USB Security Camera System

Nicholas Crimmins

- The overall goal of the project was to stream from a USB webcam connected to a Raspberry Pi and to have the stream accessible anywhere via any javascript enabled browser.
- Have stream secured via SSL (Secure Socket Layer)
- Eventually have multiple pi/camera combinations streaming to create a security system.

Research and Background Info

- Had to find a webcam that would be compatible with the RPi: http://elinux.org/RPi_USB_Webcams
  - Chosen based on power, resolution, stream quality
- “Motion” vs. “mpeg-streamer” as a package
  - Chose “Motion”; easy to install and configure, Raspbian
  - “Mpeg-streamer” higher quality, but more complicated

Research and Background Info (Cont.)

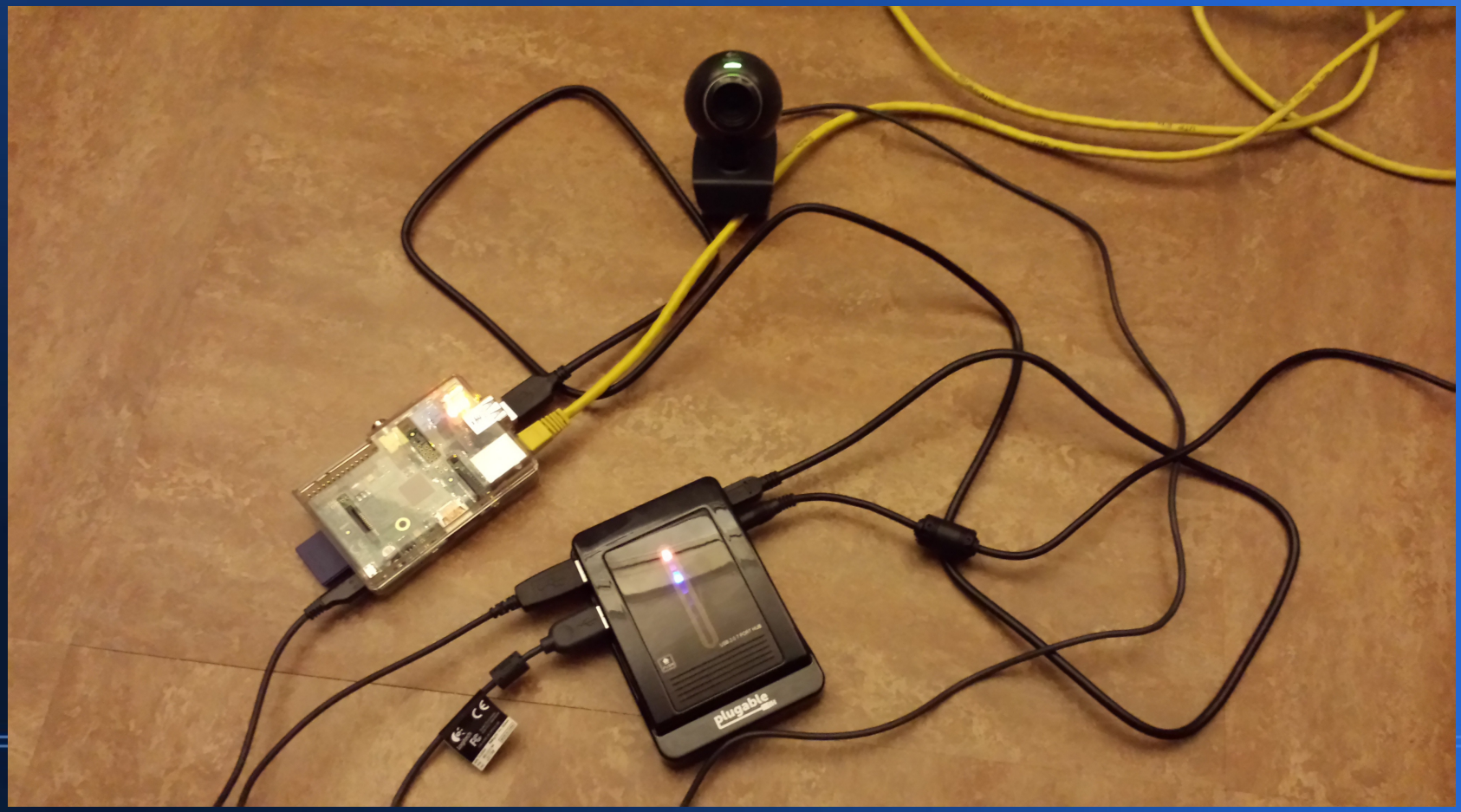
- With “Motion”, can stream only on localhost
- Had to find a way to make localhost globally accessible
- “Pagekite” does just that
  - Host stream on Pagekite server
  - Free, 100% open source, python based
  - Use own DNS sever name for address
  - Create database from saved .jpg files
  - More security via login/password
  - Easy to configure SSL

Research and Background Info (Cont.)

