Urmica Alaina Chodavarapu
Unix
Journal 2
2024-11-24

Week 2

We fixed the problem we had with the manual configuration.

These are the updated steps:

1. We focused on Raspberry Pi OS: instead of doing it with Debian we only focused on Raspberry Pi OS.

2. We added the command dpkg -l | grep motion to verify if motion is installed. This helps confirm the installation before proceeding.
    - sudo apt update: refreshed the package list.
    - sudo apt upgrade -y: ensures all installed packages are up to date.
    - sudo apt install motion: installs the motion software necessary for motion detection.

3. We used sudo raspi-config to access the interfacing option => camera => enable and reboot for the settings to take effect.

4. Since we are using a USB webcam, we installed fswebcam which is a command line to access image capture:
    - sudo apt install fswebcam
    - fswebcam test.jpg

5. Added the command to start Motion with its configuration file:
    - motion -c /etc/motion/motion.conf

6. Allows the user and group to access the file without any errors
    - sudo chown csuser:csuser /var/log/motion

7. To see all the processes related to Motion:
    - ps aux | grep motion

8. Find the IP address to set up the camera(as in access it through a website like an example I provided)
    - hostname -I
    - To access the video we need to browse http://(Ip address):8081

9. Enable motion detection:
    - sudo nano /etc/motion/motion.conf
    - In there: We adjusted the threshold sensitivity for better motion detection. The higher the value, the more sensitive it is. We also added event_gap so it saves the footage every minute.

10. To save the images we created a directory:
    - mkdir /home/pi/videos

11. We made a directory for storing the motion detection and images. -p checks for the parent directory and creates if it's not there.
    - sudo mkdir -p /var/log/motion

12. To start the video as soon the video has started we used the commands:
    - sudo systemctl enable motion
    - sudo systemctl start motion

13. To change the password:
    - passwd

So in conclusion, we didn't use FTP but we successfully set up the security camera manually using Raspberry Pi OS.(still not done we plan on adding more stuff as we go on)

The second method:

We moved on to the next method which is the MotionEyeOS method:

1. We planned to try MotionEyeOS, which is a pre-configured operation system designed specifically for camera setups.
2. First, we downloaded Raspberry Pi Imager, which we used to download MotionEyeOs into the microSD card.
3. After installing it into the motionEyeOS version, the Raspberry Pi version was not compatible.
4. We spent about 2 hours trying to fix the problem in multiple ways. We tried to download it multiple times and test it on different Raspberry Pi and tried to upgrade the Raspberry Pi using this YouTube video: How to Upgrade Raspberry Pi.
5. With more trying, we learned that MotionEyeOS is no longer actively maintained with the confirmation from the teacher, who helped us verify that the OS wasn't compatible.

6. After class, we both decided to explore any other methods possible. I tried Google, and YouTube to find any alternative methods.
7. I tried to look into any other methods possible I found ZoneMinder, Rpi-Cam-Web-Interface, and Kerberos.io. I followed half of the first video and when I tried to run it with Raspberry Pi, they all didn't work great, here are the reasons:
    - ZoneMinder: uses too many resources, which slows down the Raspberry pi.
    - Rpi-Cam-Web_interface: has very few features there is no motion detection or easy multi-camera management.
    - Kerberos.io: so hard to set up even with multiple websites and YouTube videos.
8. After trying multiple ways we finally managed to get it to work with the Raspberry Pi security Camera Network using MotionEye, it worked but we haven't explored it in depth yet but it is similar to MotionEyeOS. We choose things because of :
    - It's simple to install and configure, mainly on Raspberry Pi.
    - It runs smoothly without too much of the pi's power.
    - It is easy to understand and use.

The steps we did to make it work:

1. Update Raspberry Pi:
    - sudo apt update.
    - sudo apt upgrade

2. Installing MotionEye, we installed the entire package like the ca-certificates which will secure communication over HTTPS, curl which is used to transfer data for downloading files, pyhton3, and python-dev required to run and build python-based tools like MotionEye, libcurl4-openssl-dev, and libssl-dev is for secure data transfer and gcc is for compiling necessary program components:
    - sudo apt --no-install-recommends install ca-certificates curl python3 python3-dev libcurl4-openssl-dev gcc libssl-dev

3. Installing Python Pip if it's not installed, this is for MotionEye and dependencies:
    - pip --version
    - sudo apt install python3-pip -y

4. To fix any externally managed environment errors, we run to avoid any errors during installation:
    - sudo python3 -m pip config set global.break-system-packages true

5. Install MotionEye using Python Pip:

- sudo python3 -m pip install --pre MotionEye

6. Setting upMotionEye on the system, preparing it to manage cameras and settings:
   - sudo motioneye_init

7. To access the web interface we need to use our IP address I mentioned before here is how:
   - hostname -I
   - Open the browser and go to http”//(IPaddress):8765. Log in as admin with no password, then change it in settings.

8. Since we are using a USB camera (the first option) we used UVC under camera type (the last 3).

MotionEye is easy to use and great for managing multiple cameras. It has a user-friendly dashboard, making it simple to control all cameras in one place. It’s secure and supports remote monitoring, making it a good choice for modern setups. Motion, on the other hand, uses fewer resources, so it works better on older Raspberry Pi models. It’s good for basic setups but requires more manual configuration and doesn’t have an all-in-one dashboard like MotionEye.

In short, MotionEye is ideal for advanced setups with more cameras, while Motion is better for simpler systems or older devices.

I used ChatGPT for commands to use on my laptop to get the IP address since the commands I know are not working on my terminal.

Our GitHub repository (Alaina and Meerab): https://github.com/UAlaina/EyeOnPi.git