Transparent HTTPS filtering on CentOS

This HOWTO will tell you how to set up Squid on CentOS / RedHat Linux for transparent filtering of HTTP and HTTPS traffic with help of Diladele Web Safety ICAP server.

# Assumptions and Requirements

HTTPS protocol is used more and more in today’s web. While this may be good for privacy, it leaves modern network administrator without any means to prevent questionable or adult contents from entering his/her network. Previously it was assumed that this problem does not have a decent solution. Our howto will try to prove otherwise.

I will assume you have a network with IP addresses from 192.168.1.0 subnet, network mask is 255.255.255.0, all workstations are set to use 192.168.1.1 as default gateway. On this default gateway you have two NICs - one facing LAN with IP address 192.168.1.1, the other is plugged in into ISP network and gets its public Internet address through DHCP. It is also assumed your gateway has CentOS or RedHat Linux up and running.

# Step 1. Update and Upgrade

Before going further run the following script to upgrade your system to the most recent state.

|  |
| --- |
| #!/bin/bash  set -e  # update should be done as root  if [[ $EUID -ne 0 ]]; then  echo "This script must be run as root" 1>&2  exit 1  fi  # update and upgrade  yum update && yum upgrade  # disable selinux  sed -i s/SELINUX=enforcing/SELINUX=disabled/g /etc/selinux/config  # and reboot  reboot |

# Step 2. Install Apache Web Server

Diladele Web Safety has sophisticated web administrator console to easily manage filtering settings and policies. This Web UI is built using Python Django web framework and requires Apache web server to function correctly. Run the following script to install them.

|  |
| --- |
| #!/bin/bash  set -e  # all web packages are installed as root  if [[ $EUID -ne 0 ]]; then  echo "This script must be run as root" 1>&2  exit 1  fi  # install python libs  yum install python-setuptools python-ldap  # install python django for web ui  easy\_install django==1.5  # install apache web server to run web ui  yum install httpd php mod\_wsgi  # make apache autostart on reboot  chkconfig httpd on  # this fixes some apache errors when working with python-django wsgi  echo "WSGISocketPrefix /var/run/wsgi" >> /etc/httpd/conf.d/wsgi.conf  # and restart apache  service httpd restart |

# Step 3. Install Diladele Web Safety

Download and install latest version of Diladele Web Safety.

|  |
| --- |
| #!/bin/bash  # all packages are installed as root  if [[ $EUID -ne 0 ]]; then  echo "This script must be run as root" 1>&2  exit 1  fi  # detect current architecture (default assumes x86\_64)  ARCH\_1=`uname -m`  ARCH\_2="amd64"  if [[ $ARCH\_1 == 'i686' ]]; then  ARCH\_1="i386"  ARCH\_2="i386"  fi  # bail out on any error  set -e  # get latest qlproxy  curl http://updates.diladele.com/qlproxy/binaries/3.2.0.4CAF/$ARCH\_2/release/centos6/qlproxy-3.2.0-4CAF.$ARCH\_1.rpm > qlproxy-3.2.0-4CAF.$ARCH\_1.rpm  # install it  yum -y --nogpgcheck localinstall qlproxy-3.2.0-4CAF.$ARCH\_1.rpm    # qlproxy installed everything needed for apache, so just restart  service httpd restart |

# Step 4. Install Required Build Tools

To be able to perform HTTP/HTTPS transparent filtering we need to get the latest version of Squid (the one that comes with CentOS / RedHat by default is too outdated) and rebuild it from source. The following script installs all build tools required.

|  |
| --- |
| #!/bin/bash  # install all build tools  if [[ $EUID -ne 0 ]]; then  echo "This script must be run as root" 1>&2  exit 1  fi  # install development packages required  yum install -y gcc-c++ pam-devel db4-devel expat-devel libxml2-devel libcap-devel libtool redhat-rpm-config rpm-build openldap-devel openssl-devel krb5-devel  # squid needs perl and needs additional perl modules not present by default in CentOS 6  curl http://dl.fedoraproject.org/pub/epel/6/x86\_64/epel-release-6-8.noarch.rpm > epel-release-6-8.noarch.rpm  rpm -Uvh epel-release-6\*.rpm  yum install -y perl-Crypt-OpenSSL-X509 |

