Managing The Linux Boot Process

Creating an initrd Image

Scenario

```
As part of your server infrastructure, you plan on having some systems boot from
an NFS share. The kernel in the deployed systems doesn't have an NFS module.
Without this, your systems cannot mount an NFS share as the root file system. So,
you need to create a new initrd image so that the kernel can successfully mount
the share. First, however, you'll establish a baseline image that other images can
build off of.
```

Objectives

- Completing this activity will help you to use content examples from the following syllabus objectives:
  - 1.1 Explain the Linux boot process
  - 3.3 Summarize security best practices in a Linux environment

1. Create a new `initrd` image.

   - Log in as `student01` with `Pa22w0rd` as the password.
   - Enter `uname -r` to identify the current kernel.
   - Enter `sudo mkinitrd -v /boot/initrd-$(uname -r).img $(uname -r)`
     - `$(uname -r)` substitutes the name of the kernel in this command.
   - Examine the verbose output from mkinitrd noting the various kernel modules that are included in the initrd image by default.
   - Enter `ls -l /boot` and verify that your new initrd image was created.
   - The image should be named `initrd-.img` and should have been last modified on today's date.

2. Create an `initrd` image with an NFS module installed.

   - Enter `sudo mkinitrd -v --with=nfsv4 /boot/initrd-$(uname -r)-nfs.img $(uname -r)`
     - Check your syntax before you hit enter. Also, do not forget that Tab completion can make your life a great deal easier.
   - Enter `ls -l /boot` and verify your new NFS image was created.
   - Examine the file sizes for both `initrd` images (the base image and the NFS image) and verify that the NFS image is larger. This suggests that the additional NFS module was loaded into the image, as intended.

```
*** Including module: microcode_ctl-fw_dir_override ***
  microcode_ctl module: mangling fw_dir
    microcode_ctl: reset fw_dir to "/lib/firmware/updates /lib/firmware"
    microcode_ctl: processing data directory  "/usr/share/microcode_ctl/ucode_with_caveats/intel"...
intel: model '', path ' intel-ucode/*', kvers ''
intel: blacklist ''
    microcode_ctl: intel: Host-Only mode is enabled and ucode name does not match the expected one, skipping caveat ("06-4f-01"
not in " intel-ucode/*")
    microcode_ctl: processing data directory  "/usr/share/microcode_ctl/ucode_with_caveats/intel-06-4f-01"...
intel-06-4f-01: model 'GenuineIntel 06-4f-01', path ' intel-ucode/06-4f-01', kvers ' 4.17.0 3.10.0-894 3.10.0-862.6.1 3.10.0-693
.35.1 3.10.0-514.52.1 3.10.0-327.70.1 2.6.32-754.1.1 2.6.32-573.58.1 2.6.32-504.71.1 2.6.32-431.90.1 2.6.32-358.90.1'
intel-06-4f-01: blacklist ''
intel-06-4f-01: caveat is disabled in configuration
    microcode_ctl: kernel version "3.10.0-957.el7.x86_64" failed early load check for "intel-06-4f-01", skipping
    microcode_ctl: final fw_dir: "/lib/firmware/updates /lib/firmware"
*** Including module: shutdown ***
*** Including modules done ***
*** Installing kernel module dependencies and firmware ***
*** Installing kernel module dependencies and firmware done ***
*** Resolving executable dependencies ***
*** Resolving executable dependencies done***
*** Hardlinking files ***
*** Hardlinking files done ***
*** Stripping files ***
*** Stripping files done ***
*** Generating early-microcode cpio image contents ***
*** Constructing GenuineIntel.bin ****
*** No early-microcode cpio image needed ***
*** Store current command line parameters ***
*** Creating image file ***
*** Creating image file done ***
*** Creating initramfs image file '/boot/initrd-3.10.0-957.el7.x86_64-nfs.img' done ***
[student01@localhost ~]$ ls -l /boot
total 195940
-rw-r--r--. 1 root root   151918 Nov  8  2018 config-3.10.0-957.el7.x86_64
drwx------. 3 root root    16384 Dec 31  1969 efi
drwxr-xr-x. 2 root root       27 Jan 11  2019 grub
drwx------. 2 root root       21 Jan 11  2019 grub2
-rw-------. 1 root root 73996789 Jan 11  2019 initramfs-0-rescue-f6f7a37d52454c709b242c2d60ac77f9.img
-rw-------. 1 root root 32007830 Jan 11  2019 initramfs-3.10.0-957.el7.x86_64.img
-rw-------. 1 root root 13646428 Jan 11  2019 initramfs-3.10.0-957.el7.x86_64kdump.img
-rw-------. 1 root root 31594808 May 16 11:16 initrd-3.10.0-957.el7.x86_64.img
-rw-------. 1 root root 32063922 May 16 11:18 initrd-3.10.0-957.el7.x86_64-nfs.img
-rw-r--r--. 1 root root   314036 Nov  8  2018 symvers-3.10.0-957.el7.x86_64.gz
-rw-------. 1 root root  3543471 Nov  8  2018 System.map-3.10.0-957.el7.x86_64
-rwxr-xr-x. 1 root root  6639904 Jan 11  2019 vmlinuz-0-rescue-f6f7a37d52454c709b242c2d60ac77f9
-rwxr-xr-x. 1 root root  6639904 Nov  8  2018 vmlinuz-3.10.0-957.el7.x86_64
[student01@localhost ~]$
```