- Multiple cameras on a single Pi puts too much strain on CPU. Depending on cameras it could either not work at all or cause poor quality streaming.
- Had to make the system expandable for multiple pi/camera combinations.
  - Make separate web page to link to other streams.
  - Central access point for the entire system.
  - Use lighttpd for web server, PHP for simple scripting

Procedure

1) Found compatible webcam, set up with Pi



Procedure (cont.)

2) Installed packages from Raspbian repository using apt-get:

- motion
- lighttpd
- php5-common php5-cgi php5 *(lightweight PHP)*

Procedure (cont.)

3) Installed Pagekite from Pagekite repository

- Add repo to sources.list
- Install Pagekite via apt-get
- pagekite –signup *(free account)*

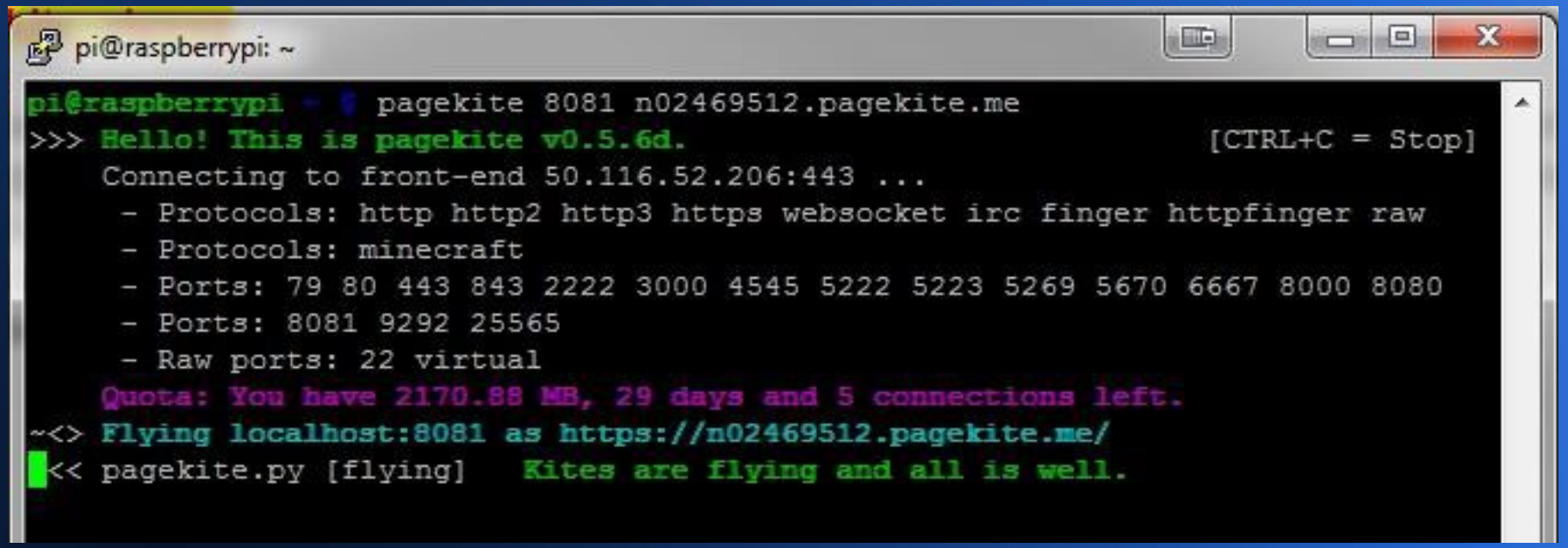
Procedure (cont.)

4) Configured motion to stream correctly by accessing /etc/motion.conf

- Enable streaming to places other than localhost
- Daemonize
- Adjust resolution for quality
- Configure port for webcam to use *(8081 by default)*
- Test streaming to localhost
    - 192.168.X.X:8081

Procedure (cont.)

