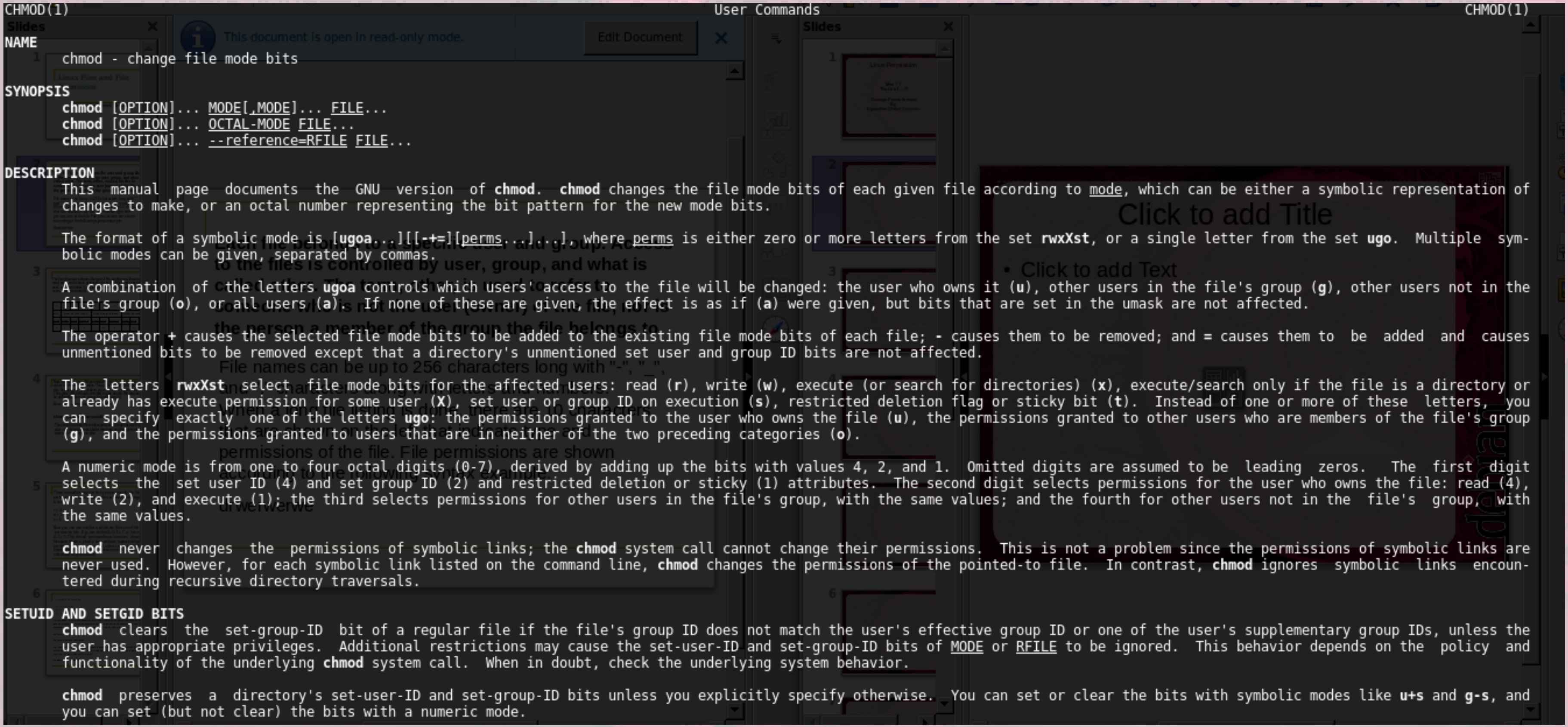
Need Permissions for Hackers??

Man ??
Whoami??
Ls -all ??
Chmod??
Chown??

