllowed，

可是然后提示ou are using the wrong browser, 'Builder browser 1.0.1' is required

You are refered from the wrong location hackover.18 would be the correct place to come fro

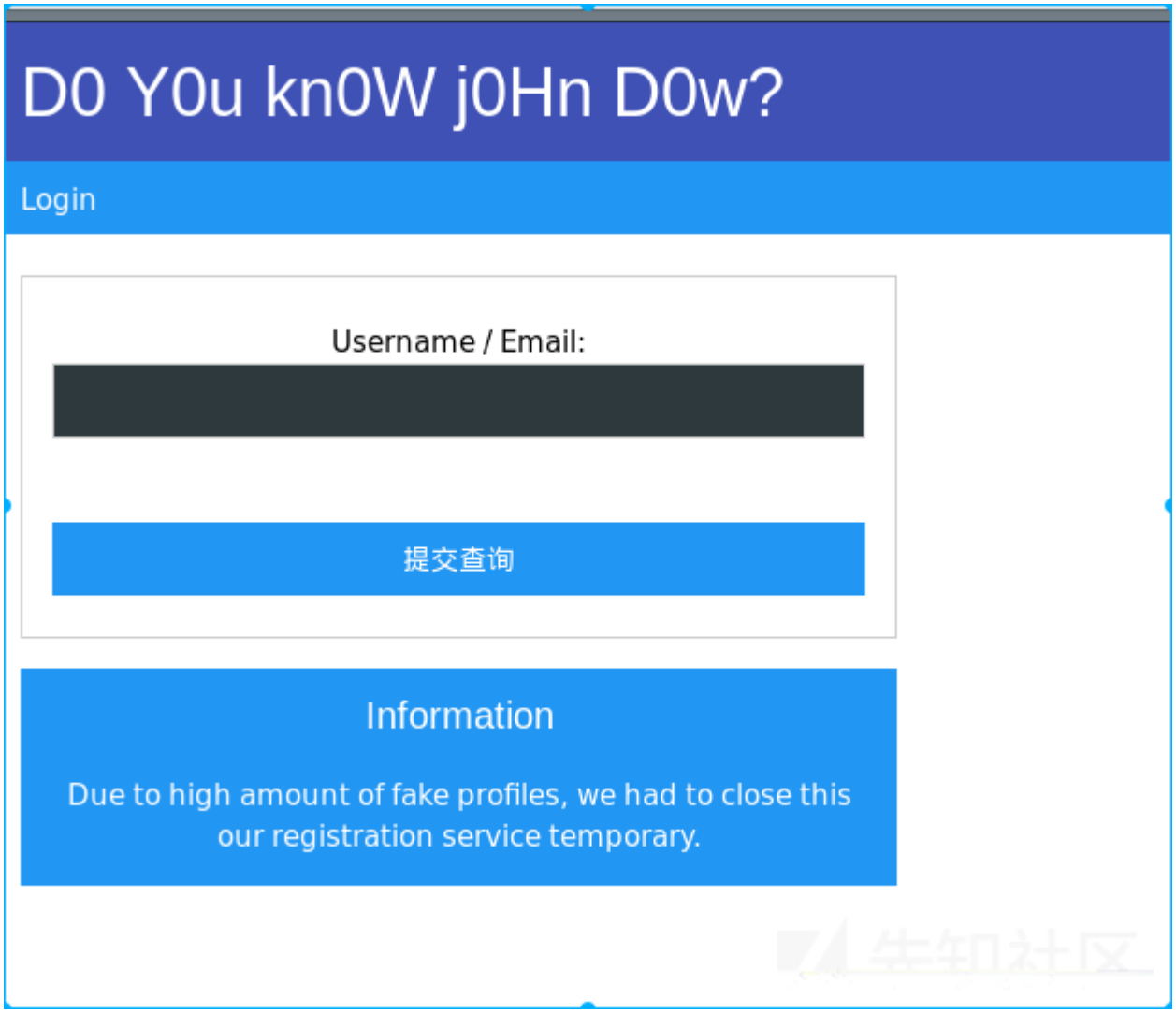
修改UA,referer,得到flag:hackover18{4ngryW3bS3rv3rS4ysN0}



who knows john dows?