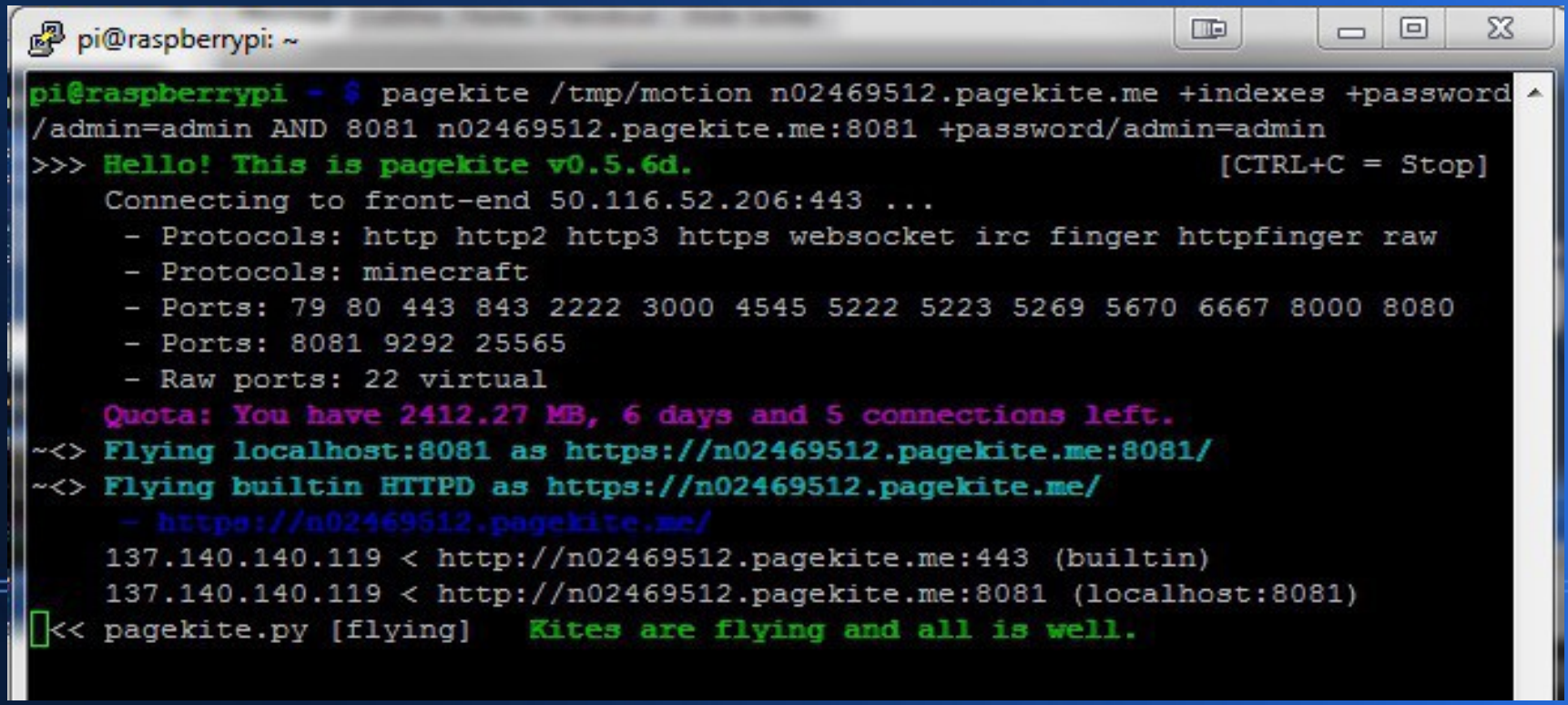
5) Streamed on localhost, then streamed globally using Pagekite, and tested:



Procedure (cont.)

6) Configured Pagekite further, and tested:

- +password/<USER>=<PASS>
- Created simple database using /tmp/motion as source



Procedure (cont.)

