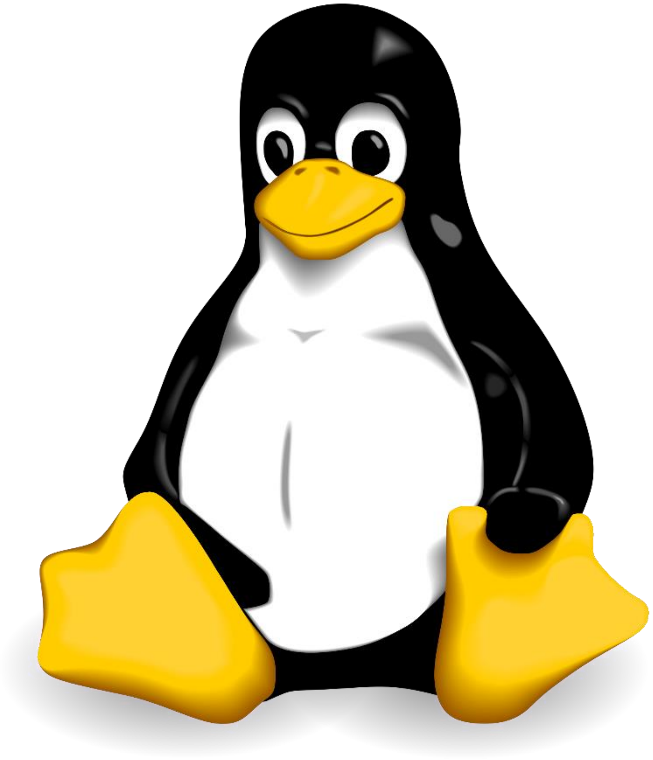
COMP3052.SEC Computer Security

Session 10: OS Security I: Unix and Linux Security



Acknowledgements

- Some of the materials we use this semester may come directly from previous teachers of this module, and other sources ...
- Thank you to (amongst others):
  - Michel Valstar, Milena Radenkovic, Mike Pound, Dave Towey, …

This Session

- Unix and Linux Security
  - Users, Groups, Root
  - Permissions

Role of the OS

- Compactly combine:
  - Identification
  - Authentication
  - Access control
  - Auditing
- User accounts to store permissions
- Installation and configuration

UID / GID

- Usernames in Unix / Linux are soft aliases, your UID is what determines permissions
    - User identities: UID
    - Group identities: GID
- Your IDs are stored in /etc/passwd
- Root has a special UID: 0

/etc/passwd

- /etc/passwd stores user accounts, not just passwords

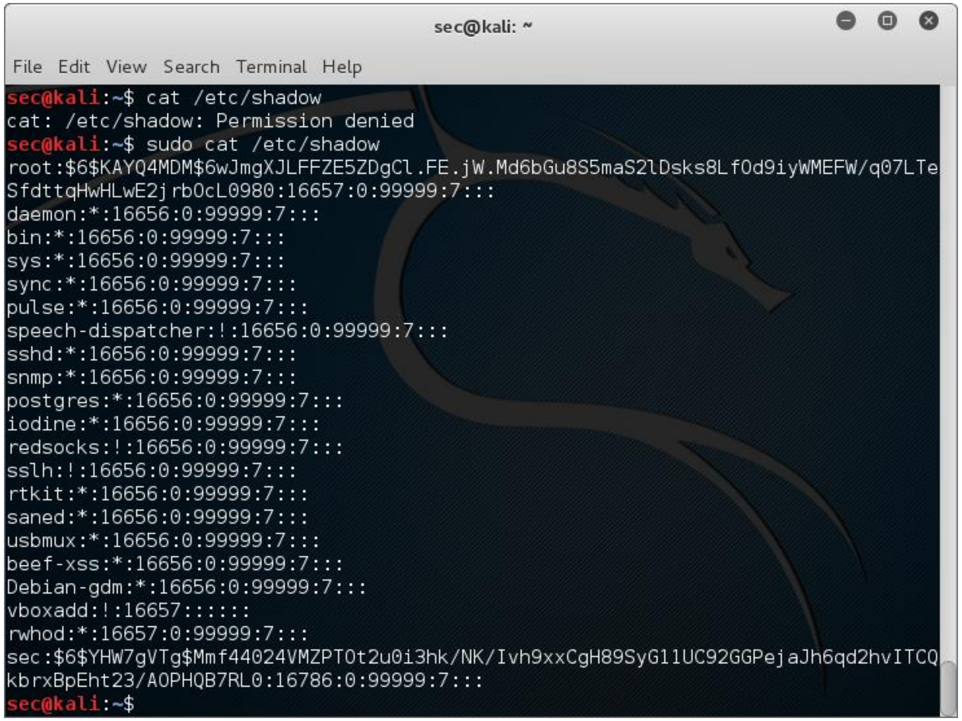
```
username:password:UID:GID:ID string:home dir:login shell

sec:x:1000:1001:Dr. S Security:/home/sec:/bin/bash

root:x:0:0:root:/root:/bin/bash
```

