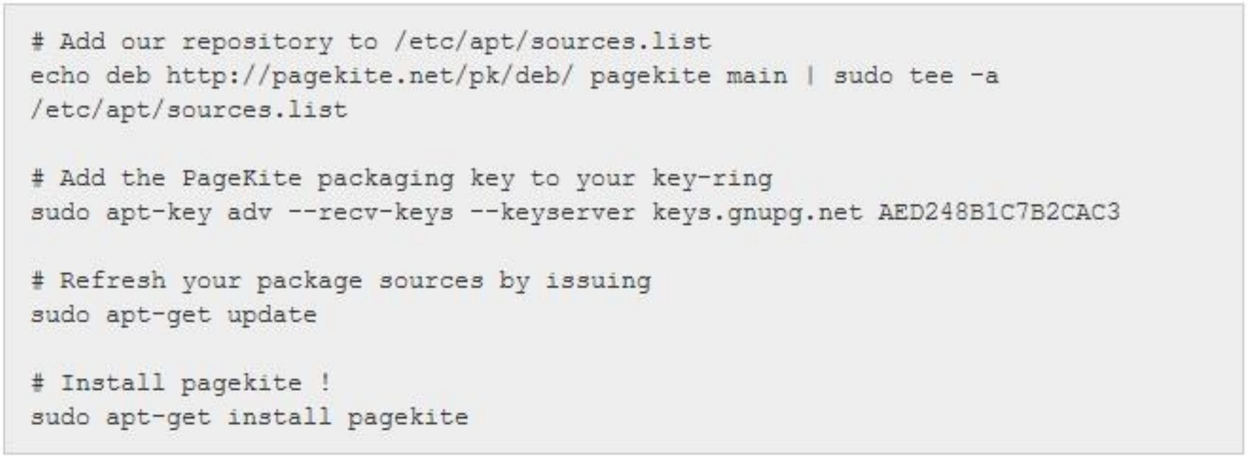
Raspberry Pi USB Security Camera System:

2. Installing and configuring “PageKite” on a Raspberry Pi

Once you have “motion” up and streaming on the RPi’s localhost, the next step will be to make that stream accessible from any javascript enabled browser, including mobile devices. After quite a bit of searching and trying out many different pieces of software, I finally came across a relatively small open source project named “PageKite” that allows either a port number or a specific folder to be made available globally and automatically secured via SSL. The guide on the PageKite website was very clear in its installation instructions: https://pagekite.net/wiki/Howto/GNULinux/DebianPackage/

The following is taken directly from the PageKite Debian installation guide, and will ensure that PageKite is installed along with the repository being added to the sources.list file:



```
# Add our repository to /etc/apt/sources.list
echo deb http://pagekite.net/pk/deb/ pagekite main | sudo tee -a
/etc/apt/sources.list

# Add the PageKite packaging key to your key-ring
sudo apt-key adv --recv-keys --keyserver keys.gnupg.net AED248B1C7B2CAC3

# Refresh your package sources by issuing
sudo apt-get update

# Install pagekite !
sudo apt-get install pagekite
```

After the software finishes installing, you’ll want to test it out to make sure everything went smoothly. To do this, I referred to the “Quick-Start Guide” on the PageKite website, though not all of these instructions apply to this particular setup: https://pagekite.net/support/quickstart/

The particular function of PageKite we are using is the “front-end connect”. This means it is taking something from the localhost and hosting it globally as a PageKite webserver. Another option is a “back-end connect” which takes a currently up-and-running webserver such as “Apache” or “lighttpd” and gives it a globally accessible PageKite domain name.

The first thing you want to do is sign up for an account. Now, PageKite is a completely free and open source software, but as it is a small project in need of funding they sort of push you to donate. However, you do NOT need to pay to use this software indefinitely for this project. The only “catch” is you’ll need to write a couple of sentences explaining what you use their software for in addition to any useful feedback in order to renew the free month-long subscription.