Configuring GRUB 2

- Scenario

> Some of your fellow administrators are claiming that their Linux servers aren't booting properly. You are assigned to the task of troubleshooting these issues. You find that someone has modified the settings in the boot loader because there is no password protection. After correcting the boot configuration, you decide to protect GRUB 2 with a password so that only authorized users can modify it.

Objectives

- Completing this activity will help you to use content examples from the following syllabus objectives:
  - 1.1 Explain the Linux boot process

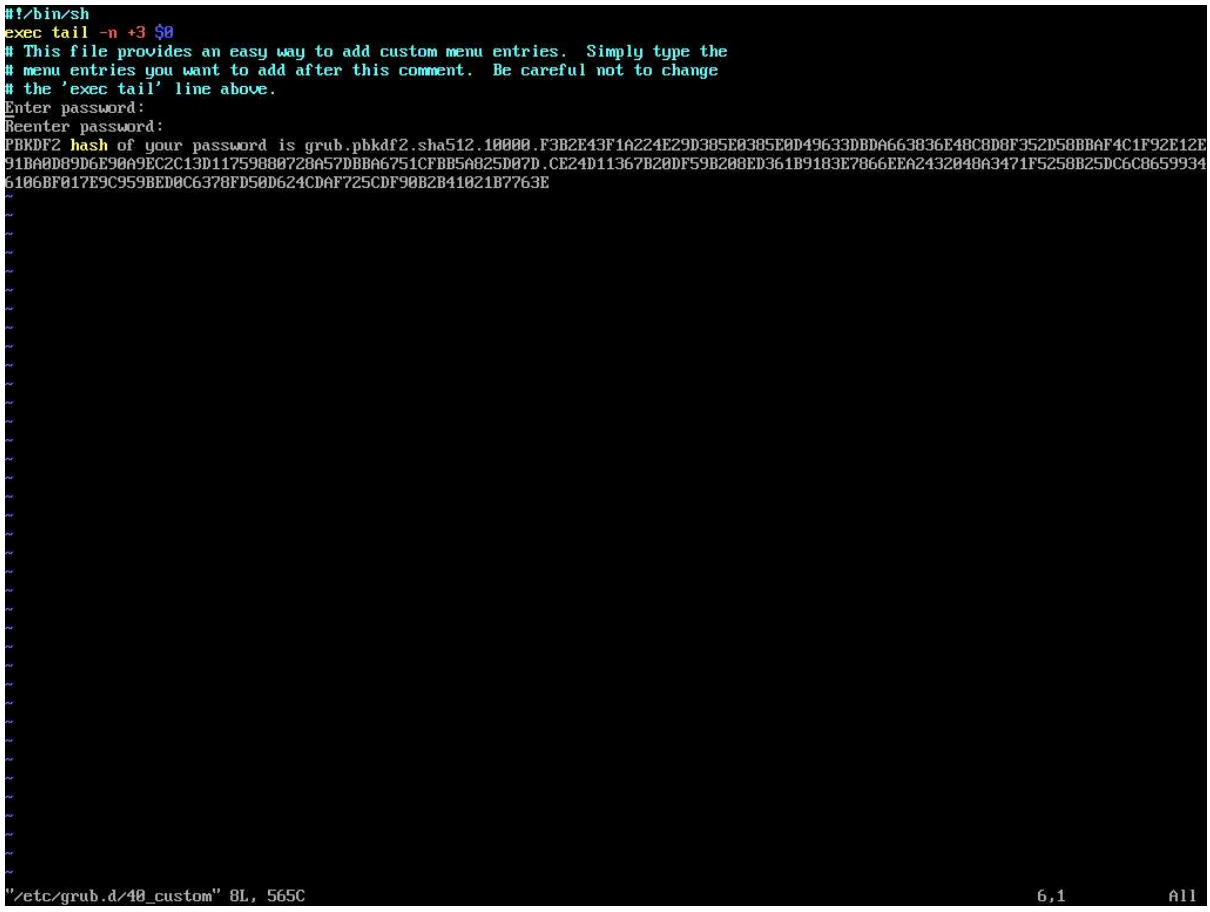
1. Verify that GRUB 2 is installed.
   - Enter `sudo ls -l /boot/efi/EFI/centos` to display the contents of the directory.
   - Verify that `grub.cfg` exists in this directory.
   - The presence of this file usually indicates that `GRUB 2` is successfully installed on the EFI system partition.
   - Enter `sudo cat /boot/efi/EFI/centos/grub.cfg` and verify that the configuration file is populated.

```
[student01@localhost modprobe.d]$ sudo cat /boot/efi/EFI/centos/grub.cfg
#
# DO NOT EDIT THIS FILE
#
# It is automatically generated by grub2-mkconfig using templates
# from /etc/grub.d and settings from /etc/default/grub
#

### BEGIN /etc/grub.d/00_header ###
set pager=1

if [ -s $prefix/grubenv ]; then
  load_env
fi
```