> Howdy mate! Just login and hand out the flag, aye! You can find on h18johndoe has all you need!

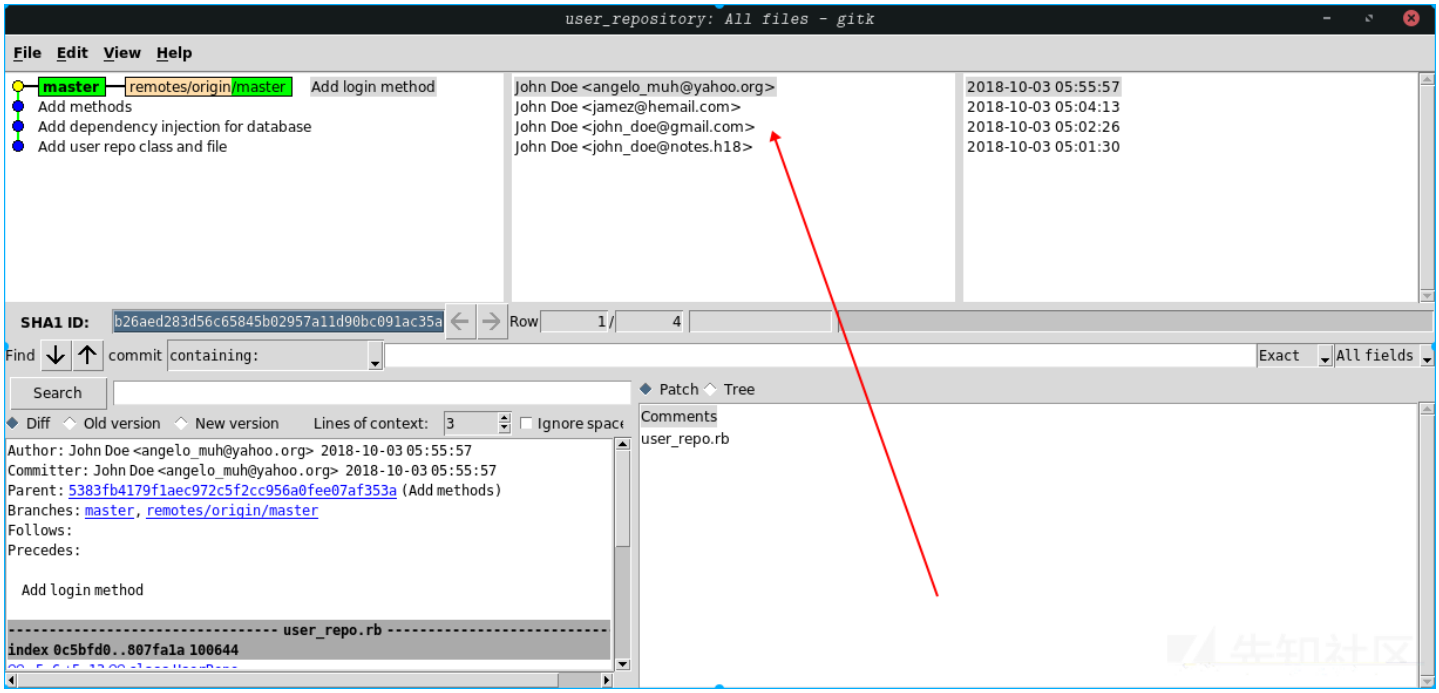
打开网站直接是要你输入用户名或邮箱



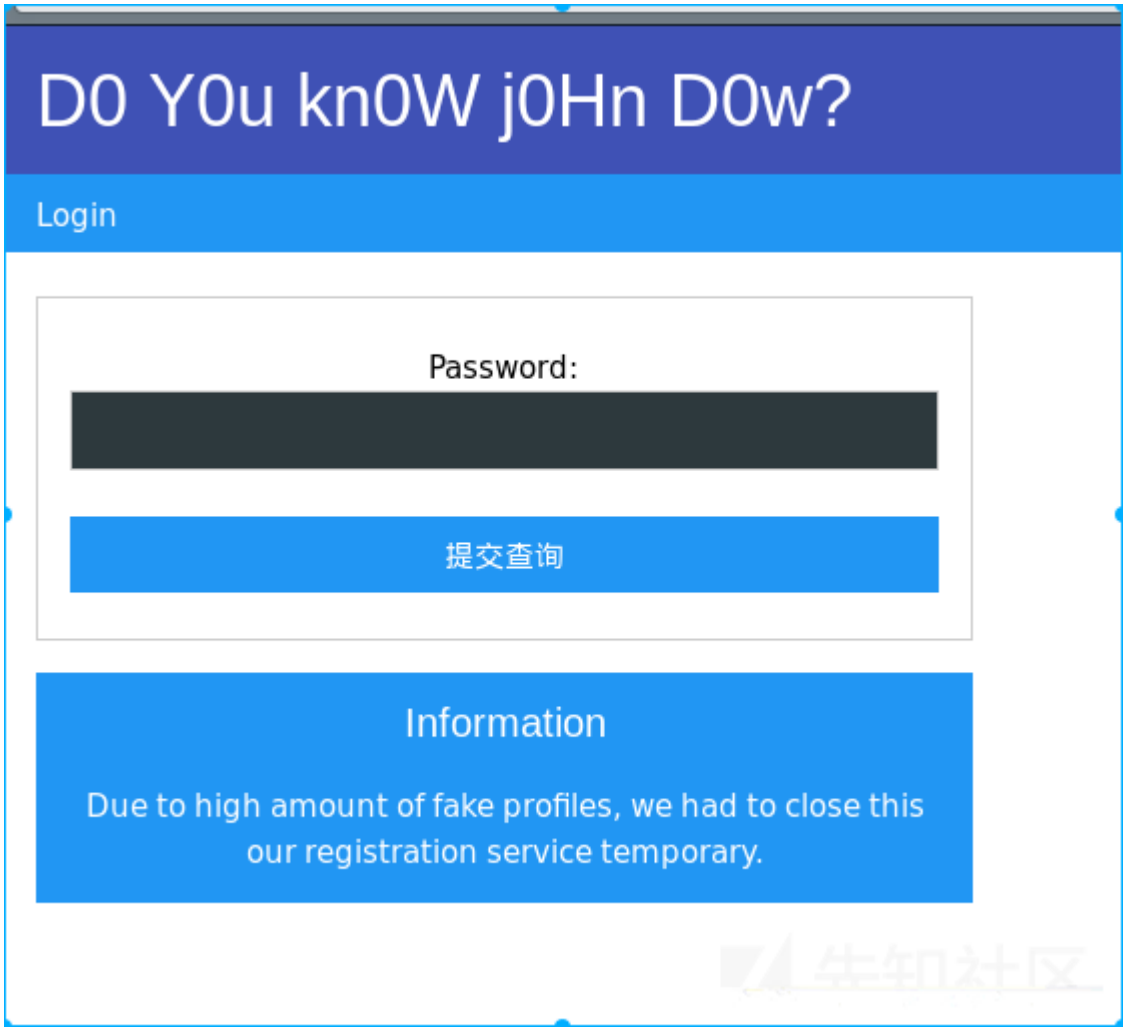
随便输一个，发现不对,根据提示h18johndoe,去github上面试试，果然搜到一个用户

github:https://github.com/h18johndoe

把这个仓库下下来`git clone https://github.com/h18johndoe/user_repository.git`



发现很多邮箱，一个一个去试试,尝试之后只有`john_doe@notes.h18`可以登录



登录之后提示输入密码，尝试万能密码' OR 1=1 --:,成功！！

flag:hackover18{I_KN0W_H4W_70_STALK_2018}

secure-hash

> We advise you to replace uses of unordered_hash with our new SecureHashtable class, since we added advanced crypto to make it 14.3 times more secure.
> Update: the binary was compiled with g++ and libstdc++, 64bit
> We're running a demo version, try it now:
> nc secure-hash.ctf.hackover.de 1337