# Step 5. Build Squid from Source

Rebuild the Squid RPM by running the following script.

|  |
| --- |
| #!/bin/bash  # stop on any error  set -e  # rpm build MUST be run as normal user  if [[ $EUID -eq 0 ]]; then  echo "This script must NOT be run as root" 1>&2  exit 1  fi  # get squid sources  pushd rpmbuild/SOURCES  curl http://www.squid-cache.org/Versions/v3/3.4/squid-3.4.4.tar.xz > squid-3.4.4.tar.xz  curl http://www.squid-cache.org/Versions/v3/3.4/squid-3.4.4.tar.xz.asc > squid-3.4.4.tar.xz.asc  popd  # build the binaries RPMs out of sources  pushd rpmbuild/SPECS  rpmbuild -v -bb squid.spec  popd |

# Step 6. Install Squid

After build finishes, install Squid. It is advisable to uncomment the lines which generate your own root certification authority. Default installation of Diladele Web Safety does have its own ca but trusting it may pose serious security risk of your devices are used by users outside of your network.

|  |
| --- |
| #!/bin/bash  # stop on every error  set -e  # install RPMs as root  if [[ $EUID -ne 0 ]]; then  echo "This script must be run as root" 1>&2  exit 1  fi  # detect current architecture (default assumes x86\_64)  ARCH\_1=`uname -m`  ARCH\_2="amd64"  ARCH\_3="lib64"  if [[ $ARCH\_1 == 'i686' ]]; then  ARCH\_2="i386"  ARCH\_3="lib"  fi  pushd rpmbuild/RPMS/$ARCH\_1  yum localinstall -y squid-3.4.4-0.el6.$ARCH\_1.rpm  popd  # set up the ssl\_crtd daemon  if [ -f /bin/ssl\_crtd ]; then  rm -f /bin/ssl\_crtd  fi  ln -s /usr/$ARCH\_3/squid/ssl\_crtd /bin/ssl\_crtd  /bin/ssl\_crtd -c -s /var/spool/squid\_ssldb  chown -R squid:squid /var/spool/squid\_ssldb  # uncomment to regenerate certificates for SSL bumping if you do not like defaults  # openssl req -new -newkey rsa:1024 -days 1365 -nodes -x509 -keyout myca.pem -out myca.pem  # openssl x509 -in myca.pem -outform DER -out myca.der  # then copy certificates  # cp myca.pem /etc/opt/quintolabs/qlproxy/  # cp myca.der /etc/opt/quintolabs/qlproxy/  # make squid autostart after reboot  chkconfig squid on |

# Step 7. Integrate Squid with Diladele Web Safety

Integrate Squid and Diladele Web Safety by running the following script.

|  |
| --- |
| #!/bin/bash  # stop on any error  set -e  # integration should be done as root  if [[ $EUID -ne 0 ]]; then  echo "This script must be run as root" 1>&2  exit 1  fi  # allow web ui read-only access to squid configuration file  chmod o+r /etc/squid/squid.conf  # perform integration by replacing squid.conf file  mv /etc/squid/squid.conf /etc/squid/squid.conf.original && mv squid.conf /etc/squid/squid.conf  # parse the resulting config just to be sure  /usr/sbin/squid -k parse  # restart squid to load all config  /sbin/service squid restart |