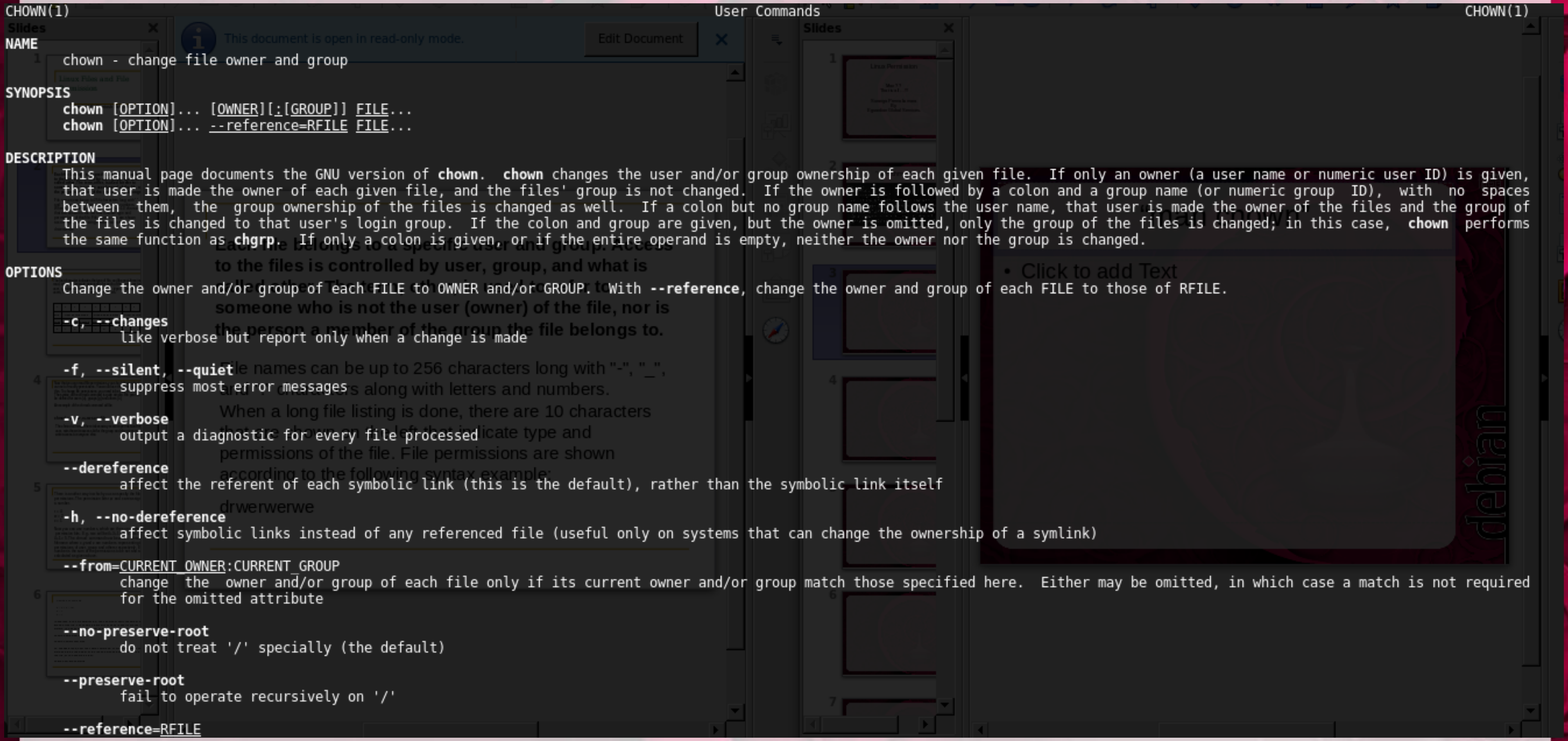
Suranga Premakumara
By
Eguardian Global Services.

“man chmod”



