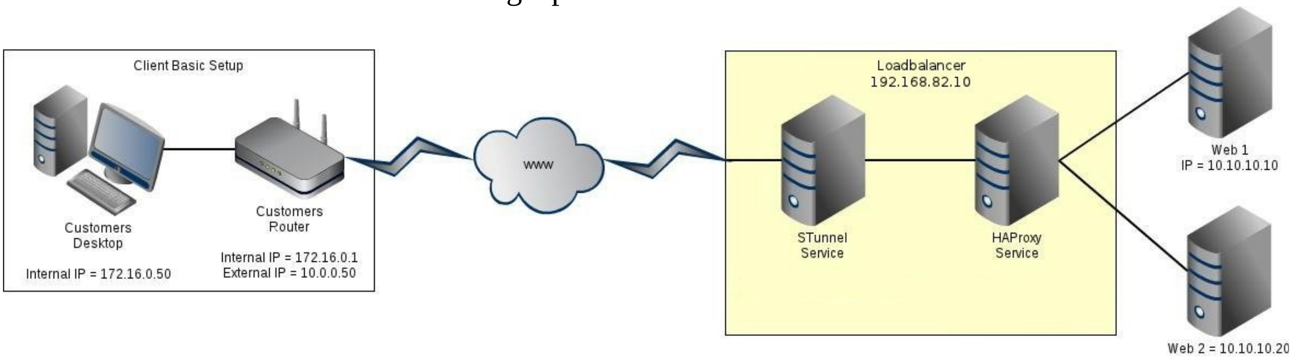
# *Setting up STunnel with HAProxy in Transparent mode on Centos 6.2*

1. Network Topology Diagram
   This is the overall planned network topology that should be achieved at the end of this document. The loadbalanced website with its SSL will be accessible on https://192.168.82.10 the physical loadbalancer server will run on 192.168.82.9 both of these IP addresses are configured on the units primary ethernet port (eth0), the secondary ethernet port (eth0) will be configured with 10.0.0.1/24 and will then loadbalance two Debian Webservers each running Apache 2.



2. Loadbalancer Software Configuration
   The Loadbalancer will be running Centos6.2 64bit minimum installation, you will most likely need to run a '*yum update*' first to ensure that the system is fully updated.
   Before installing any of the other software there are some key items that also need to be installed. These can be installed as follows:

   *yum install make wget gcc pcre-static pcre-devel*

3. Installing HAProxy
   I'm using the latest Development version of HAProxy for this build which at the time of writing is HAProxy 1.5 dev7 and is not available via the repository so needs to be downloaded and installed manually. These steps should enable you to do just that.

   i. *wget http://haproxy.1wt.eu/download/1.5/src/devel/haproxy-1.5-dev7.tar.gz (please note that this link may change if a new version becomes available so you may want to check http://haproxy.1wt.eu/download/1.5/src/devel/ first)*
   ii. *tar -zxf haproxy-1.5-dev7.tar.gz*
   iii. *cd haproxy-1.5-dev7*
   iv. *make TARGET=linux26 USE_STATIC_PCRE=1 USE_LINUX_TPROXY=1*
   v. *cp haproxy /usr/sbin/haproxy*
   vi. *cp examples/haproxy.cfg /etc/haproxy.cfg*
   vii. vim */etc/haproxy.cfg*

4. Setting Up HAProxy

   This should now in theory give you a working installation of HAProxy. However, first off all you need to set up the actual configuration file so that it knows what IP Address to listen on and what addresses it should be loadbalancing your traffic onto.

   For now just to test that we have a working service edit your HAProxy configuration file to match the following:

   *vim /etc/haproxy.cfg*

```
global
        daemon
        log /dev/log local4
        maxconn 40000
        ulimit-n 81000
defaults
        log global
        contimeout      4000
        clitimeout      42000
        srvtimeout      43000
listen http1
        bind 192.168.82.10:80
        mode http
        balance roundrobin
        server http1_1 10.0.0.10:80 cookie http1_1 check  inter 2000 rise 2 fall 3
        server http1_2 10.0.0.20:80 cookie http1_2 check  inter 2000 rise 2 fall 3
```

NOTE:This is only the most basic of setup options and will allow us to show that the HAProxy service is working correctly.