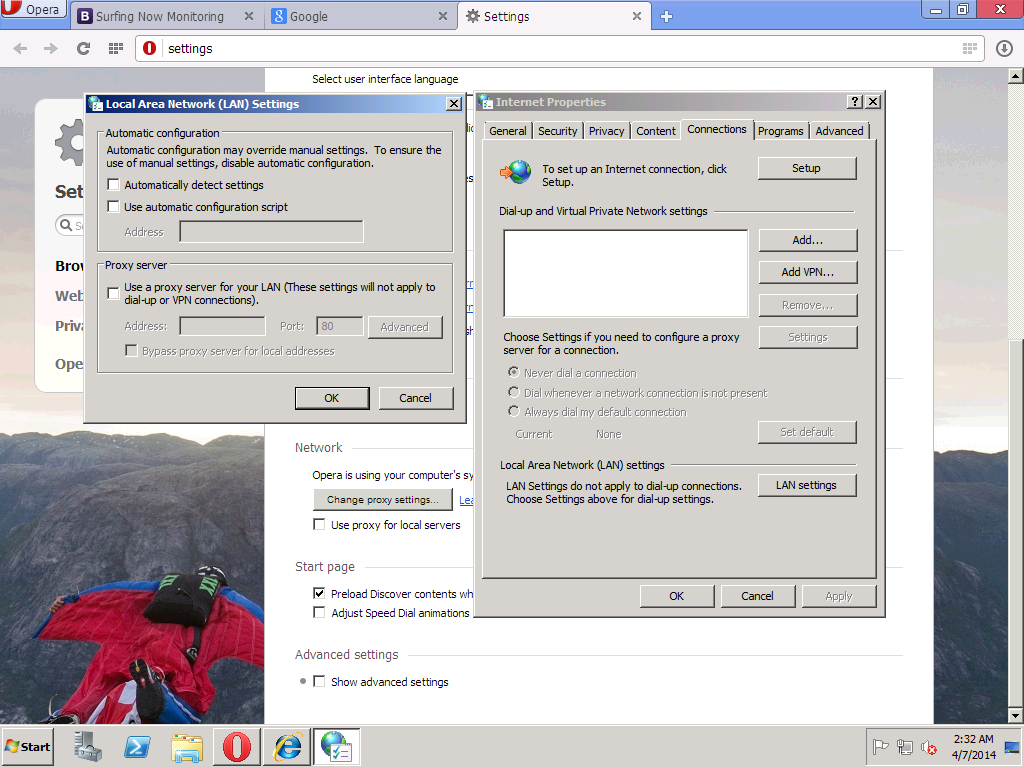
# Step 8. Transparently Redirect HTTPS traffic to Squid

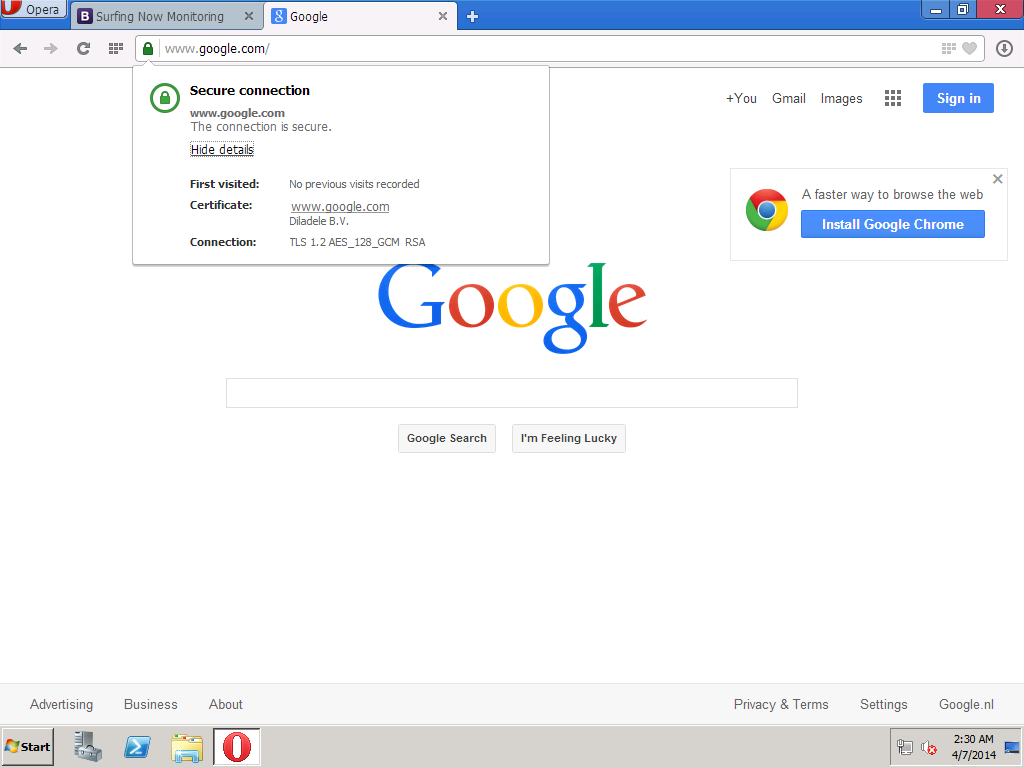
Transparently filtering HTTP and HTTPS traffic will be implemented by redirecting traffic to ports 80 and 443 to Squid using iptables. This implies the box with Squid acts as default gateway for your LAN. Please note this is only one way to implementing transparent filtering, other possible solutions are described in [Squid’s Wiki](http://wiki.squid-cache.org/ConfigExamples/Intercept).

