RHCE

BOOT CAMP

Users and Groups

redhat®

CERTIFIED

ENGINEER

USERS AND GROUPS

- Users and Groups define access to the operating system through the file permission scheme.
- Root is the super user, and the only user with special permissions
- Every user is a member of at least one group, which is called their primary group. The main purpose of this primary relationship is to define group owner of created files.
- Users can have a secondary group membership in as many groups as needed. These secondary relationships exist to broaden a user’s access to the files on the system.

CONFIG FILES

- User information is stored in two files:
  - `/etc/passwd`
  - `/etc/shadow`
- Group information is stored in one file:
  - `/etc/group`

/ETC/PASSWD

- List of user records, one per line, with columns separated by colons. Format:
- `login:x:userid:groupid:gecos:homedir:shell`
- Examples:
  - `root:x:0:0:root:/root:/bin/bash`
  - `mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/bash`

/ETC/SHADOW

- Similar colon-separated-column list of records:
- `login:password:`*password aging fields*
- Aging fields track dates for password resets, locks, etc
- Examples:
  - root:pB8msP1fCbCqc:13904:0:99999:7:::
  - nisburgh:vRoPw6a/jQsp.:14466:0:99999:7:::

/ETC/GROUP

- Same colon-separated-column list of records format
- `groupname:grouppassword:groupid:secondarymembers`
- Group passwords allow temporary management to a group, are rarely used and not set up by default
- Examples:
  - `daemon:x:2:root,bin,daemon`
  - `apache:x:48:jack,nisburgh`

MANAGEMENT

- While it is possible to edit the three files directly, it’s easier and safer to use the management commands to create, modify and delete users and groups:
  - `useradd, usermod, userdel`
  - `groupadd, groupmod, groupdel`

USERADD

- `useradd`: Add a new user to the system
- Accepts various arguments to control the settings on the user account. Most common is the -g option to specify the primary group of the user, and the -G option to list secondary group memberships. Examples:
  - `useradd lisa`
  - `useradd -g clowns -G trouble,simpson bart`

USERMOD, USERDEL

- `usermod`: Modify a user's settings. Example:
  - `usermod -G detention bart`
- `userdel`: Remove a user from the system. Main option to consider is `-r`, which tells `userdel` to remove the user's home and spool directories. Example:
  - `userdel moe`

GROUP COMMANDS

- `groupadd`: Adds a new group to the system. Example:
  - `groupadd bullies`
- `groupmod`: Mainly used to rename a group. Example:
  - `groupmod -n mktg mkg`
- `groupdel`: Remove a group. Example:
  - `groupdel microsoft`

PASSWORDS

- `passwd`: Change login password.
- Root can change the password for any user on the system
- Root can also setup password aging, allowing for timed password resets and account disabling (or use `chage`)
- `passwd` is also the preferred way to lock a user account:
  - `passwd -l mary`

PASSWORD AGING

- To set the maximum lifetime for a user's password:

  -
    ```
    passwd -x days login
    ```
- When a user's password has expired, you can set the number of days it can remain expired before disabling the account completely:

  -
    ```
    passwd -i days login
    ```

IMPORTANT USER ENVIRONMENT FILES

- `/etc/skel` default template for a newly-added user's home directory
- `/etc/profile` sets environmental variables used by all users
- `/etc/profile.d` contains scripts specific to certain rpms
- `/etc/bashrc` contains global aliases and system settings
- `~/.bashrc` contains user aliases and functions
- `~/.bash_profile` contains user environment settings and can be set to automatically start programs at login

LAB

1. Create a new group ‘`dev`’. Create a new user ‘`alice`’ as a secondary member of the ‘`dev`’ group, with a description of “`Alice from Dev`” and a default shell of ‘`/bin/csh`’. Use the `passwd` command to set a password for `alice`, then log in as `alice` and verify her access.

