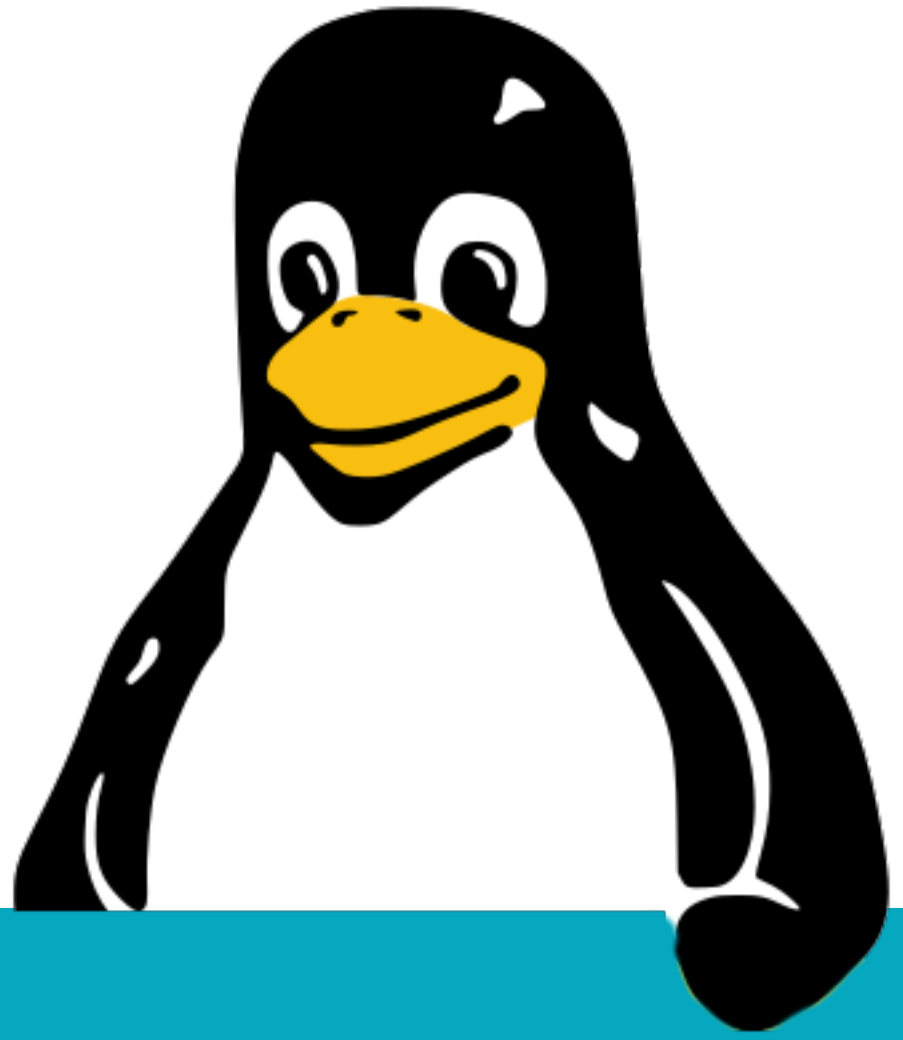
Linux, day 5



Objectives covered

Objective	Summary	Boek
3.1	Given a scenario, create simple shell scripts to automate common tasks.	3 4 25

LAB: Shell scripting: Flow

Assignment

- Write a shell script which:
  - Checks if it’s run as root; if not, "*exit 1*".
  - Reads a command from passed parameters.
    - This command is either "create" or "remove".
  - Asks how many users to create/remove (keep it <20).
  - Creates or removes "user1", "user2", "user3" etc.
  - Uses HEREDOC to make "welcome.txt" in their homedir.

Assignment

- If you run it, this looks like:

```
$ sudo /tmp/dummy-users.sh create
How many?
5
```

Solution

- I have a sample solution on Github.
- It's here, in the Lesson 004 directory.

LABS: Repeats

Repeat exercises

- If you didn’t get things done last week, start here.

- All exercises are also available as text, via Teams.

Assignment 1, from last class

- Create a script which:
  - Runs ;)
  - Reads a name from the first, passed parameter.
  - Asks for a greeting interactively.
  - Outputs a greeting to the name.

Assignment 2, from last class

- Find all files with passwd in their name?
- Find all files, literally called "passwd"?
- Find all world-writable files?

- Use a HEREDOC to SSH to localhost and run:
  - touch /tmp/foobar
  - ls /tmp

Assignment 3

- Write a shell script which:
  - Checks if it’s run as root; if not, "*exit 1*".
  - Reads a command from passed parameters.
    - This command is either "create" or "remove".
  - Asks for a number (for now, keep it <20).
  - Either creates or removes "user1", "user2", "user3" etc.
  - Uses HEREDOC to make "welcome.txt" in their homedir.

LABS: New assignments

New exercises

- Now let’s try a few other things.

- All exercises are also available as text, via Teams.

Assignment 4

- Remember greet.sh? Change it so ...
  - The desired greeting is stored in a file.
  - The script checks if the file exists.
    - If it does, read the greeting from the file.
  - Add a flag, like -r to allow a "reset" of the greeting.

Assignment 5

- Make a shell script which:
  - Reads a list of hostnames (or IPs) from a file.
  - SSHs to each of these hosts.
  - And on that host, finds files with the *setuid* flag.
- Output each list to ~*/setuid/${NameOfTarget}.txt*
  - On the source host!

Assignment 6

- Make a script that:
  - Uses */etc/passwd* to find "real" users.
  - Takes their login name,
  - And uses it to generate an email address.
  - e.g. tess@mydomain.local.
  - Store these in a TAB-delimited file, as "*login email*"
- Hint: use assignment 3 to create the test-users.

Assignment 7

- Make a backup copy of */etc/login.defs*
- Using the command line, NOT an text editor,
  - Change the setting of UMASK from 022 to 027.
- Make sure to <u>only</u> change the one correct line.

Really advanced

- Go over the scripts you've made so far.
  - See where you "program defensively".
  - Check user input before using it.
  - Check command output and status.
  - "*Don't trust that it works, test it to be sure!*".

Optional: Bandit

- You've made it very far!
- Have you already played the "Bandit" wargame?
  - https://overthewire.org/wargames/bandit/
- Alternatively, want to try simulated "work"?
  - https://kodekloud-engineer.com

Closing

Homework

- Reading:
  - Chapter 2
  - Chapter 6, p. 159-172
  - Chapter 7

- Go do:
  - Unfinished labs, or ping Tess for the extra task!

Reference materials

Resources

- An excellent, in-depth study guide for shell scripting
- Why you shouldn’t parse the output of "ls"
- OverTheWire "Bandit" wargame
- Arithmetic in Bash
- In-depth look at Bash maths
- Shell Scripting tutorial for beginners (YouTube)