

# Linux, day 5



# LABS: Repeats

# Repeat exercises

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

# 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

- Go do:
  - Create a shell script to "*ping sweep*" a network.
    - The script should ask for a base address and netmask,
    - So it can calculate the addresses to ping.
    - e.g. *./pingsweep 192.168.10.0 24*

# SSH keys and ssh-agent



# Setup

- Ensure that you have two Linux VMs.
- And that you have an account on both.
- You can use this for assignment 5.

# Assignment

- Double-check that SSHd runs on both servers.
- Generate a new key pair on one of the accounts.
  - Make it type ECDSA, with a password.
  - Setup its pub.key for authentication on the other VM.
  - Test your SSH key authentication.

# Assignment

- Start "*eval \$(ssh-agent)*".
- Add / load the private key you generated into the running "*ssh-agent*", with the "*ssh-add*" command.
  - This should ask your password once.
- Try SSH-ing to the other VM again.
  - This should not ask your password.

# Assignment

- Reconfigure "*sshd\_config*" on one of the VMs,
  - So it will only allow group "*sshusers*" to login.
- Give your own account the new group "*sshusers*"
- Restart the SSH daemon and test that you can login.
  - Also make sure that another user cannot.

# LABS: New assignments

# New exercises

- Now let's try a few other things.

# 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](mailto: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

# Reference materials

# Resources

- [An excellent, in-depth study guide for shell scripting](#)
- [Why you shouldn't parse the output of "ls"](#)
- [OverTheWire "Bandit" wargame](#)