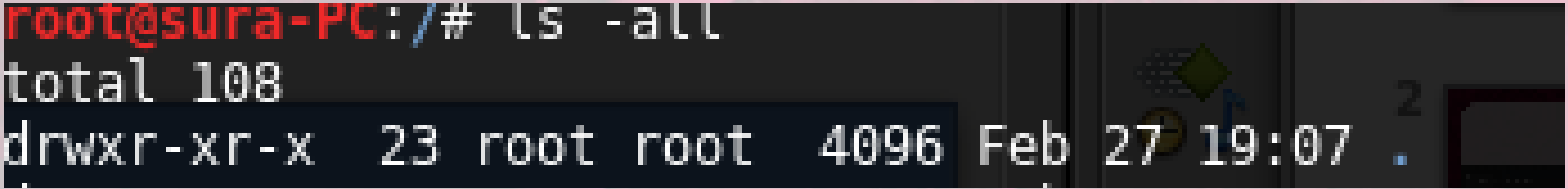
```
CHMOD(1)                                         User Commands                                         CHMOD(1)

NAME
       chmod - change file mode bits

SYNOPSIS
       chmod [OPTION]... MODE[,MODE]... FILE...
       chmod [OPTION]... OCTAL-MODE FILE...
       chmod [OPTION]... --reference=RFILE FILE...

DESCRIPTION
       This  manual  page  documents  the  GNU  version  of chmod.  chmod changes the file mode bits of each given file according to mode, which can be either a symbolic representation of
       changes to make, or an octal number representing the bit pattern for the new mode bits.

       The format of a symbolic mode is [ugoa...][[-+=][perms...]...], where perms is either zero or more letters from the set rwxXst, or a single letter from the set ugo.  Multiple  sym-
       bolic modes can be given, separated by commas.

       A  combination  of  the letters ugoa controls which users' access to the file will be changed: the user who owns it (u), other users in the file's group (g), other users not in the
       file's group (o), or all users (a).  If none of these are given, the effect is as if (a) were given, but bits that are set in the umask are not affected.

       The operator + causes the selected file mode bits to be added to the existing file mode bits of each file; - causes them to be removed; and = causes them to  be  added  and  causes
       unmentioned bits to be removed except that a directory's unmentioned set user and group ID bits are not affected.

       The  letters  rwxXst  select  file mode bits for the affected users: read (r), write (w), execute (or search for directories) (x), execute/search only if the file is a directory or
       already has execute permission for some user (X), set user or group ID on execution (s), restricted deletion flag or sticky bit (t).  Instead of one or more of these  letters,  you
       can  specify  exactly  one of the letters ugo: the permissions granted to the user who owns the file (u), the permissions granted to other users who are members of the file's group
       (g), and the permissions granted to users that are in neither of the two preceding categories (o).

       A numeric mode is from one to four octal digits (0-7), derived by adding up the bits with values 4, 2, and 1.  Omitted digits are assumed to be  leading  zeros.   The  first  digit
       selects  the  set user ID (4) and set group ID (2) and restricted deletion or sticky (1) attributes.  The second digit selects permissions for the user who owns the file: read (4),
       write (2), and execute (1); the third selects permissions for other users in the file's group, with the same values; and the fourth for other users not in the  file's  group,  with
       the same values.

       chmod  never  changes  the  permissions of symbolic links; the chmod system call cannot change their permissions.  This is not a problem since the permissions of symbolic links are
       never used.  However, for each symbolic link listed on the command line, chmod changes the permissions of the pointed-to file.  In contrast, chmod ignores  symbolic  links  encoun-
       tered during recursive directory traversals.

SETUID AND SETGID BITS
       chmod  clears  the  set-group-ID  bit of a regular file if the file's group ID does not match the user's effective group ID or one of the user's supplementary group IDs, unless the
       user has appropriate privileges.  Additional restrictions may cause the set-user-ID and set-group-ID bits of MODE or RFILE to be ignored.  This behavior depends on the  policy  and
       functionality of the underlying chmod system call.  When in doubt, check the underlying system behavior.

       chmod  preserves  a  directory's set-user-ID and set-group-ID bits unless you explicitly specify otherwise.  You can set or clear the bits with symbolic modes like u+s and g-s, and
       you can set (but not clear) the bits with a numeric mode.
```

“man chown”