2. Create a password to lock the `GRUB 2` configuration with.
    - Enter `sudo grub2-mkpasswd-pbkdf2 | sudo tee -a /etc/grub.d/40_custom`
    - You're redirecting the output to the custom configuration file. You'll clean up this file shortly.
    - Enter `Pa22w0rd` as the password.
    - Reenter the same password.
    - Verify that the `PBKDF2` password is generated.
    - `PBKDF2` uses a cryptographic technique called hashing to protect the password in storage.
3. Adjust the custom `GRUB 2` configuration file to require your password.
    - Using `sudo`, open `/etc/grub.d/40_custom` in the text editor of your choice.



    - Move the cursor to the line that says "`Enter password:`" and cut this entire line.
    - Cut the line after it that shows the `reenter password prompt`.

- From the "`PBKDF2`" line, delete the string of text that says "`PBKDF2 hash of your password is`".
- On the same line, replace the text you deleted with `password_pbkdf2 student01`

- Insert a new line above this that says:
- `set superusers="student01"`

- Save and close the file.

4. Update the main GRUB 2 configuration file to apply your changes.

   - Enter `sudo grub2-mkconfig -o /boot/efi/EFI/centos/grub.cfg` to cause grub to create a new configuration file.
   - There may be an I/O error message for the `fd0` (floppy drive) device. This will not affect the boot process. Verify that no other errors are returned and that the "`done`" message is displayed.
   - This indicates that the new `GRUB 2` configuration file has been generated successfully.

5. Test the password from the GRUB 2 boot menu.

   - Enter `reboot` to restart the server.
   - On the `GRUB 2` boot menu screen, press `Esc` to stop the default selection timer.
   - Press `e` to edit the GRUB 2 configuration.
   - At the `Enter username` prompt, enter the account name `student01`.
     - Input your user name and password very carefully, as you will be unable to edit any mistakes.
   - At the `Enter password` prompt, enter `Pa22w0rd`
   - Verify that you can see the `GRUB 2` configuration on the screen.

- Press `Esc` to exit editing mode.
- Press `Enter` to boot back into the default selection.
- Log back in as your student account.