Users and groups

Files-based

This is for users and groups that are defined by /etc/passwd and /etc/group *et al*.

Group creation

For non-system groups:

```
addgroup <group name>
```

Each user's primary group must exist before the user is created unless --user-group is specified on the useradd command.

List users in a group

```
gentent group <group name>
```

Blue Light user bl

Blue Light's standard Linux system includes user bl.

bl receives a copy of all email sent to root. This can be useful when investigating problems.

bl can use sudo to assume root privileges. This is a security weakness because bl's passwords are widely known. It must not be configured on computers with a public IP address. Its use on other computers is entrenched. As Andrey commented in FINANCESRV-227 about disabling this facility "Not okay for regular machines, since those often need to be accessed physically without difficulties".

User creation

This varies, depending on the type of user. Maybe the user will be a full Linux user; maybe the user is only required for use with samba.

TODO: collect any standard and other localised examples to augment the Savitri Bhavan examples

For a full Linux user, this example is as used at Savitri Bhavan on "archiver":

```
useradd --comment 'Vadivel' --create-home --groups archiver,sb-users --user-group vadivel
```

For a user who will only log on to samba, this example is as used at Saitri Bhavan on "archiver":

```
useradd --create-home --groups 'sb-users' --home-dir /mnt/archive/private/users/$u/ --user-group $u
```

User disabling and removal

Reference: http://www.howtoforge.com/linux_remove_users

Everything belonging to a user should be removed before the user is removed so a three stage process is required: disabling, artifact removal and finally user removal.

Disable the user

After this is done the user cannot create any further artifacts. In the examples below $u is the user name.

Disable logons and ssh sessions:

```
usermod --lock --shell /usr/sbin/nologin $u
rm -fr /home/$u/{.ssh{,2},.{s,r}hosts,.forward}
```

Kill any processes belonging to the user.

Remove any associated users from /etc/sasldb2, .htaccess files, personal MySQL and postgres users, samba ...

Edit any user's crontab, commenting out any job lines:

```
crontab -u $u -e
```

Remove or comment out any entries for the user in /etc/sudoers:

```
visudo
```

Disable associated authentications

Remove the user's membership of secondary groups:

```
usermod --groups '' $u
```

Remove the user's public key from /root/.ssh/authorized_keys if present.

Change any common passwords known by the user, for example: shared KeePass, common MySQL and postgres users, root ...

Remove the user's artifacts

Unless storage space is short, better to preserve the user's home directory (in case something is needed later or for audit) but render it inaccessible:

```
mv /home/$u{,.preserved} && chmod 000 /home/$u.preserved
```

Find any other files and directories owned by the user (the command will also search network mounted file systems) and remove or change ownership as appropriate:

```
find / -user $u
```

In case the user has a dedicated group with the same name:

```
find / -group $u
```

De-configure the user

Remove references to the user from miscellaneous configurations: /etc/samba/smb.conf(.source),

Remove the user and any dedicated group

There may be very little benefit from doing this step.

```
deluser $u
```

In case the user has a dedicated group with the same name:

```
delgroup $u
```

LDAP-based

Linux groups and users by OpenLDAP

Problem solving

User can't log on

Check ownerships of their home directory and contents, including hidden files.

Check the "shell" configured in /etc/passwd.

TODO: check if they have been locked out by special string(s) in /etc/passwd or shadow (needs research)