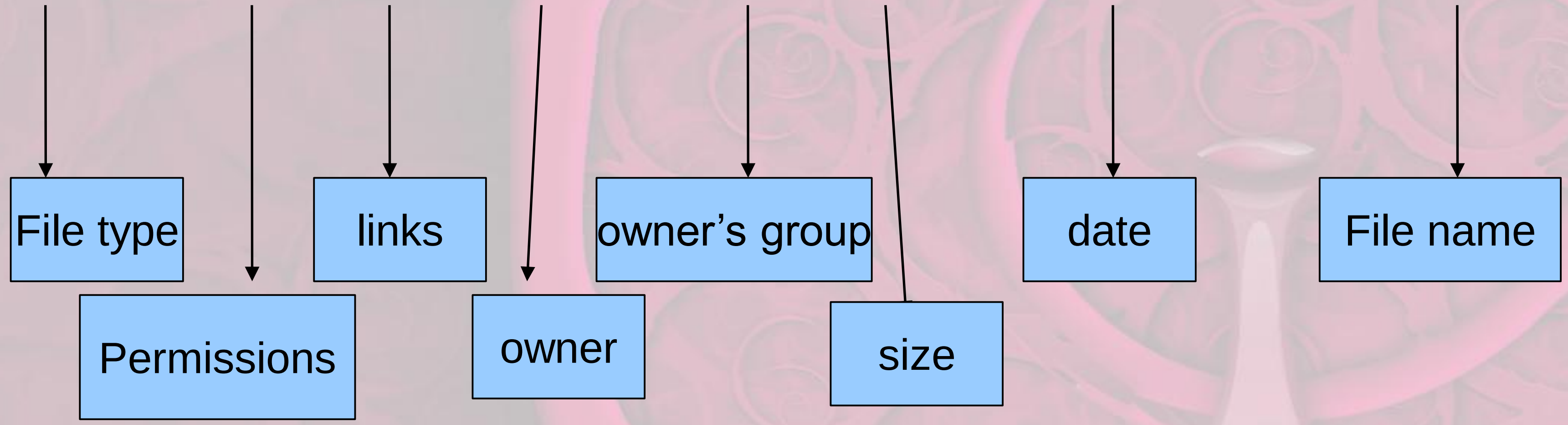


```
CHOWN(1)                                   User Commands                                   CHOWN(1)

NAME
       chown - change file owner and group

SYNOPSIS
       chown [OPTION]... [OWNER][:[GROUP]] FILE...
       chown [OPTION]... --reference=RFILE FILE...

DESCRIPTION
       This manual page documents the GNU version of chown.  chown changes the user and/or group ownership of each given file.  If only an owner (a user name or numeric user ID) is given,
       that user is made the owner of each given file, and the files' group is not changed.  If the owner is followed by a colon and a group name (or numeric group  ID),  with  no  spaces
       between  them,  the  group ownership of the files is changed as well.  If a colon but no group name follows the user name, that user is made the owner of the files and the group of
       the files is changed to that user's login group.  If the colon and group are given, but the owner is omitted, only the group of the files is changed; in this case,  chown  performs
       the same function as chgrp.  If only a colon is given, or if the entire operand is empty, neither the owner nor the group is changed.

OPTIONS
       Change the owner and/or group of each FILE to OWNER and/or GROUP.  With --reference, change the owner and group of each FILE to those of RFILE.

       -c, --changes
              like verbose but report only when a change is made

       -f, --silent, --quiet
              suppress most error messages

       -v, --verbose
              output a diagnostic for every file processed

       --dereference
              affect the referent of each symbolic link (this is the default), rather than the symbolic link itself

       -h, --no-dereference
              affect symbolic links instead of any referenced file (useful only on systems that can change the ownership of a symlink)

       --from=CURRENT_OWNER:CURRENT_GROUP
              change  the  owner and/or group of each file only if its current owner and/or group match those specified here.  Either may be omitted, in which case a match is not required
              for the omitted attribute

       --no-preserve-root
              do not treat '/' specially (the default)

       --preserve-root
              fail to operate recursively on '/'

       --reference=RFILE
```

Ls - all



```
root@sura-PC:/# ls -all
total 108
drwxr-xr-x  23 root root  4096 Feb 27 19:07 .
```



File type

Permissions

links

owner

owner’s group

size

date

File name

Only owner or root can change the permission.