|  |
| --- |
| #!/bin/bash  # firewall setup should be done as root  if [[ $EUID -ne 0 ]]; then  echo "This script must be run as root" 1>&2  exit 1  fi  # check kernel forwarding is enabled  enabled=`cat /proc/sys/net/ipv4/ip\_forward`  if [[ $enabled -ne 1 ]]; then  echo "Kernel forwarding seems to be disabled, enable it in /etc/sysctl.conf, reboot and rerun this script" 1>&2  exit 1  fi  # set the default policy to accept first (not to lock ourselves out from remote machine)  iptables -P INPUT ACCEPT  # flush all current rules from iptables  iptables -F  # allow pings from eth0 and eth1 for debugging purposes  iptables -A INPUT -p icmp -j ACCEPT  # allow access for localhost  iptables -A INPUT -i lo -j ACCEPT  # accept packets belonging to established and related connections  iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT  # allow ssh connections to tcp port 22 from eth0 and eth1  iptables -A INPUT -p tcp --dport 22 -j ACCEPT  # allow connection from LAN to ports 3126, 3127 and 3128 squid is running on  iptables -A INPUT -i eth0 -p tcp --dport 3126 -j ACCEPT  iptables -A INPUT -i eth0 -p tcp --dport 3127 -j ACCEPT  iptables -A INPUT -i eth0 -p tcp --dport 3128 -j ACCEPT  # redirect all HTTP(tcp:80) traffic coming in through eth0 to 3126  iptables -t nat -A PREROUTING -i eth0 -p tcp -m tcp --dport 80 -j REDIRECT --to-ports 3126  # redirect all HTTPS(tcp:443) traffic coming in through eth0 to 3127  iptables -t nat -A PREROUTING -i eth0 -p tcp -m tcp --dport 443 -j REDIRECT --to-ports 3127  # configure forwarding rules  iptables -A FORWARD -i eth0 -o eth1 -p tcp --dport 22 -j ACCEPT  iptables -A FORWARD -i eth1 -o eth0 -p tcp --sport 22 -j ACCEPT  iptables -A FORWARD -p icmp -j ACCEPT  iptables -A FORWARD -i eth0 -o eth1 -p tcp --dport 80 -j ACCEPT  iptables -A FORWARD -i eth1 -o eth0 -p tcp --sport 80 -j ACCEPT  iptables -A FORWARD -i eth0 -o eth1 -p tcp --dport 53 -j ACCEPT  iptables -A FORWARD -i eth0 -o eth1 -p udp --dport 53 -j ACCEPT  iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT  iptables -A FORWARD -j REJECT --reject-with icmp-host-prohibited  # enable NAT for clients within LAN  iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE  # set default policies for INPUT, FORWARD (drop) and OUTPUT (accept) chains  iptables -P INPUT DROP  iptables -P FORWARD DROP  iptables -P OUTPUT ACCEPT  # list created rules  iptables -L -v  # save the rules so that after reboot they are automatically restored  /sbin/service iptables save  # enable the firewall  chkconfig iptables on  # and reboot machine  reboot |