To sign up for an account, simply type the following in the command line:

```
$ pagekite –signup
```

Note that in the installation guide, they incorrectly state that you need to add ".py" to the end of the "pagekite" command. However, this does not apply to the Debian release. During the signup process, you will be asked to enter an email and password, along with a username. This username will be used as the main domain name for your kites, i.e. a username of "jsmith" will mean that the default kitename will be "jsmith.pagekite.me". Sub-domains can be added later. Now that you have an account set up with a main domain name available, you can try making the "motion" stream on the localhost globally accessible. To do this, enter the following command:

$ pagekite 8081 yourkitename.pagekite.me

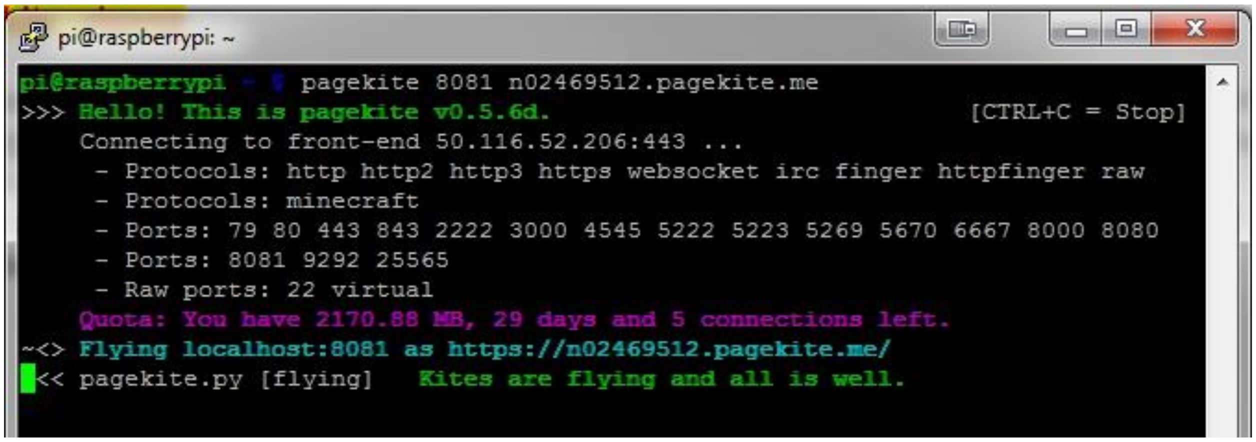
The above command is broken down into three sections:

1. The "8081" corresponds to the source. This can be a localhost port, like we have above, or another source such as a folder on the RPi.
2. The kitename, which by default must include the main domain name (your username). To created sub-domains, simply add a '-' or a '.' before the main domain name, like so:
   subname-yourkitename.pagekite.me
3. The last section, which we have not used here, is the additional configuration options, or "flags" for the kite you are about to fly. All of the available flags are listed in the Technical Manual, along with their functions. Flags such as +password and +indexes will become important in this particular project. A link to these commands:
   https://pagekite.net/wiki/Floss/TechnicalManual/

For now a complete command may look like the following:

$ pagekite 8081 subname-yourkitename.pagekite.me +password/USER=PASS

In the command line, the following will be visible:



```
pi@raspberrypi ~ $ pagekite 8081 n02469512.pagekite.me
>>> Hello! This is pagekite v0.5.6d.                           [CTRL+C = Stop]
    Connecting to front-end 50.116.52.206:443 ...
     - Protocols: http http2 http3 https websocket irc finger httpfinger raw
     - Protocols: minecraft
     - Ports: 79 80 443 843 2222 3000 4545 5222 5223 5269 5670 6667 8000 8080
     - Ports: 8081 9292 25565
     - Raw ports: 22 virtual
    Quota: You have 2170.88 MB, 29 days and 5 connections left.
~<> Flying localhost:8081 as https://n02469512.pagekite.me/
 << pagekite.py [flying]   Kites are flying and all is well.
```

In the command line, the current quote is displayed (more than enough for what we're doing, but keep an eye on how long the current subscription will last). Additionally, the localhost source and globally accessible link are displayed. If everything is working as it should, the message "Kites are flying and all is well" will be displayed. In this area, you can also see the IP address of those who attempt to access your PageKite page and whether they were able to successfully access the page or if they were denied access.