The Shadow File

- In an attempt to improve password security, we can store password hashes in a shadow file
  - Readable only by root users
- /etc/shadow stores the hashed passwords needed to authenticate users



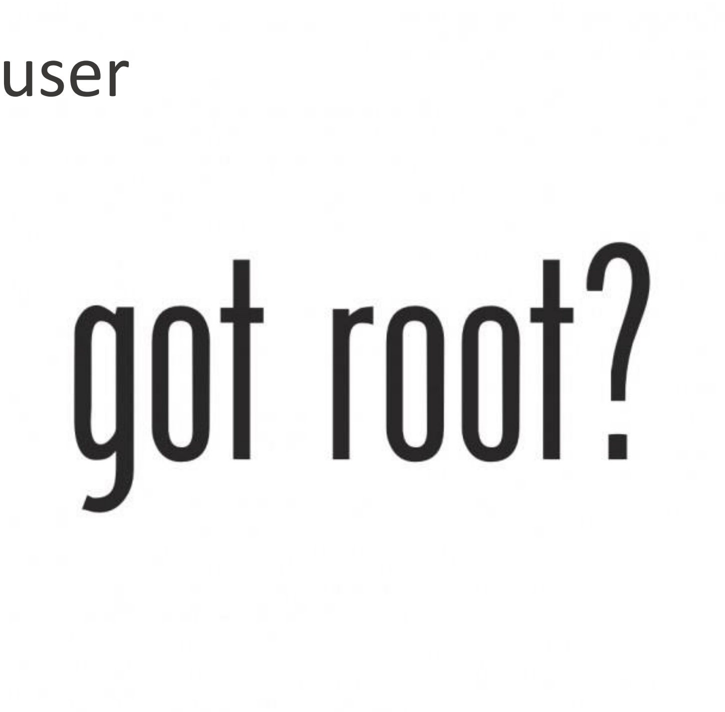
```
sec@kali:~$ cat /etc/shadow
cat: /etc/shadow: Permission denied
sec@kali:~$ sudo cat /etc/shadow
root:$6$KAYQ4MDM$6wJmgXJLFFZE5ZDgCl.FE.jW.Md6bGu8S5maS2lDsks8Lf0d9iyWMEFW/q07LTe
SfdttqHwHLwE2jrb0cL0980:16657:0:99999:7:::
daemon:*:16656:0:99999:7:::
bin:*:16656:0:99999:7:::
sys:*:16656:0:99999:7:::
sync:*:16656:0:99999:7:::
pulse:*:16656:0:99999:7:::
speech-dispatcher:!:16656:0:99999:7:::
sshd:*:16656:0:99999:7:::
snmp:*:16656:0:99999:7:::
postgres:*:16656:0:99999:7:::
iodine:*:16656:0:99999:7:::
redsocks:!:16656:0:99999:7:::
sslh:!:16656:0:99999:7:::
rtkit:*:16656:0:99999:7:::
saned:*:16656:0:99999:7:::
usbmux:*:16656:0:99999:7:::
beef-xss:*:16656:0:99999:7:::
Debian-gdm:*:16656:0:99999:7:::
vboxadd:!:16657::::::
rwhod:*:16657:0:99999:7:::
sec:$6$YHW7gVTg$Mmf44024VMZPTOt2u0i3hk/NK/Ivh9xxCgH89SyG11UC92GGPejaJh6qd2hvITCQ
kbrxBpEht23/AOPHQB7RL0:16786:0:99999:7:::
sec@kali:~$
```