源代码如下:

```
#include <openssl/evp.h>
#include <unordered_set>
#include <iostream>
#include <fstream>
#include <unistd.h>
// TODO - Make an #ifdef to detect openssl/libressl.
//#define EVP_CREATE_FN() EVP_MD_CTX_new()
//#define EVP_DESTROY_FN(x) EVP_MD_CTX_free(x)
#define EVP_CREATE_FN() EVP_MD_CTX_create()
#define EVP_DESTROY_FN(x) EVP_MD_CTX_cleanup(x)
enum auth_result {
   AUTH_FAILURE,
   AUTH_SUCCESS,
   AUTH_TIMEOUT,
};
class SecureHashtable {
private:
   const int MAX_SIZE = 15000;
   std::unordered_set<std::string> values;
   std::string sha512sum(const std::string& name, const std::string& password) {
```

```
        EVP_MD_CTX *mdctx;
        const EVP_MD *md;
        unsigned char md_value[EVP_MAX_MD_SIZE];
        unsigned int md_len;
        mdctx = EVP_CREATE_FN();
        md = EVP_get_digestbyname("sha512");
        EVP_MD_CTX_init(mdctx);
        EVP_DigestInit_ex(mdctx, md, NULL);
        EVP_DigestUpdate(mdctx, name.c_str(), name.size());
        EVP_DigestUpdate(mdctx, password.c_str(), password.size());
        EVP_DigestFinal_ex(mdctx, md_value, &md_len);
        EVP_DESTROY_FN(mdctx);
        return std::string(reinterpret_cast<char*>(md_value), md_len);
    }

public:
    SecureHashtable() {
        values.reserve(MAX_SIZE);
    }

    bool insert_keyvalue(const std::string& name, const std::string& password) {
        if (values.size() >= MAX_SIZE)
            return false; // Size limit exceeded.
        std::string digest = sha512sum(name, password);
        values.insert(digest);
        return true;
    }

    auth_result lookup_keyvalue(const std::string& name, const std::string& password) {
        std::string digest = sha512sum(name, password);
        size_t bucket = values.bucket(digest);
        auto it = values.begin(bucket), end = values.end(bucket);
        size_t iterations = 0;
        size_t MAX_ITERATIONS = 1000;
        while (it != end) {
            if (*it++ == digest)
                return AUTH_SUCCESS;
            // Avoid DoS attacks by fixing upper time limit.
            if (iterations++ >= MAX_ITERATIONS)
                return AUTH_TIMEOUT;
        }
        return AUTH_FAILURE;
    }
};


int main() {
    OpenSSL_add_all_digests();
    std::ifstream ifs("./flag.txt");
    std::string flag;
    ifs >> flag;
    SecureHashtable table;
    table.insert_keyvalue("root", flag);
    while (true) {
        usleep(1000);
        int choice;
        std::string name, password;
        printf("Main menu:\n1 - Register new user\n2 - Login\n");
        std::cin >> choice;

        printf("Name: ");
        std::cin >> name;

        printf("Password: ");
        std::cin >> password;

        if (choice == 1) {
            if (name == "root") {
                        printf("You are not root!\n");
```

```
                continue;
            }
            table.insert_keyvalue(name, password);
        } else if (choice == 2) {
            if (table.lookup_keyvalue(name, password)) {
                printf("Success! Logged in as %s\n", name.c_str());
                if (name == "root") {
                    printf("You win, the flag is %s\n", flag.c_str());
                    return 0;
                }
            } else {
                printf("Invalid credentials!\n");
            }
        } else {
            printf("Invalid choice!\n");
        }
    }
}
```

分析一下流程，首先要用户注册，可是不能注册root用户，但是在登录的时候要以root身份登录才可以获取flag

`EVP_MD_CTX_init`
该函数初始化一个EVP_MD_CTX结构

`EVP_DigestInit_ex`
该函数使用参数impl所指向的ENGINE设置该信息摘要结构体，参数ctx在调用本函数之前必须经过初始化。参数type通常是使用象EVP_sha1这种函数的返回值。假设impl为

