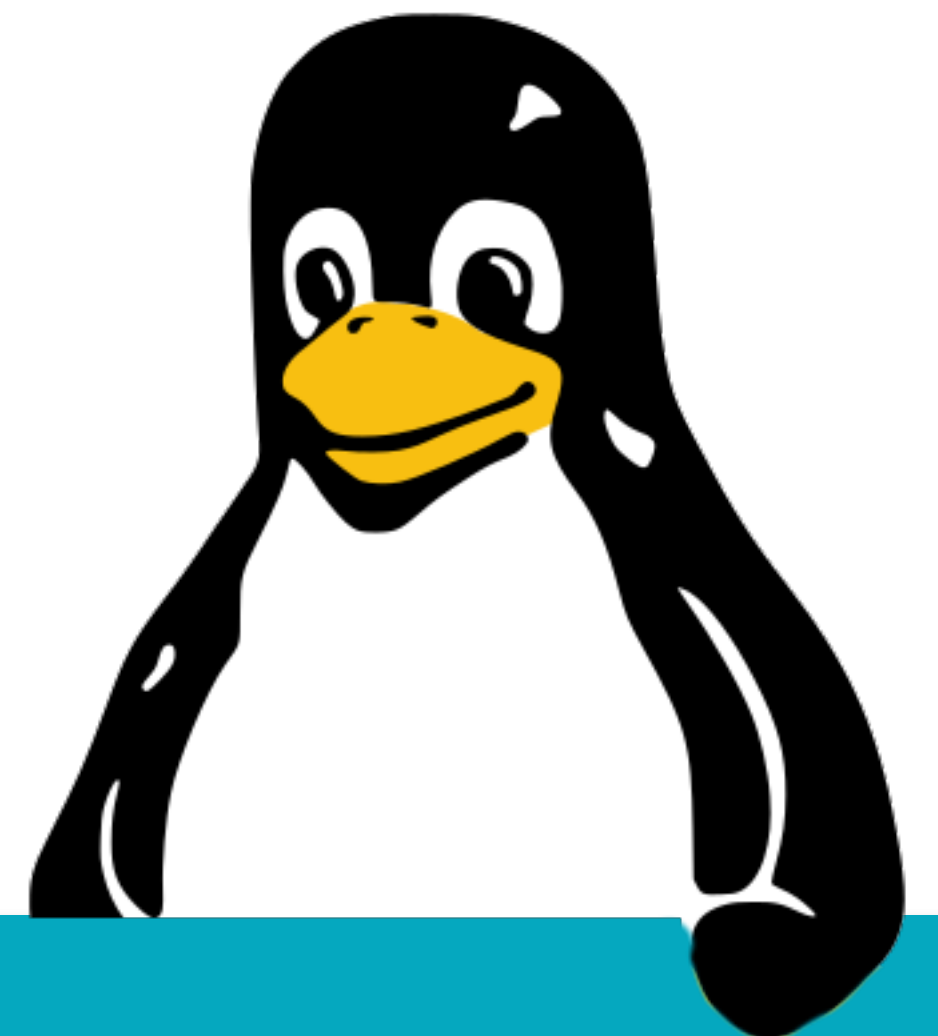


Linux, day 5

This lab is licensed under Creative Commons BY-NC-SA 4.0.
<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.en>

You are free to share and adapt, but NOT for commercial purposes and you must attribute the source and share your own adaptations under the same license.



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, from last class

- 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.

Assignment 4, our homework

- Create a shell script to "*ping sweep*" a network.
 - It should ask for the first three octets of an IP.
 - Then it should cycle through all 254 addresses.
 - e.g. *./pingsweep 192.168.10.0 24*

LABS: New assignments

New exercises

- Now let's try a few other things.
- All exercises are also available as text, via Teams.

Assignment 1, evolved

- Remember greet.sh? Change it so ...
 - The desired greeting is stored in a file.
 - The script checks if the file exists.
 - If it does, no need to ask the user for the greeting!
 - Add a flag, like -r to allow a "reset" of the greeting.

Assignment 5

- Make a shell script which:
 - Takes a list of target hostnames (or IPs)
 - SSHs to each of these hosts
 - And on that host, looks for *setuid* files.
- Output each list to *~/setuid/targethost.txt*
 - On the source host!

Assignment 6

- Take the "ping sweep" script from our homework.
 - Adjust it so, for each found "up" host,
 - It will test one or more specified ports.
 - You can start by hard-coding the ports in a var.
- Hint: You can test a port with Bash net redirection.
 - *timeout 3 bash -c 'echo 1 >/dev/tcp/192.168.200.20/22'*

Assignment 7

- 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 8

- 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 only 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. 148-175
 - 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)