7) Finally, created a central access point for the whole system using lighttpd and PHP:

- Simple PHP script to embed links in text on the web page
- Can add more links for more streams from additional pi/camera combinations

“index.php”

```
<?php
  print <<< EOT
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>USB Security Camera System</title>
</head>
<body>
<h1>USB Security Camera System</h1>
<p>Each stream is linked below, along with the corresponding archives for
that camera:</p>
<a href="https://stream1-n02469512.pagekite.me">Stream 1</a>
<a href="https://archives1-n02469512.pagekite.me">Stream 1 - Archives</a>
<br>
<p>Stream2 Stream2 - Archives</p>
</body>
</html>
EOT;
?>
```

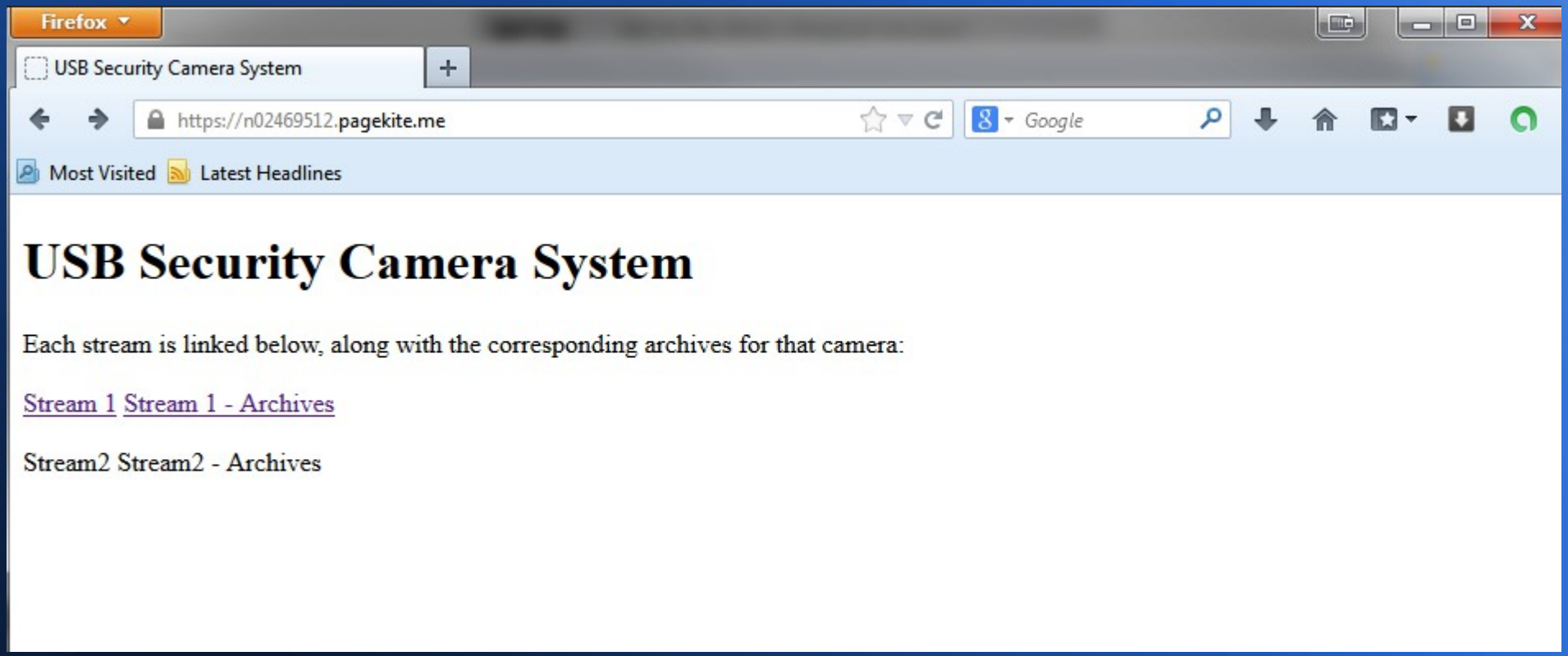
Running Everything Together

- Three different PageKite pages running:
- Stream (port 8081)
- Database (/tmp/motion)
- Central access point (lighttpd server on port 80)

```
pi@raspberrypi ~ $ pagekite 8081 stream1-n02469512.pagekite.me +password/admin=a
dmin AND /tmp/motion archives1-n02469512.pagekite.me +indexes +password/admin=ad
min AND 80 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 2169.20 MB, 29 days and 4 connections left.
~<> Flying builtin HTTPD as https://archives1-n02469512.pagekite.me/
     - https://archives1-n02469512.pagekite.me/
~<> Flying localhost:8081 as https://stream1-n02469512.pagekite.me/
~<> Flying localhost:80 as https://n02469512.pagekite.me/
<< pagekite.py [flying]   Kites are flying and all is well.
```