If you are denied access for any reason, it could be due to any one of the following:

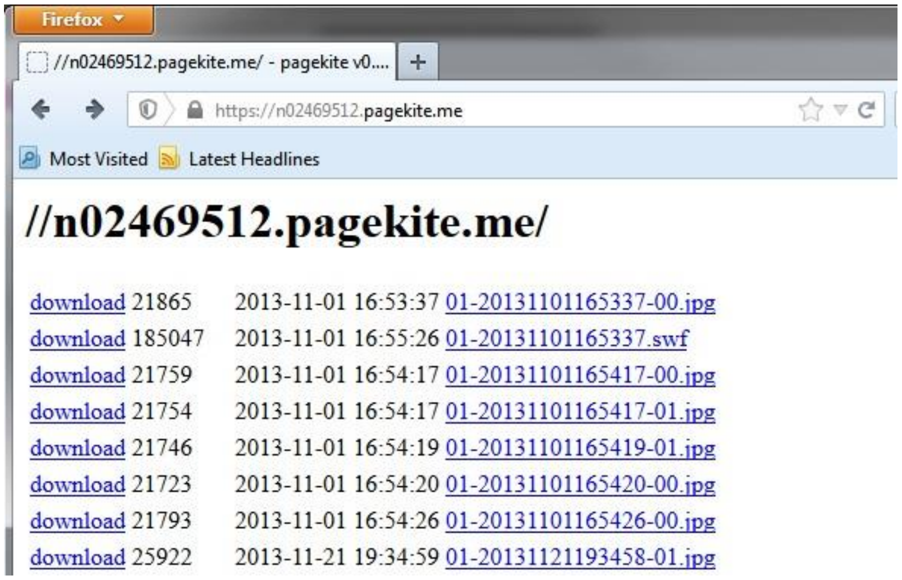
1. You are having issues connecting to the internet, check your connection.
2. If you're on a network with limited access privileges such as a college network, make sure you have internet access on the RPi.
3. The current subscription has expired. This means that you have to fill out a short form for feedback to keep using the software for free. Once this is completed, you will be able to operate PageKite normally again. This can be found here, on the right side where it says "Feeling Broke?" after moving the slider all the way to the left:
   https://pagekite.net/signup/?more=bw

If all goes well, you can enter the link highlighted above in light blue into your browser's address bar and access the stream from "motion" on localhost port 8081. Once you have the stream accessible globally, you can add passwords or IP restrictions with the +password and +ip configuration flags as shown before.

Having a stream secured via SSL is nice, but as a security system it is important to have some sort of archives of the footage. This can be easily accomplished without the need for a separate database such as "MySQL" or "sqlite". As mentioned before, PageKite can take any source on the localhost and make it globally accessible, including folders. It just so happens that since "motion" functions by taking a snapshot every time it detects motion in the feed, we can use these saved images as a database of archived footage. The location of these ".jpg" files can be changed in the "motion.conf" file but by default they will be stored in "/tmp/motion". Using this fact, we can enter the following command to create a globally accessible database:

$ pagekite /tmp/motion subname-yourkitename.pagekite.me +indexes

Again, a sub-domain name can be added in addition to a password. I prefer to use "stream-mykitename.pagekite.me" as the stream and "archives-mykitename.pagekite.me" as the database. The addition of "+indexes" will cause the files in the source folder to be automatically indexed on the PageKite webpage. After entering the above command, you will again be given a link to the webpage. You should see something similar to this for the database:


Firefox
//n02469512.pagekite.me/ - pagekite v0....
https://n02469512.pagekite.me
Most Visited
Latest Headlines