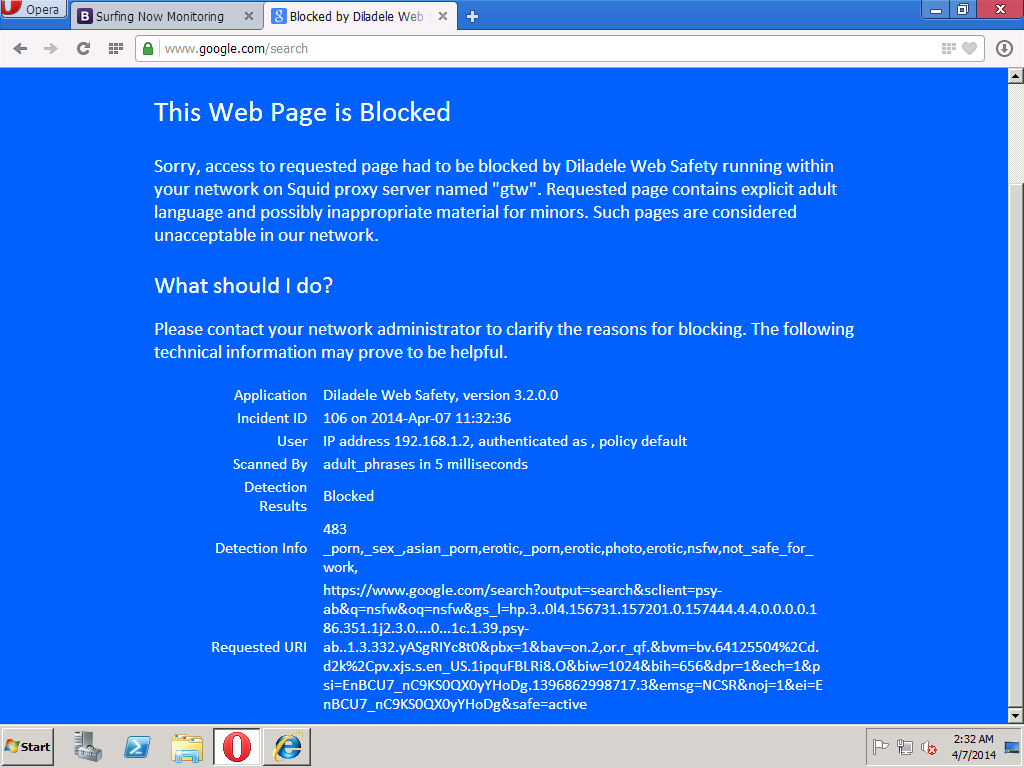
# Check HTTPS is Transparently Filtered

Please note, in order for HTTPS filtering to function correctly we **must** install the proxy certificate from /etc/opt/quintolabs/qlproxy/myca.der into Trusted Root Certification on all workstations in our network. The following screenshots show that HTTPS requests were decrypted and filtered transparently.





Browsing to Google and searching for an adult term (e.g. NSFW) we get the HTTPS request filtered and blocked transparently.



# Resume.

We now have the default gateway in our network capable of transparently filtering HTTP and HTTPS traffic. All workstations in our network trust the root certificate from proxy and thus get their HTTPS request decrypted and filtered. Browsing environment in our network became much safer.

# Links

1. [Archive with all scripts mentioned in this HOWTO](scripts.centos6.zip)
2. [Diladele Web Safety web site](http://www.quintolabs.com/)
3. [Online Documentation of Diladele Web Safety](https://github.com/ra-at-diladele-com/qlproxy_external/wiki/Administrators-Guide)
4. [Squid Proxy Wiki](http://wiki.squid-cache.org/)