# University of Colorado at Colorado Springs CS4930/5930 - Spring 2019 Privacy and Censorship Assignment 2 - Setting Up Your Own AdBlock

**Instructor: Yanyan Zhuang**
**Total Points: 100**
**Out: 4/1/2019**
**Due: 11:59 pm, Friday, 4/19/2019**

## Description

In this assignment, you will learn how to see http requests and responses when you visit a website, and how to block content by yourself. Please note that **the following instructions will work for a Mac or Linux machine**. If you are using a Windows machine, you may (1) lookup how to setup docker, and how to configure the `/etc/hosts` file[1], or (2) setup a virtual machine using VirtualBox, and install Ubuntu 16.04 or 18.04.

Assignment submission: please answer the following questions (Q1 – Q4), and attach screenshots when necessary.

### Step 1: Setting up a local webserver using Docker (30 pts)

First, install Docker Community Edition (CE) by following the instructions at `https://docs.docker.com/install/`. Docker is like a light-weight virtual machine. You can reuse any of its existing containers (like existing virtual machine images).

If you are using Mac, please follow this link: `https://docs.docker.com/docker-for-mac/install/`. They now require that you create an account (Sorry!).

If you are using Linux (Ubuntu 16.04 as an example), do the following (please note that **Docker requires a Linux kernel version of 3.10 or newer**.):

```
sudo apt install docker.io
```

To verify the installation, do

```
sudo docker run hello-world
```

The first time it will download hello-world and display:

```
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
...
Hello from Docker!
This message shows that your installation appears to be working correctly.
```

[1] For example, `https://gist.github.com/zenorocha/18b10a14b2deb214dc4ce43a2d2e2992`

In case the installation fails, please refer to here and use the "Official Docker Way": `https://askubuntu.com/questions/938700/how-do-i-install-docker-on-ubuntu-16-04-lts`.

Next, download the `nginx` container. This step is the same for both Mac and Linux (you may need sudo in Ubuntu):

```
docker pull nginx
```

The `nginx` container is like a VM that has a webserver setup for you. You'll see something like

```
Using default tag: latest
latest: Pulling from library/nginx
2a72cbf407d6: Pull complete
fefa2faca81f: Pull complete
080aeede8114: Pull complete
Digest: sha256: ......
Status: Downloaded newer image for nginx:latest
```

Then run `nginx` like running a normal process (you may need sudo in Ubuntu):

```
docker run -p 8080:80 nginx
```

Now you can browse `127.0.0.1:8080` in your browser, where you will see a welcome message from `nginx`. You can stop `nginx` by Ctrl+C.

Now replace the welcome message with something else, by doing the following **in your current directory**:

```
mkdir html && echo "LOL" >> html/index.html
```

Then stop the previous docker process by Ctrl+C, and run this **in your current directory** (don't move away from where you created the `html` directory):

```
docker run -v $PWD/html/:/usr/share/nginx/html:ro -p 8080:80 nginx
```

The second command above uses a volume (the `-v` option) to share information between your laptop and the Docker container. Then, refresh `127.0.0.1:8080` in your browser. You should now see `LOL`.

**Q1:** What do you think has happened here? Please attach a screenshot when you browse `127.0.0.1:8080` in your browser.

### Step 2: Redirecting HTTP requests to your local webserver (30 pts)

Now edit `/etc/hosts` (with sudo permission):

```
...
127.0.0.1       localhost
255.255.255.255 broadcasthost
::1             localhost
```

Add the following line at the end of this file:

```
127.0.0.1 www.4930-5930.com
```

Now type `www.4930-5930.com:8080` in your browser, with your `nginx` container running. You should see a page with `LOL`. To get rid of `:8080`, stop and restart `nginx` with the following (you may need sudo in Ubuntu):

```
docker run -v $PWD/html/:/usr/share/nginx/html:ro -p 80:80 nginx
```

Note that we are using port 80 instead of 8080 above. Now type `www.4930-5930.com` in your browser. You again should see the page with `LOL`.

**Q2:** What do you think has happened here? `www.4930-5930.com` is a domain that I made up, i.e., it doesn't exist. Why can I access it from my browser now? Please attach a screenshot when you go to `www.4930-5930.com` in your browser.

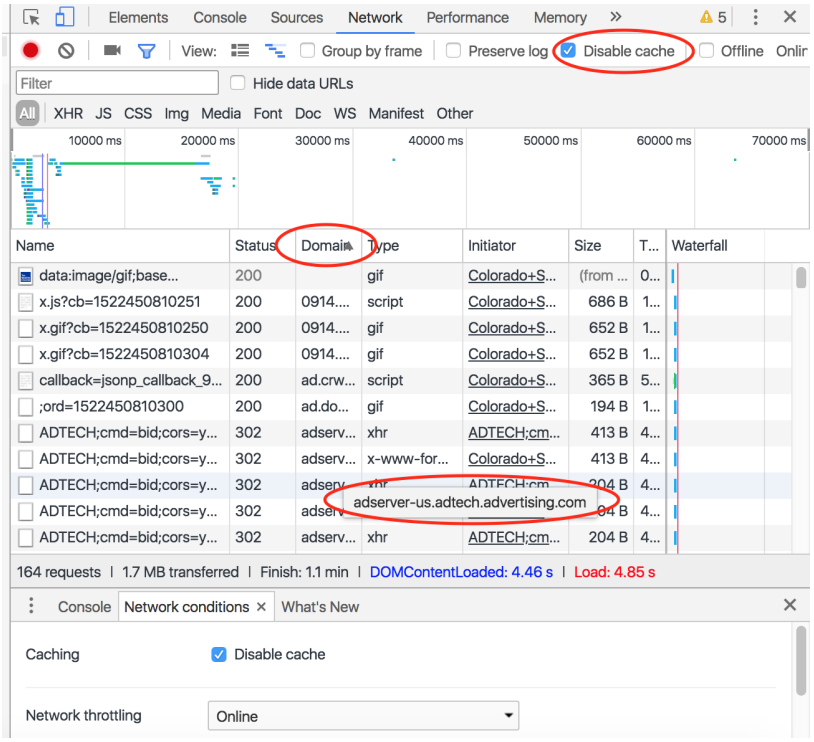


Figure 1: Developer Tools – Network.