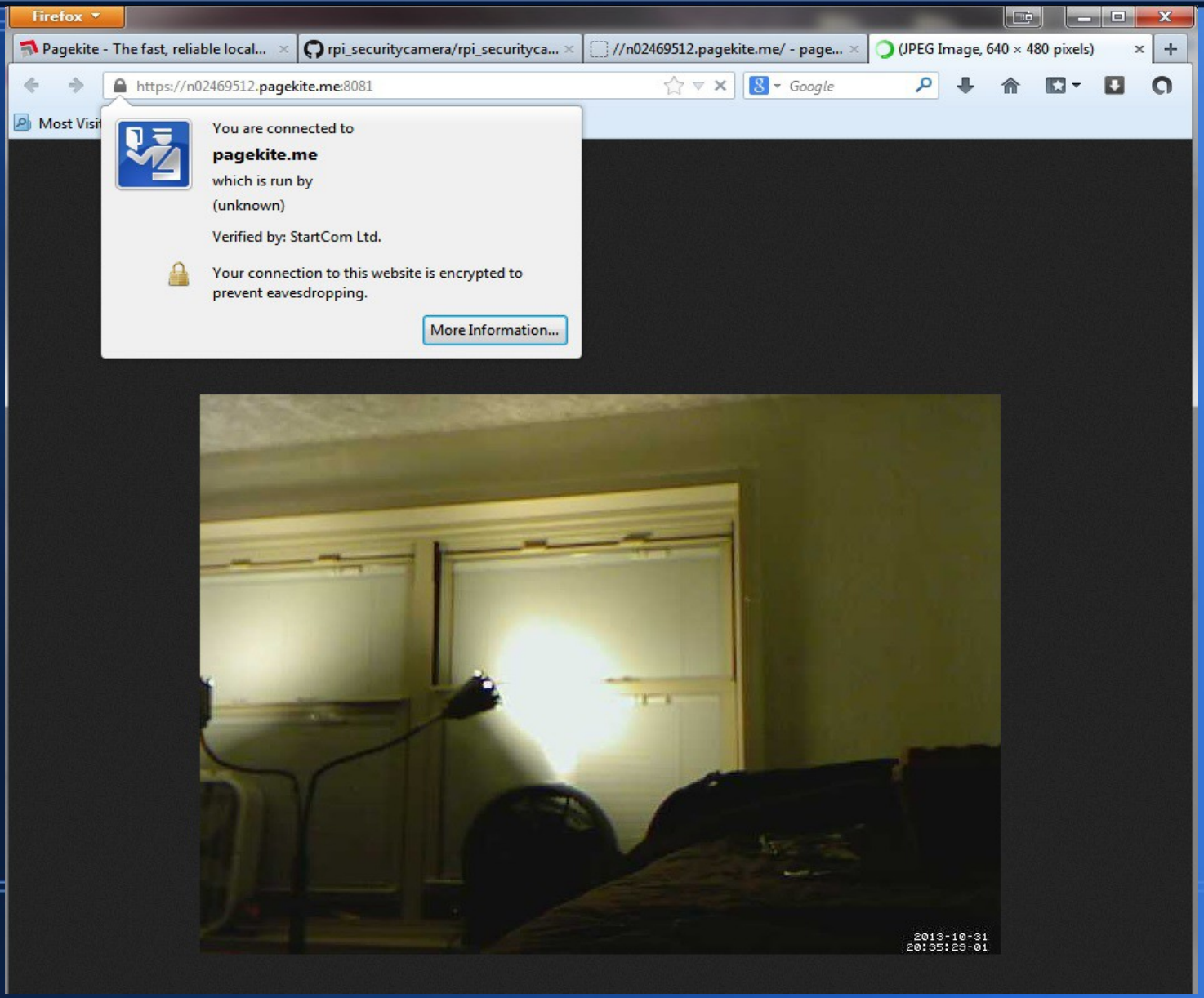
Central Access Point



- Links to both the stream and the database
- Add additional streams/archives from other camera/pi combinations