Permission definition

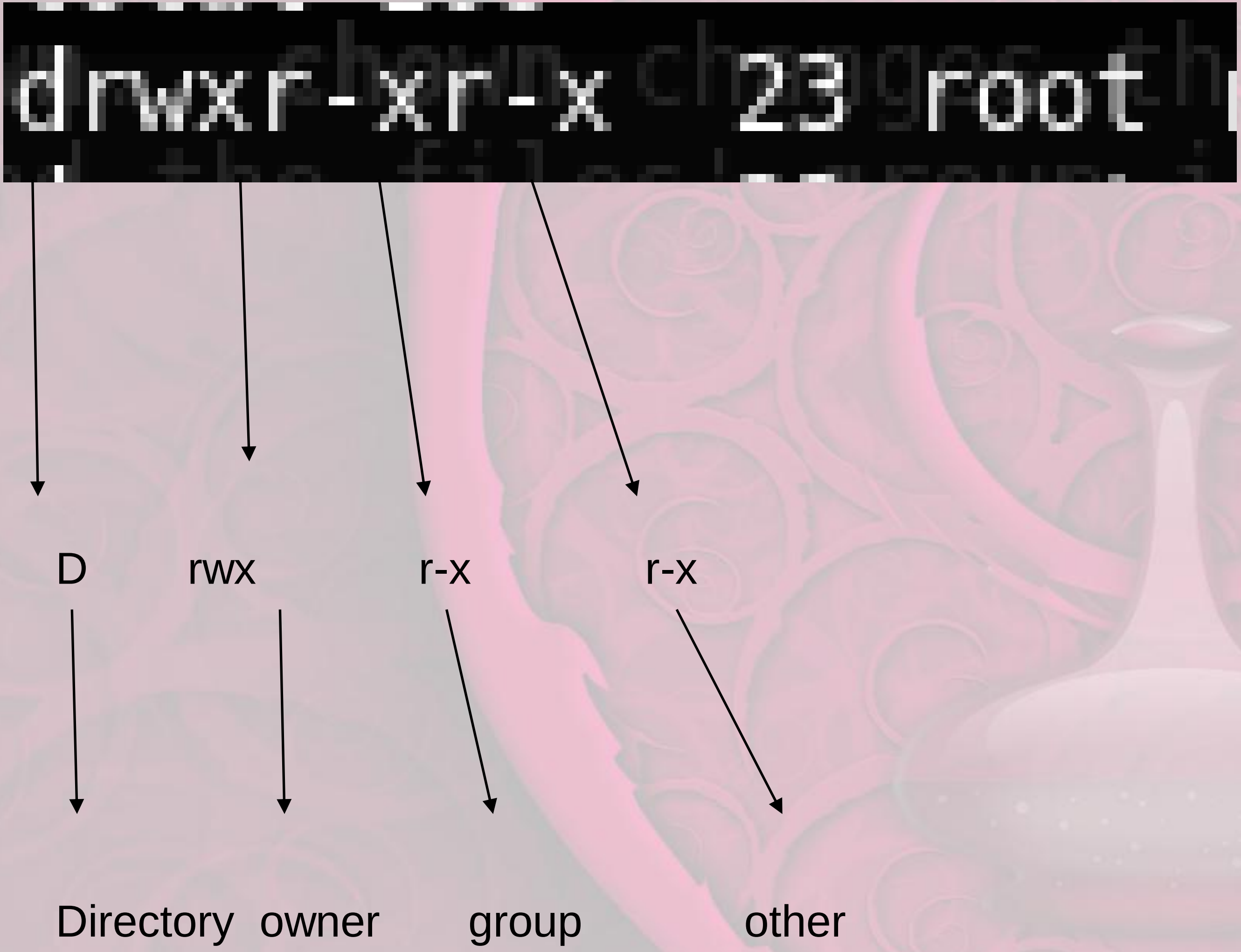
- Numarical – 4 2 1
- Symbolic – R W E

Access Mode		meaning of symbol for Files	Meaning of symbols for Directory
r	4	only display the content of the file (read permission)	can list the contents of the directory
w	2	can modify or append to the file (wire permission)	Can create or remove files of directories
x	1	can execute the file	can enter into the directory

```
EG: chmod 777 folder-or-file-name
    chmod 651 folder-or-file-name
```

