Lab X3: Exploring the Login Process (Extra Credit)

The purpose of this lab is to explore the various processes that take place when a user logs in to a UNIX/Linux machine. To be sure, the login process can vary among the different varieties of UNIX, but there is more in common than not. Three main processes are run during a login:

1. getty - opens a terminal device and prompts for a login name
2. login - obtains the user password, performs authentication, and launches a shell
3. shell - runs through a series of configuration scripts establishing the users environment.

Supplies

- VMware Server 1.05 or higher
- Benji VM (CentOS 5)

Procedure

You will be creating a text file named labX3 to turn in. This file will have your answers to the questions being asked below as well as include some of the configuration files you make or modify.

Forum

If you get stuck on one of the steps below don't beat your head against the wall. Use the forum to ask for assistance or post any valuable tips and hints once you have finished. Forum is at: http://simms-teach.com/forum/viewforum.php?f=13

Part One: The getty Process

The getty program is launched by the init program at startup and whenever the getty process is terminated (respawn). getty uses the configuration file, */etc/issue* to advertise the system to the user.

1. Create your labX3 text file and add your name, assignment, and class information to the top. You will be **recording the answers to the questions (Q??) below** in this file.
2. Log on as *root* on terminal tty1 (<Ctrl><Alt>F1)
   Note the message above the **login:** prompt, and note any other messages you receive prior to arriving at the shell prompt.
3. View the file */etc/issue*
   Q1 How does /etc/issue compare to what you saw on the screen prior to logging in?
4. View the manual page for **mingetty** to discover the meaning of the various escape characters that can be used in the issue file.
5. Back up the existing */etc/issue* file to */etc/issue.00*, and replace the architecture type with the tty (line) number, and add a third line that gives the current date. **Q2** Record the contents of your modified /etc/issue here.
6. Log off.
   Note: the other virtual ttys will not change until they are respawned, i.e. after you log on and off.
7. Log back on and list all the **getty** processes that are running
   ps -e | grep getty
   Note: the **getty** program that Linux uses on the console is: mingetty
   Q3 Why is there no getty program on tty1?
8. Jot down the PID number of the mingetty running on tty3
9. Switch screen to virtual terminal tty3 (<Ctrl><Alt>F3), and type root followed by the enter key, but don't type the password.
10. Switch back to tty1 and rerun the ps command from above.
    Note that mingetty is no longer running on tty3.
11. Now grep the process listing for the login processes:
    ps -e | grep login
    Q4 How does the PID number of the login process on tty3 compare to the getty PID you noted above?
    Q5 How does one process become another process without changing its PID?
12. After a short period of time, the login process on tty3 will timeout, and a new mingetty process will be spawned by init.

Part Two: The Login Process

In this procedure you will observe the steps taken by the login process in authenticating the user and in posting such messages as the last time the user logged in, the message of the day (motd), and the ubiquitous, "You have mail" message.

1. View the manual page for **login** to see how login can be configured. In this section we will be using .hushlogin, /etc/nologin, /etc/motd, and /etc/securetty files mentioned on this man page.
2. Make a backup of the */etc/motd* file to */etc/motd.00* and add a message of the day of your own choosing to that file. **Q6** Record the actual contents now of your modified /etc/motd file here.
3. Send a mail message from root to root, but don't read the message you send.
4. Log off the system and log back in again as root, noting the messages the login program gives you. Note: **Login** starts with the Password: prompt.
5. Now that you are logged in and have noted the last time you logged in, the message of the day, and the fact that you have new mail, create the hidden file *.hushlogin* in root's home directory:
   touch .hushlogin