/etc/pam.d/common-password

```
password [success=1 default=ignore]     pam_unix.so obscure sha512 rounds=65535
                                         pam_cracklib.so
```

Root

- The all powerful Unix Superuser
- Login
- Audio
- I/O
- Limits:
    - Decrypt hashed passwords



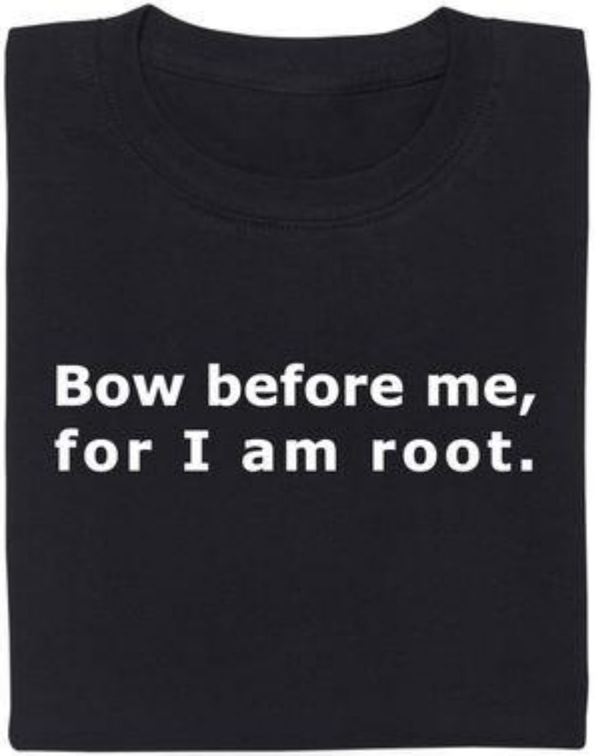
Root in Linux

- Root’s UID 0 is actually hard coded into the Linux kernel at multiple points
    - This means you shouldn’t change root’s UID!
- In 2003, this anonymous change was made to the error value return in the wait4 function in Linux:

```
if ((options == (__WCLONE|__WALL))
               && (current->uid = 0))
        retval = -EINVAL;
```

Root Management

- Write protect /etc/passwd and /etc/group – Obviously!
- Separate superuser duties
- Never use root as a normal use
- Audit su and sudo usage



Remote Access

- Being able to SSH etc. using root is a very bad idea
  - You can disable root logins in /etc/ssh/sshd_config by changing “PermitRootLogin”
  - You can also enforce public-key authentication for users using RSA
- Never use FTP or Telnet

Objects

- In Unix, everything is a file
- Files really represent resources
- Organised in a tree structure, with alterations depending on the file system
- Inodes store permission information
- Every resource has an owner and a group

inodes

- Inodes in Unix and Linux store the metadata for files
- Each file name links to an inode, which stores security information

```
michael@psbss01:~$ stat /etc/passwd
  File: `/etc/passwd`
  Size: 2104            Blocks: 8          IO Block: 4096   regular file
Device: fc00h/64512d    Inode: 3410837     Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2016-03-08 16:17:01.101881805 +0000
Modify: 2016-02-25 15:53:50.618716909 +0000
Change: 2016-02-25 15:53:50.662716909 +0000
 Birth: -
michael@psbss01:~/deep-learning/wheat-detection$
```

