Pre – conditions:

1. Linux machine with Ubuntu 14.04 preferred.
2. This machine must be the bottleneck between the endpoints and the outside world.

Setup:

1. Install perl/CGI scripts with apache2:

<http://perlmaven.com/perl-cgi-script-with-apache2>

1. Go to step 4 in the following tutorial and copy the txt file at the end of the step:

<http://www.instructables.com/id/Simple-linux-commands-from-a-web-page/step4/The-main-program/>

1. Directory that stores the .cgi/.pl files: /var/cgi-bin

Note: that is done only after completing the tutorial above.

1. Create a file and paste the txt file into it.
2. Edit the beginning of the file to be #!/bin/sh
3. Step 6- add a feature (with or without an input).
4. Make the command available without sudo:

\*edit- sudoer file:

<http://askubuntu.com/questions/159007/how-do-i-run-specific-sudo-commands-without-a-password>

The user that should be granted authorization is

**www-data**

1. Find out who is running apache:

<http://serverfault.com/questions/125865/finding-out-what-user-apache-is-running-as>

1. add CSS style and images- put those files in /var/www/html (not in the cgi-bin folder)

# Added Commands:

## Traffic control basic guide:

<http://www.linuxfoundation.org/collaborate/workgroups/networking/netem>

## Bandwidth limitation:

<http://mark.koli.ch/slowdown-throttle-bandwidth-linux-network-interface>

#download:  
sudo tc qdisc add dev eth1 handle 1: root htb default 11  
sudo tc class add dev eth1 parent 1: classid 1:1 htb rate "$DLWONDERSHAPER"kbps  
sudo tc class add dev eth1 parent 1:1 classid 1:11 htb rate "$DLWONDERSHAPER"kbps

#upload:  
sudo tc qdisc add dev eth0 handle 1: root htb default 11  
sudo tc class add dev eth0 parent 1: classid 1:1 htb rate "$ULWONDERSHAPER"kbps  
sudo tc class add dev eth0 parent 1:1 classid 1:11 htb rate "$ULWONDERSHAPER"kbps

## Packetloss

sudo /sbin/tc qdisc add dev eth0 root netem loss "$LOSS"%

sudo /sbin/tc qdisc change dev eth0 root netem loss "$LOSS"%

sudo /sbin/tc qdisc add dev eth1 root netem loss "$LOSS"%

sudo /sbin/tc qdisc change dev eth1 root netem loss "$LOSS"%

## Jitter

sudo /sbin/tc qdisc add dev eth0 root netem delay "$JITTER\_START"ms "$JITTER\_END"ms reorder 25% 50%

sudo /sbin/tc qdisc change dev eth0 root netem delay "$JITTER\_START"ms "$JITTER\_END"ms reorder 25% 50%

sudo /sbin/tc qdisc add dev eth1 root netem delay "$JITTER\_START"ms "$JITTER\_END"ms reorder 25% 50%

sudo /sbin/tc qdisc change dev eth1 root netem delay "$JITTER\_START"ms "$JITTER\_END"ms reorder 25% 50%

## Latency

sudo /sbin/tc qdisc add dev eth0 root netem delay "$DELAY"ms

sudo /sbin/tc qdisc change dev eth0 root netem delay "$DELAY"ms

sudo /sbin/tc qdisc add dev eth1 root netem delay "$DELAY"ms

sudo /sbin/tc qdisc change dev eth1 root netem delay "$DELAY"ms

## iptables - use to block a specific ip

These are the clear command(s).

In order to add a command simply replace the -D with -A

sudo /sbin/iptables -D FORWARD -s "$IP\_CLEAR" -p tcp --sport 0:65535 --dport 443 -m state --state NEW,ESTABLISHED -j ACCEPT  
sudo /sbin/iptables -D FORWARD -s "$IP\_CLEAR" -p tcp --sport 0:65535 --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT  
sudo /sbin/iptables -D FORWARD -s "$IP\_CLEAR" -p tcp --dport 80 -j DROP  
sudo /sbin/iptables -D FORWARD -s "$IP\_CLEAR" -p tcp --dport 17990 -j DROP  
sudo /sbin/iptables -D FORWARD -s "$IP\_CLEAR" -p tcp --dport 17992 -j DROP  
sudo /sbin/iptables -D FORWARD -s "$IP\_CLEAR" -p tcp --dport 5061 -j DROP  
sudo /sbin/iptables -D FORWARD -s "$IP\_CLEAR" -p udp --match multiport --dports 50000:65535 -j DROP  
sudo /sbin/iptables -D FORWARD -s "$IP\_CLEAR" -p tcp --dport 8443 -j DROP  
sudo /sbin/iptables -D FORWARD -s "$IP\_CLEAR" -p tcp --dport 443 -j DROP  
sudo /sbin/iptables -D FORWARD -s "$IP\_CLEAR" -p tcp --sport 0:65535 --dport 5061 -m state --state NEW,ESTABLISHED -j ACCEPT  
sudo /sbin/iptables -D FORWARD -s "$IP\_CLEAR" -j DROP