2. Set a maximum password lifetime of 4 weeks for the `alice` account. Look at the `passwd`, `shadow` and `group` files.

3. Configure the users `guido`, `linus`, and `richard`. Set all their passwords to “`linux`”.

4. Make these users part of the `ru` group in a secondary capacity.

5. Configure the directory `/home/linux` so that each user from the `ru` group can read, create, and modify files.

6. Configure the directory `/home/linux/work` so that each user can create and read files, but only the file’s owner can delete.

7. Use ACL’s to allow `alice`, not in `ru`, read/write access to the `work` folder and all created sub objects.

PAM

- Applications which are compiled against `libpam.so` may use PAM’s modules to customize how individual applications verify their users. Each application has its own configuration file in `/etc/pam.d`

- The first field of the configuration file indicates how the module will be used:

  - **Authentication management (`auth`)** Establishes the identity of a user.

  - **Account management (`account`)** Allows or denies access to the account.

  - **Password management (`password`)** Enforces password management policies.

  - **Session management (`session`)** Starts, stops, and records each session.

PAM

- The second field of the configuration file indicates the effect that the module will have on the application:
  - `Required` If this module fails, access will not be granted, but all other modules will still be run.
  - `Requisite` If this module fails, access will not be granted and no other modules will be run.
  - `Sufficient` If this module succeeds, access will be granted and no other modules will be run.
  - `Optional` The result of this module is ignored.

PAM

- The third field of the configuration file indicates the name of the actual PAM module to be used for the config line in question.

- Side note:

  - The config file `system-auth` is a collection of many PAM modules commonly used by many authentication services. You will see it included by many of the other configuration files. *Do not modify this file directly*.

PAM

- **`pam_unix`** Authenticates users by UNIX password
- **`pam_securetty`** Only allows root to log in from secure terminals listed in `/etc/securetty`
- **`pam_nologin`** Will not allow any non-root user to login if `/etc/nologin` exists
- **`pam_time`** Can be configured to allow/deny access based on the system time
- Helpful PAM documentation can be found in:
  - `/usr/share/doc/pam-<version>`

LAB

1. Using PAM, prevent “`guido`” from being able to login on Virtual Console 2. `Guido` should still be able to login elsewhere.

   Hint: Configure the `pam_access` module.

2. Set up the `pam_time` module to restrict `linus` so he can only login between 8am and 5pm Monday through Friday, and block out all non-root users from logging in midnight to 2am Sundays for a maintenance period.

NIS

- NIS Servers can be configured to centrally manage system and account information. These servers can share the contents of `/etc/passwd`, `/etc/shadow`, `/etc/group`, and several other files among any number of clients.
- To configure a client, you must install the `ypbind` and `rpcbind` RPMs, and then you can run `system-config-authentication`.
- This command will make the proper entries in:
  - `/etc/sysconfig/network`
  - `/etc/yp.conf`
  - `/etc/nsswitch.conf`
  - `/etc/pam.d/system-auth`

LAB

1. Configure your server to authenticate against `server1.example.com` using NIS.

2. You should then be able to log in to your server as `station#` (where # is your station number) with the password: `station#`

LDAP

- LDAP Servers can also be configured to centrally manage system and account information. LDAP is much more secure and flexible than a default NIS configuration, and as such is becoming much more popular.

- To configure a client, you must install the `nss-pam-ldap` and `openldap` RPMs, and then you can run **`system-config-authentication`**.

- This command will make the proper entries in:
  - `/etc/ldap.conf`
  - `/etc/openldap/ldap.conf`
  - `/etc/nsswitch.conf`
  - `/etc/pam.d/system-auth`

KERBEROS

- Kerberos is a secure authentication method which never needs to send passwords over the network, except in the case of changing a password, which is handled with strong encryption.
- All that is needed for a client to set up Kerberos authentication is:
  - Realm
  - KDC - Key Distribution Center
  - Admin Server (often same server as KDC)

LAB

1. Disable NIS authentication and verify you can no longer authenticate as `station#`.

2. Configure your server to authenticate against `server1.example.com` using LDAP and Kerberos passwords. KDC/Admin server: server1.example.com, realm: EXAMPLE.COM

3. You should then be able to log in to your server as `station#` (where # is your station number) with the password: `station#`

```
slideshow.end();
```