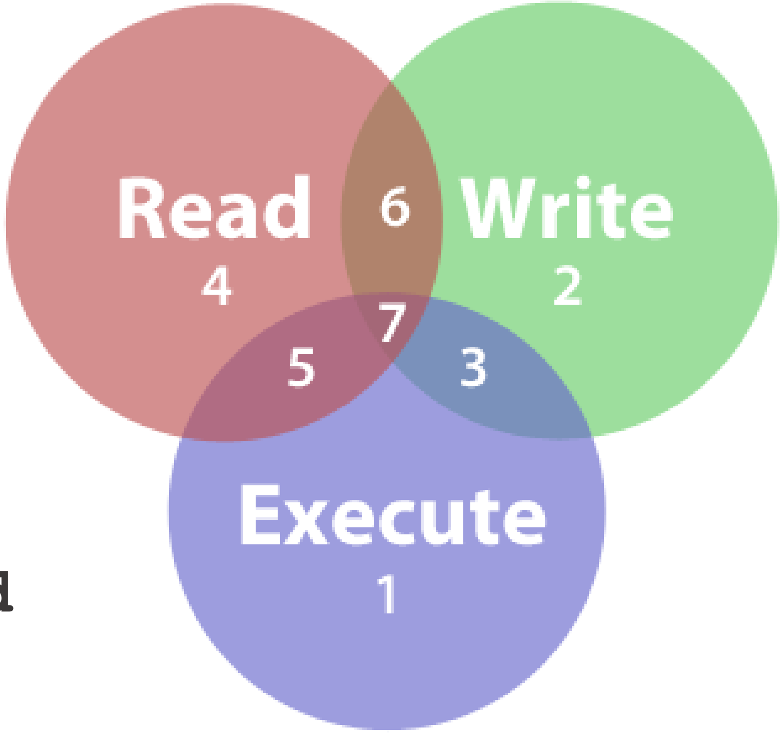
Permissions

- Every resource has permission bits
  - held in the inode metadata
- Permissions for the user / group / others
- Octal representation (bit-wise, really)
  - Bit 3: read (0x4, octal/decimal)
  - Bit 2: write (0x2, octal/decimal)
  - Bit 1: execute (0x1, octal/decimal)

Octal

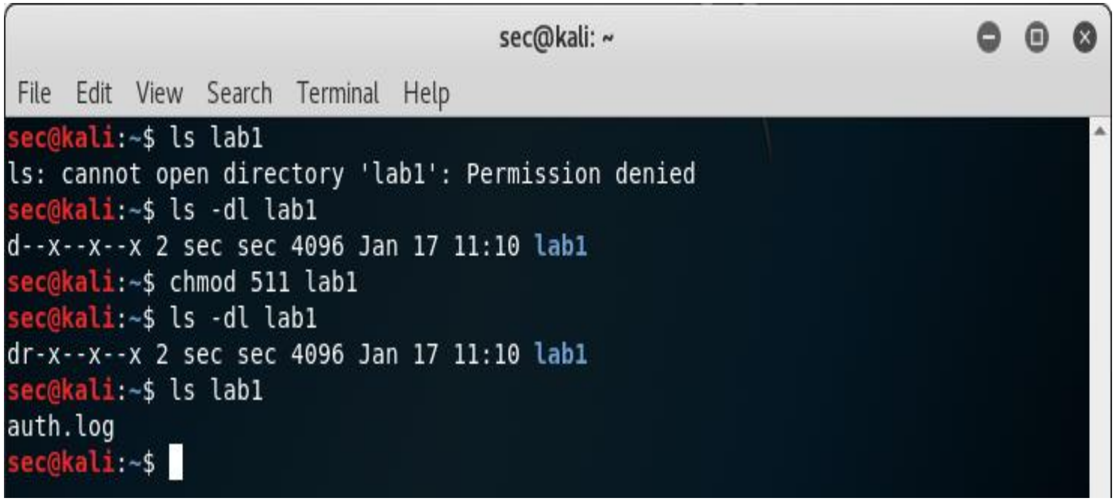
- Permissions are changed using chmod, usually by passing three octal values
- E.g.

```
chmod 754 /etc/passwd
```



Directories

- Directory permissions are slightly different to files:
  - r – List files within the directory
  - w – add or remove files
  - x – traverse directory, open files in the directory



```
sec@kali:~$ ls lab1
ls: cannot open directory 'lab1': Permission denied
sec@kali:~$ ls -dl lab1
d--x--x--x 2 sec sec 4096 Jan 17 11:10 lab1
sec@kali:~$ chmod 511 lab1
sec@kali:~$ ls -dl lab1
dr-x--x--x 2 sec sec 4096 Jan 17 11:10 lab1
sec@kali:~$ ls lab1
auth.log
sec@kali:~$ 
```