## Step 3: Viewing HTTP requests and identify ads (20 pts)

Open a new tab in your browser (Chrome as an example), and type in the address:

`https://weather.com/weather/tenday/l/Colorado+Springs+CO+USCO0078:1:US`

**If you have AdBlock, Disconnect or any blocking tool installed, please disable it for this experiment**. In your browser (Chrome as an example), stay on this page, open Developer Tools by clicking "View" – "Developer" – "Developer Tools", and click the "Network" tab. You should see something like Figure 1. Please check the "Disable cache" box (the upper right corner). If you are using Firefox, you can open "Tools" – "Web Developer" – "Toggle Tools", and click the "Network" tab.

Now we want to see the domains where each HTTP(S) request is going to. So if you do not see the "Domain" column, right click the "Name" column and check "Domain". It will appear as in Figure 1. Hover over some cells in "Domain", you can see many of them are obviously Ad Networks, like `adserver-us.adtech.advertising.com` in Figure 1.

**Q3:** Please identify 2 or 3 other Ad Networks by following the steps above. Please attach a screenshot of your Network tab in Developer Tools. You may try different websites (e.g., `https://www.nytimes.com/`) and identify more Ad domains.

## Step 4: Blocking Ad requests (20 pts)

Choose one of the identified Ad network domain in Step 3, and edit `/etc/hosts` with sudo permission (I'm using `adserver-us.adtech.advertising.com` as an example):

```
...
127.0.0.1       localhost
255.255.255.255 broadcasthost
```

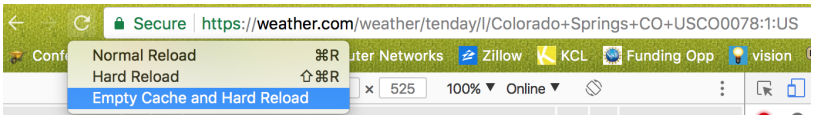


Figure 2: Cache-less Reload.

```
::1             localhost
127.0.0.1   adserver-us.adtech.advertising.com
```

Next, please do a cache-less reload in your browser, by right-clicking the refresh button to the left of the address line, then click "Empty Cache and Hard Reload", see Figure 2. While you reload, please make sure that the `nginx` container is running on port `80` (for HTTP) or `443` (for HTTPS), e.g,:

```
docker run -v $PWD/html/:/usr/share/nginx/html:ro -p 443:80 nginx
```

You may look into the Name column of the Network tab to see if a request is HTTPS. After reload, I can see that all the requests to `adserver-us.adtech.advertising.com` are blocked. See Figure 3. Can you see anything from `nginx` logs?

**Q4:** Please replicate the steps above, and attach a screenshot where the requests to an Ad network domain were blocked. You are free to try different websites (e.g., NYTimes, CNN, FoxNews) and different Ad domains.

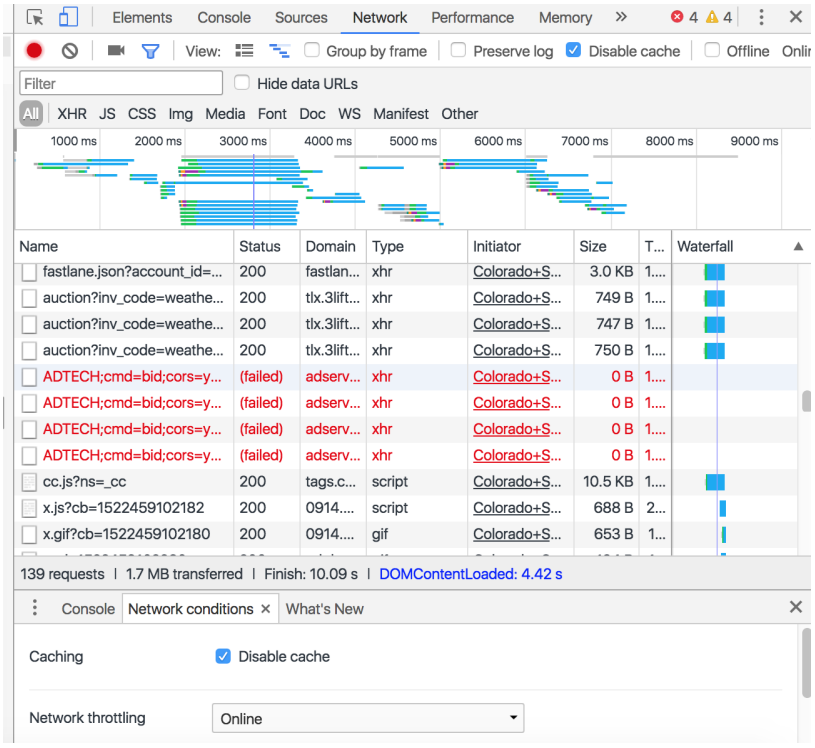


Figure 3: Requests blocked.