6. Log off and back in again, and note that you don't have to put up with noisy announcements. This is yet another way that UNIX empowers the user.
7. Let's look at the authentication part of login. Authentication is handled by the PAM modules in */etc/pam.d*. (PAM = Pluggable Authentication Modules).
   cd /etc/pam.d
8. Display the contents of the *login* file in this directory. We are going to investigate the initial lines dealing with authentication.
9. Comment the second **auth** line in the file by inserting a comment character (#) in front of the line:
   #auth include system-auth
10. After saving the file's contents, log off and back on again.
    Q7 Record the actual contents of your modified /etc/pam.d/login file here.
    Q8 How did this modification effect the login process?
11. Remove the comment character from second auth line in the file */etc/pam.d/login*.
12. Change directory to */etc* and edit the file */etc/securetty*.
13. Place a comment character (#) in front of the tty1 line in this file:
    #tty1
    Save the file, Log off, and log back on again. **Q9** Record the contents of your modified /etc/securetty here. **Q10** What happened with this modification?
14. **tty1** is no longer considered secure for root logins! Don't worry, you can log in on tty2 and fix the problem by removing the comment from the securetty file.
15. Create the file */etc/nologin* with text such as the following:
    echo "No more logins allowed until further notice. -SA" > /etc/nologin
16. Switch to a different tty, and login as a regular user, using your personal account.
    Q11 Can a non-root user login? **Q12** Can root login? **Q13** Record the contents of your /etc/nologin file here.
17. Login as root and remove the */etc/nologin* file.
18. The last task of the login process is to launch the user's shell as defined in the */etc/passwd* file. **Q14** What are the values of your $USER, $HOME, and $SHELL ?
    Q15 Do they correspond to the entry in /etc/passwd?

Part Three: The Shell Process

In this procedure you will observe how the user's shell environment is established from both a system perspective as well as incorporating user customization.

1. Your shell is a child process of the login process that came from the getty running on your tty. Observe the PID and PPID of your shell and it's parent:
   ps -f
   ps -f PPID # where PPID is the PPID number of your bash shell
2. All login shells execute the */etc/profile* shell script to initialize system-wide environment variables. Display this file, and note the variables that are initialized. Note how the root account gets additional paths added to its PATH variable. Near the bottom of the file you may be able to pick out how the shell scripts in the directory *./etc/profile.d* get executed in turn. List the scripts in the profile.d directory:
   ls /etc/profile.d
3. After system initialization comes user customization. In the case of the bash shell, the hidden file, *.bash_profile* in the user's home directory is executed.
   This in turn, makes calls to other scripts such as *.bashrc*. Review these files.
4. Below are six events that take place during the initialization of the shell's environment. Your task is to determine the actual order that the events occur when a user logs in. In other words, you must find where and when these commands are

executed. **The order they are listed below is random**. **Q16** What is the correct order of the following events?

- PS1 variable gets initialized specifying what shell prompt the user gets.
- The ls command is aliased to show files in different colors
- The user's umask is set, e.g. umask 002
- The location of the user's system mailbox is determined
- The rm command is aliased to "rm -i"
- The user's $HOME/bin directory is added to the PATH variable

To turn in

Review the contents of the labX3 text file. Did you answer each of the questions? Are your answers clearly marked to show the question they are answering? You can use the spell command to find misspellings (**spell labX3**). It should look something like:

```
Benji Simms
Lab X3 (Logging in)
CIS 191B Fall 2008

A1)  blah, blah, blah
A2)  blah, blah, blah
  .
  .
  .
A16) blah, blah, blah
```

Now copy the labX3 file to the CIS191 account on opus.cabrillo.edu using the following command:

`scp labX3 cis191@opus.cabrillo.edu:labX3.`*`logname`*

where *logname* is your Opus username.

Cleanup

Some files you may have already cleaned up. Please make sure you have taken care of the following tasks:

1. mv /etc/issue.00 /etc/issue
2. mv /etc/motd.00 /etc/motd
3. rm /root/.hushlogin
4. rm /etc/nologin
5. Restore /etc/pam.d/login to original
6. Restore /etc/securetty to original

Grading rubric (25 points)

(2 points) for submitting before Thanksgiving
(2 points) for professional quality labX3 (readability, spelling, grammar, formatting, clear and concise answers, organized, correctly submitted as described above)
(Up to 15 points) for correctly answering Q1- Q15
(Up to 6 points) for correctly answering Q16