`EVP_DigestUpdate`
该函数将参数d中的cnt字节数据进行信息摘要到ctx结构中去。该函数能够被调用多次。用以对很多其它的数据进行信息摘要。操作成功返回1，否则返回0。

`EVP_DigestFinal_ex`
本函数将ctx结构中的摘要信息数据返回到参数md中，假设参数s不是NULL，那么摘要数据的长度（字节）就会被写入到参数s中，大多数情况下，写入的值是EVP_MAX_MD

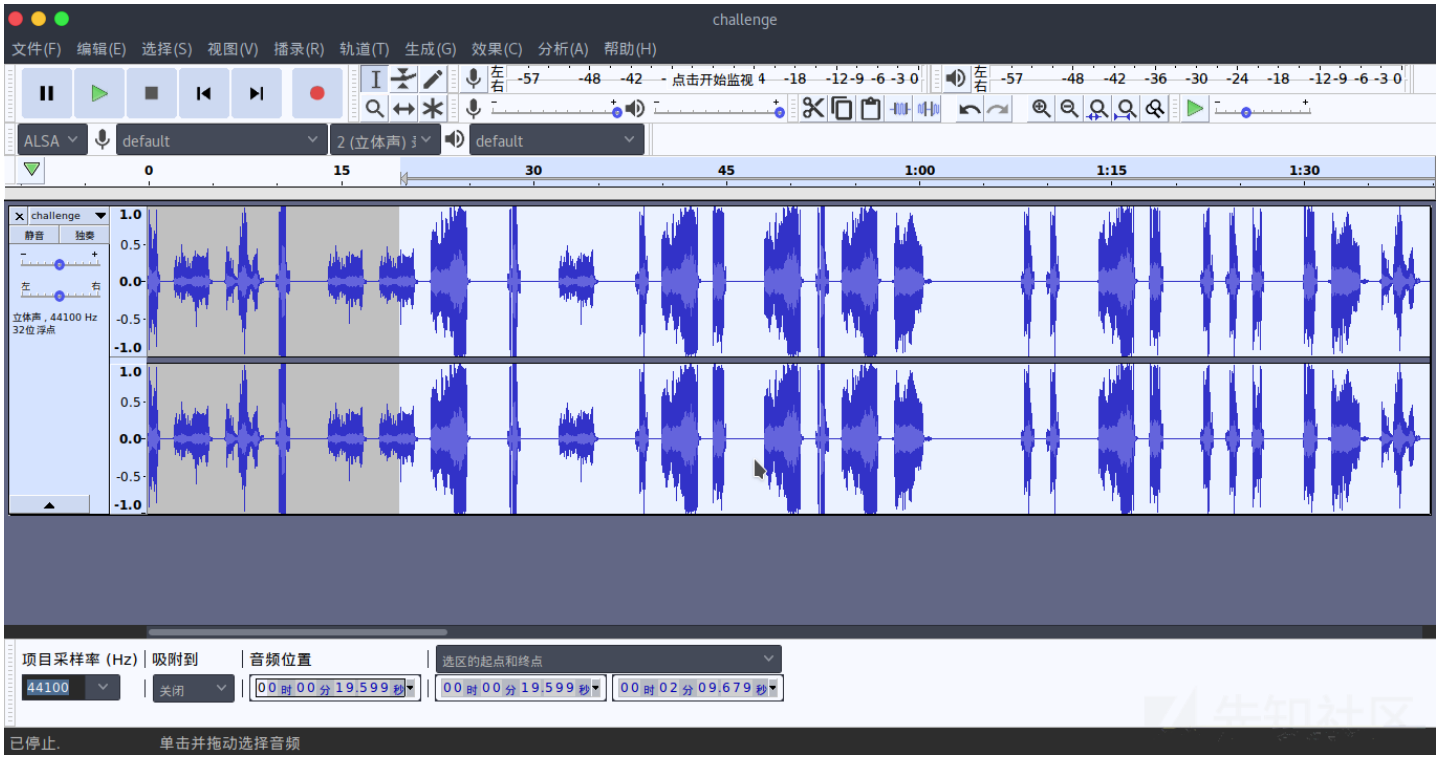
特别注意，名称和密码一个接一个地添加到摘要中，并且使用std :: string :: size确定添加的字节数，它返回字符串的实际字节数，不包括null字节。使用例如name ==“fo”和password =“obar”可以实现相同的结果，因此这两组凭证将导致相同的摘要，因此std :: unordered_set中的桶相同。现在我们来试试。我们注册名称为==“ro”且密码==“ot1”的用户，然后只需尝试登录名称==“root”和密码==“1”