Can't view files in lab1

View permissions on lab1

Add user read permission

We can now look at files in lab1

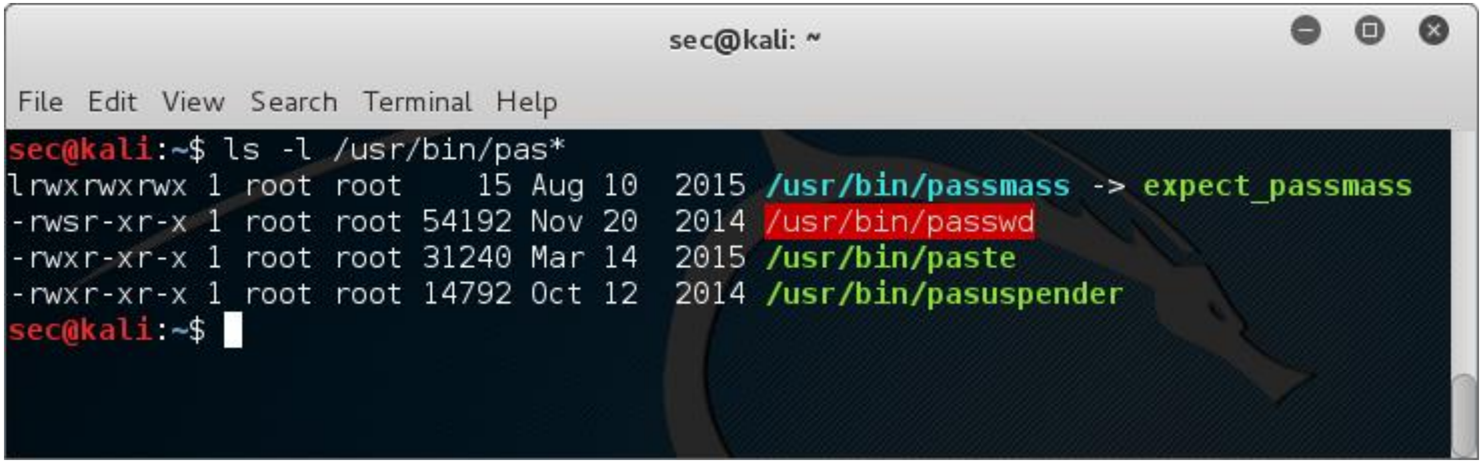
Exercise

```
-rwxr-xr-x 1 sbzmpp staff 8844 Mar 23 14:20 index.html
```

- What are the permissions on index.html in octal? 755
- Given a directory with octal permissions 754, can others:
  - List its contents? Yes
  - Execute known files? No – no execute
  - Delete files? No – no write

SUID

- Set UID: set the effective user to be the file owner when executed
- Necessary to allow non-privileged access to privileged actions e.g. passwords
- **Dangerous!**



```
sec@kali:~$ ls -l /usr/bin/pas*
lrwxrwxrwx 1 root root    15 Aug 10  2015 /usr/bin/passmass -> expect_passmass
-rwsr-xr-x 1 root root 54192 Nov 20  2014 /usr/bin/passwd
-rwxr-xr-x 1 root root 31240 Mar 14  2015 /usr/bin/paste
-rwxr-xr-x 1 root root 14792 Oct 12  2014 /usr/bin/pasuspender
sec@kali:~$ 
```

Search Paths

- Environment variables that many processes can set
- Can be easily misused to place Trojans

- Use ‘which’ to determine which resource is really used
- Use full paths where possible

```
michael@psbss01:~$ echo $PATH
usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
michael@psbss01:~$ which ls
/bin/ls
michael@psbss01:~$ sudo mv trojan /usr/bin/ls
michael@psbss01:~$ which ls
/usr/bin/ls
michael@psbss01:~$
```

Summary

- Unix and Linux Security
  - Users, Groups, Root
  - Permissions

Gollmann Chapter 7