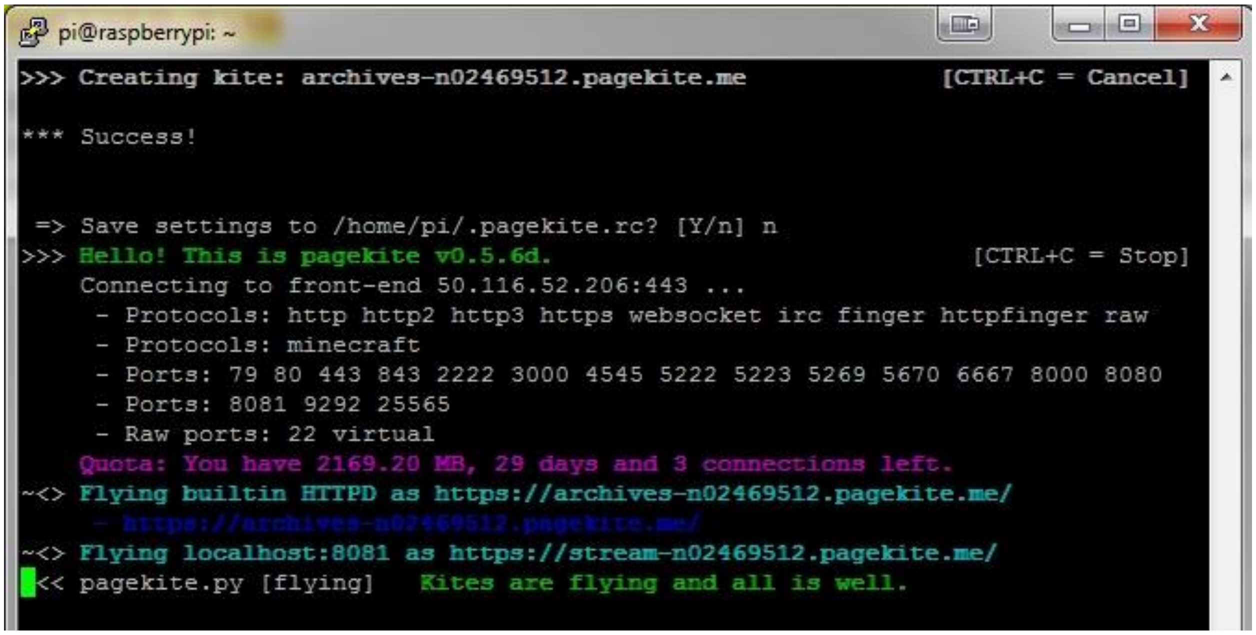
//n02469512.pagekite.me/

```
download 21865    2013-11-01 16:53:37 01-20131101165337-00.jpg
download 185047   2013-11-01 16:55:26 01-20131101165337.swf
download 21759    2013-11-01 16:54:17 01-20131101165417-00.jpg
download 21754    2013-11-01 16:54:17 01-20131101165417-01.jpg
download 21746    2013-11-01 16:54:19 01-20131101165419-01.jpg
download 21723    2013-11-01 16:54:20 01-20131101165420-00.jpg
download 21793    2013-11-01 16:54:26 01-20131101165426-00.jpg
download 25922    2013-11-21 19:34:59 01-20131121193458-01.jpg
```

Now that you have the stream and database running individually, you can run them simultaneously by simply adding an "AND" in between the two commands like so:

$ pagekite /tmp/motion archives-yourkitename.pagekite.me +indexes +password/USER=PASS AND 8081 stream-yourkitename.pagekite.me +password/USER=PASS

You'll be prompted to accept the name for each kite, just type Y for the first two. The third prompt will ask if you want to save the settings to "/home/pi/.pagekite.rc". This will mean that you cannot reuse the same sub-domain name without getting an error. Type "N" for this prompt, and you'll be able to enter the same sub-domain in later uses. The following will be displayed:


pi@raspberrypi: ~

```
>>> Creating kite: archives-n02469512.pagekite.me             [CTRL+C = Cancel]

*** Success!


 => Save settings to /home/pi/.pagekite.rc? [Y/n] n
>>> Hello! This is pagekite v0.5.6d.                             [CTRL+C = Stop]
    Connecting to front-end 50.116.52.206:443 ...
     - Protocols: http http2 http3 https websocket irc finger httpfinger raw
     - Protocols: minecraft
     - Ports: 79 80 443 843 2222 3000 4545 5222 5223 5269 5670 6667 8000 8080
     - Ports: 8081 9292 25565
     - Raw ports: 22 virtual
    Quota: You have 2169.20 MB, 29 days and 3 connections left.
~<> Flying builtin HTTPD as https://archives-n02469512.pagekite.me/
     - https://archives-n02469512.pagekite.me/
~<> Flying localhost:8081 as https://stream-n02469512.pagekite.me/
 << pagekite.py [flying]   Kites are flying and all is well.
```

Note: This can be running as a daemon process by adding "--daemonize" directly after the "pagekite" command, like so:

$ pagekite --daemonize /tmp/motion…

As shown above, both kites are flying simultaneously, each with their own address. Each page is password protected and the database files are indexed. While this is nice to have both the stream and database accessible with a single command, you still have to manually switch between each one by entering the corresponding addresses. In the next guide, we'll examine how to eliminate this need by creating a central webserver page that links to the stream and archives. Additionally, this will allow multiple camera/pi combinations to be implemented by simply linking the PageKite pages for the other streams/databases on this central page.