flag:hackover18{00ps_y0u_mu5t_h4ve_h1t_a_v3ry_unlikely_5peci4l_c4s3}

Hummel Hummel

> There is no "hackover18{*}" in the word. the solution has to be inserted as hackover18{mysolution}.

下载下来一个mp4的文件，播放看见一个马在打屁，可是很有节奏，用audacity查看，发现



看起来像是莫斯密码,上下相连的为.,其他为-,全部连起来就是这样

```
.--. --- . - .-. -.-- / .. -. ... .--. .. .-. . -.. / -... -.-- / -... .- -.- . -.. / -... . .- -. .....
```

flag:hackover18{poetry inspired by baked beans}

UnbreakMyStart

题目是xz文件，但是看起来好像损坏了

```
$ xxd unbreak_my_start.tar.xz
0000000: 504b 0304 1400 0800 0800 04e6 d6b4 4602  PK............F.
0000010: 0021 0116 0000 0074 2fe5 a3e0 07ff 007d  .!.....t/......}
0000020: 5d00 331b 0847 5472 2320 a8d7 45d4 9ae8  ].3..GTr# ..E...
0000030: 3a57 139f 493f c634 8905 8c4f 0bc6 3b67  :W..I?.4...O..;g
0000040: 7028 1a35 f195 abb0 2e26 666d 8c92 da43  p(.5.....&fm...C
0000050: 11e1 10ac 4496 e2ed 36cf 9c99 afe6 5a8e  ....D...6.....Z.
0000060: 311e cb99 f4be 6dca 943c 4410 8873 428a  1.....m..<D..sB.
0000070: 7c17 f47a d17d 7808 b7e4 22b8 ec19 9275  |..z.}x..."....u
0000080: 5073 0c34 5f9e 14ac 1986 d378 7b79 9f87  Ps.4_......x{y..
0000090: 0623 7369 4372 19da 6e33 0217 7f8d 0000  .#siCr..n3......
00000a0: 0000 001c 0f1d febd b436 8c00 0199 0180  .........6......
00000b0: 1000 00ad af23 35b1 c467 fb02 0000 0000  .....#5..g......
00000c0: 0459 5a                                                  .YZ
```

这个PK是zip文件常见的,参考这个xz文件格式https://tukaani.org/xz/xz-file-format-1.0.4.txt

我们尝试用我们构造的头替换文件的前11个字节

```
$ dd if=unbreak_my_start.tar.xz of=trimmed.bin bs=1 skip=11
184+0 records in
184+0 records out
184 bytes transferred in 0.000920 secs (199988 bytes/sec)
$ (printf "\xFD7zXZ\x00\x00\x04"; cat trimmed.bin) > fixed.tar.xz
$ xz -d fixed.tar.xz
$ cat fixed.tar
flag.txt000644 001750 001750 00000000045 13340067500 013221 0ustar00heddhaheddha000000 000000 hackover18{U_f0und_th3_B3st_V3rs
```

得到flag

flag:hackover18{U_f0und_th3_B3st_V3rs10n}

上一篇：伊朗用户的Instagram电报安... 下一篇：伊朗用户的Instagram电报安...

1. 0 条回复
   - 动动手指，沙发就是你的了！

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板