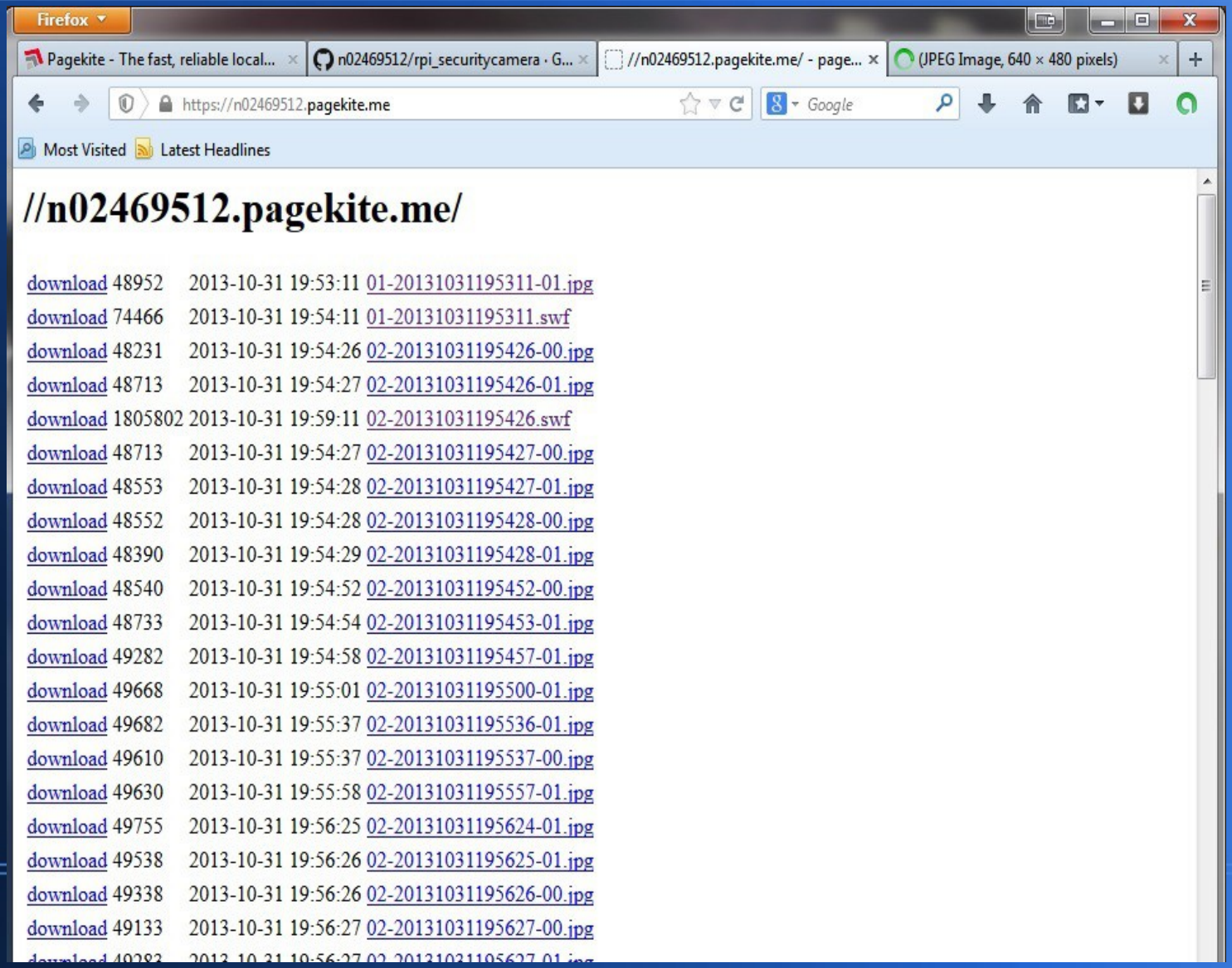
The Stream

- Access to the stream granted after entering username and password
- Secured via SSL



The Database

- Access to the database also requires user/pass
- View individual .jpg files
- File size, date/time recorded, and file name shown.
- Uploaded from /tmp/motion on RPi



Conclusion

- Learned basics of python & PHP as well as SSL, front-end connects (making localhost public), static IP addresses, DNS server names, SSH, etc.
- Problems faced:
  - Making localhost public (before using Pagekite)
  - Streaming with multiple cameras on a single Pi
  - Having multiple streams on a single web page
- Future:
  - Use as personal security system
  - Control remotely, record only when certain events take place (light turns on, specific times, etc.)