6 = owner (4+2)
5 = group (4+1)
1 = other (1)

cont(1)



```
drwxr-xr-x 23 root
```

D rwx r-x r-x

Directory owner group other

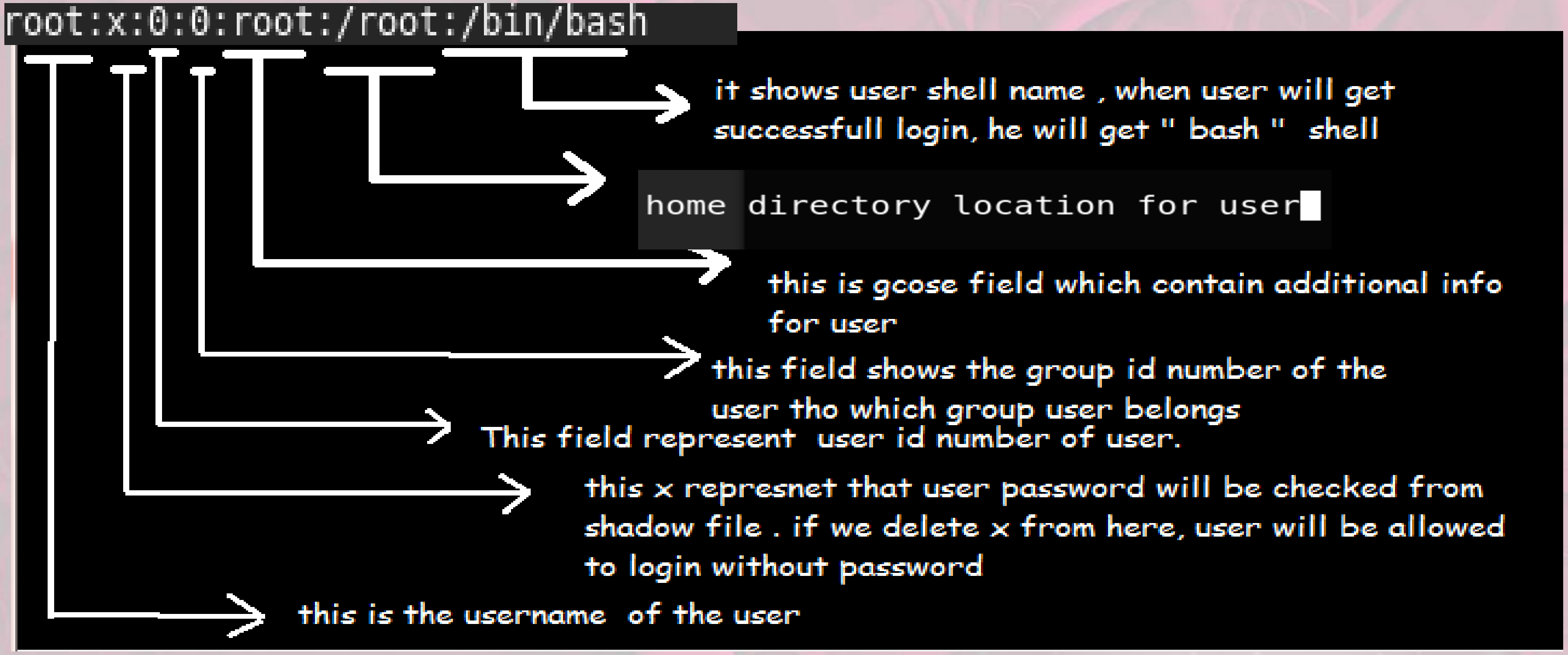
Chown – changine the owner of the file or folder

- Command usage,
- Chown user:group file/folder-name

Additional

- Cat /*etc*/passwd → to find out and modify user information
- Cat /*etc/*group/ → to find out group information

Durty Hands ;)



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

it shows user shell name , when user will get successfull login, he will get " bash " shell

home directory location for user

this is gcose field which contain additional info for user

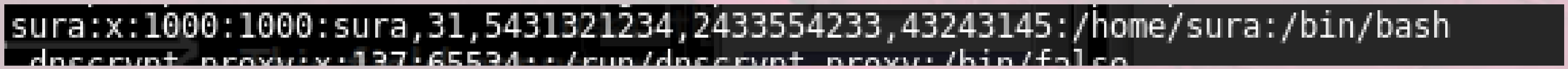
this field shows the group id number of the user tho which group user belongs

This field represent user id number of user.

this x represnet that user password will be checked from shadow file . if we delete x from here, user will be allowed to login without password

this is the username of the user

You can edit this with vi command and change as you wish.



```
sura:x:1000:1000:sura,31,5431321234,2433554233,43243145:/home/sura:/bin/bash
dnscrypt-proxy:x:137:65534::/run/dnscrypt-proxy:/bin/false
```

Thank you…!