RHCE

BOOT CAMP

PAM, Kerberos and Software
RAID

redhat®
CERTIFIED
ENGINEER

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.

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#`

SOFTWARE RAID

- Software RAID can all be configured, monitored, and modified with the `mdadm` command.

- To create a RAID array, you can run the following command:

  -
    ```
    mdadm -C <RAID dev> -l <LEVEL> -n <# DISKS> <partitions>
    ```

- To verify the RAID array, use either of the following commands:

  -
    ```
    mdadm --detail <RAID device>
    ```

  -
    ```
    cat /proc/mdstat
    ```

LAB

1. Create a RAID-5 array on your machine consisting of:
    - 4 partitions
    - each 512MiB in size
    - one of which should be reserved for use as a hot spare
2. Format this array with ext4 and mount it with support for user quotas so that it will persist across reboots.

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