To start HAProxy type:

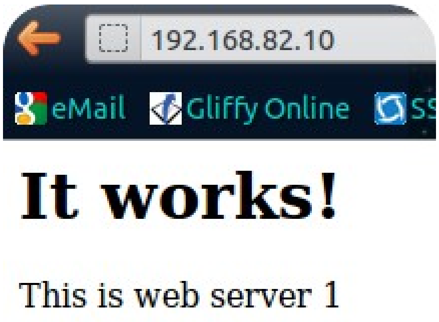
```
    /usr/sbin/haproxy -f /etc/haproxy.cfg
```

5. Web Server Test Pages

   Again keeping everything as simple as possible I just edited a very simple HTML page on each of my two web servers each with a line letting me know which Web Server I was looking at. So for example web server one has the page below:

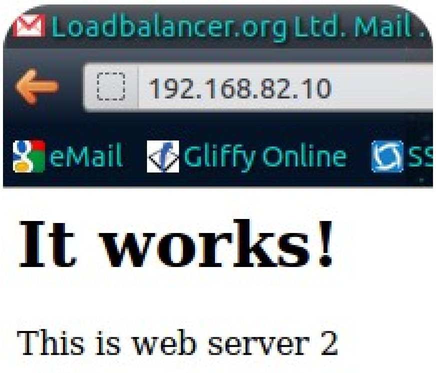
```
    <html>
        <body>
                <h1>
                        It works!
                </h1>
                <p>
                        This is web server 1
                </p>
        </body>
</html>
```

6. Testing It Works
   First of all if your running this on a new Centos Build you may want to check your 'iptables' rules as I forgot the first time around a simple '*iptables -nvL*' will show you if you have any rules in place.



It works!

This is web server 1



It works!

This is web server 2

   If you do not get one of your test pages showing and you have cleared any firewall rules you may find the source of your problem by stopping HAProxy with:

   *killall haproxy*

   And then starting HAProxy again with the debuging option enabled, this will then output all the connection events onto the console display. To start HAProxy in debug mode type:

   */usr/bin/haproxy -d -f /etc/haproxy.cfg*

   To stop the debuging simply press Ctrl+C

7. Setting HAProxy To Transparent Mode
   Assuming that you are now able to view your two web pages as outlined above you can see by looking at the Apache logs on either or both of your web servers that the IP Address that is reported to have connected to your website is the same address as your Internal Interface in my case 10.0.0.1 which is not very helpful for things like Webalizer or AWStats etc. so what we can do is set HAProxy into Transparent mode. First of all edit your 'haproxy.cfg' file so that the listen section looks like this:

```
listen http1
        bind 192.168.82.10:80
        mode http
        option http-server-close
        option  forwardfor
        source 0.0.0.0 usesrc clientip
        balance roundrobin
        server http1_1 10.0.0.10:80 cookie http1_1 check  inter 2000 rise 2 fall 3
        server http1_1 10.0.0.20:80 cookie http1_1 check  inter 2000 rise 2 fall 3
```

   You also need a set of firewall rules which I created a new file for so that these can be run as and when needed.

```
 vi /etc/hatpfw.sh
```

```
#!/bin/bash
iptables -t mangle -N DIVERT
iptables -t mangle -A PREROUTING -p tcp -m socket -j DIVERT
iptables -t mangle -A DIVERT -j MARK --set-mark 111
iptables -t mangle -A DIVERT -j ACCEPT
ip rule add fwmark 111 lookup 100
ip route add local 0.0.0.0/0 dev lo table 100
```

Save the file and then run it 'sh /etc/hatpfw.sh' now restart HAProxy as mentioned before. If you now browse to you site again and check the Apache logs you should see the address of your desktop and not the private network gateway address.

8. Installing STunnel

Again the CentOS repository does not have the latest version of STunnel which, at the time of writing is 4.53. To install the latest version of STunnel follow these steps:

i. yum install openssl-devel
ii. wget http://mirror.bit.nl/stunnel/stunnel-4.53.tar.gz
iii. tar -zxf stunnel-4.53.tar.gz
iv. cd stunnel-4.53
v. ./configure
vi. make
vii. make install

When asked for the details for the SSL Certificate enter the details as needed this will create a 1024bit local SSL certificate as '/usr/local/etc/stunnel/stunnel.pem'

STunnel is now installed. However, you will need to create the configuration file. A sample configuration file is available at '/usr/local/etc/stunnel/stunnel.conf-sample'. However a working version is shown below.

9. Writing The STunnel Configuration File

A sample configuration file that will pass encrypted HTTP traffic on our existing IP Address of 192.168.82.10 to our two web servers of 10.0.0.10 & 10.0.0.20 which should be written as '/usr/local/etc/stunnel/stunnel.conf' is as follows:

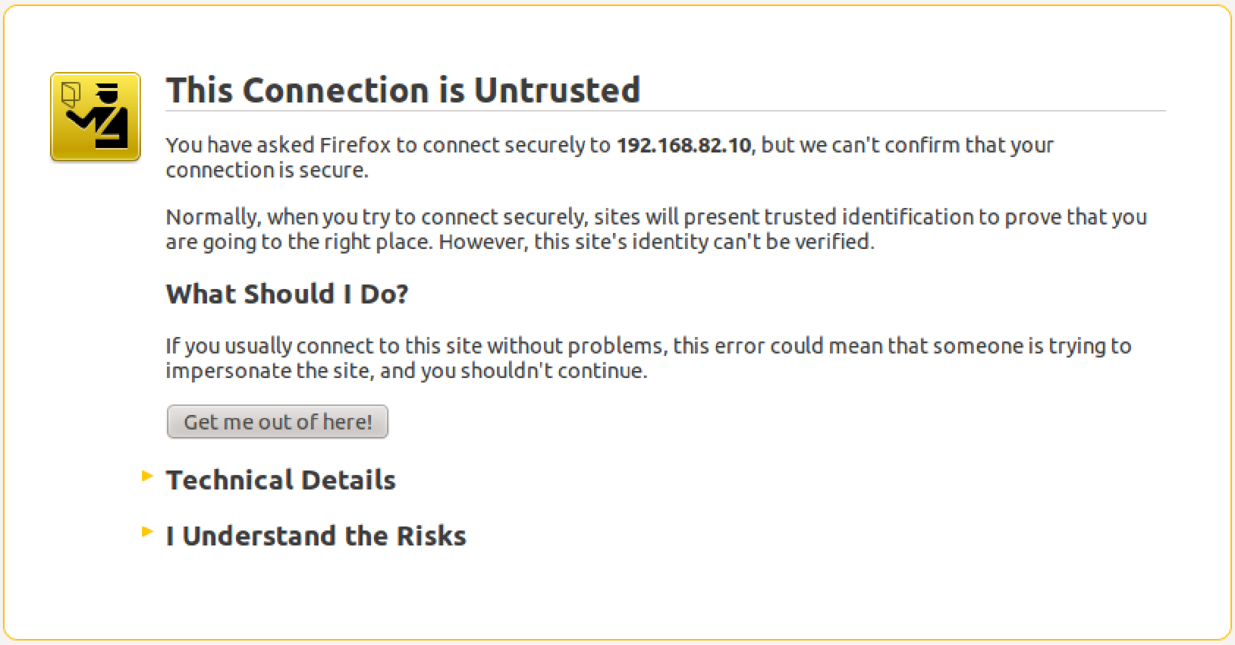
```
chroot = /usr/local/var/lib/stunnel/
setgid = nobody
pid = /stunnel.pid
cert = /usr/local/etc/stunnel/stunnel.pem
options = NO_SSLv2
[https]
accept  = 192.168.82.10:443
connect = 192.168.82.10:80
```

To start the STunnel Service type:

```
stunnel /usr/local/etc/stunnel/stunnel.conf
```

10. Testing SSL HTTP Connections

    If you now open a web browser and navigate to https://192.168.82.10 you should be shown a warning informing you that the SSL Certificate is not valid or from a trusted source



This Connection is Untrusted

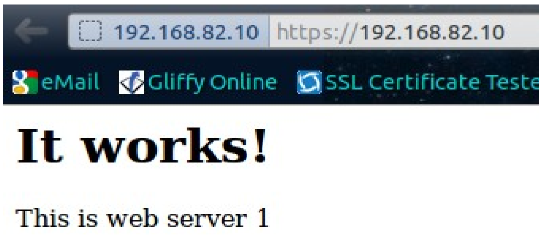
You have asked Firefox to connect securely to **192.168.82.10**, but we can't confirm that your connection is secure.

Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.

What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.

Click on the 'I Understand the Risks' and then the 'Add Exception' button, this will then show you a dialogue box where you should click on the 'Confirm Security Exception' button. This will then allow you to proceed to the Secure web page.



It works!

This is web server 1

11. Setting HAProxy & STunnel to Transparent Mode.

    Now that we know both HAProxy & STunnel are working together in both secure and none-secure mode we have the problem that when you access the site the logs show that the connection has come from the Loadbalancers IP Address in this case 192.168.82.10 which does not help us if you are using a traffic logging system.

    We can therefore enable Transparent Proxy mode in both HAProxy and STunnel so that we can see the IP Address of the person that is accessing our web server. To do this we need to change both of the configuration files that we created earlier.

    First of all add “accept-*proxy*” to the HAProxy configuration file after the “*bind 192.168.82.10:80*” so it looks like this:

```
listen http1
        bind 192.168.82.10:80 accept-proxy
        option http-server-close
        option  forwardfor
        source 0.0.0.0 usesrc clientip
        balance roundrobin
        server http1_1 10.0.0.10:80 cookie http1_1 check  inter 2000 rise 2 fall 3
        server http1_1 10.0.0.20:80 cookie http1_1 check  inter 2000 rise 2 fall 3
```

Finally add “*protocol = proxy*” to the end of the STunnel configuration file like so:

```
[https]
accept = 192.168.82.10:443
connect = 192.168.82.10:80
protocol = proxy
```

If you now restart both HAProxy and STunnel and browse to your website at https://192.168.82.10 you will see a secure page and looking at your Apache logs on either server you should see the address of the desktop computer that you browsed to this site from.
You should also note that you can not now browse to 192.168.82.10 on port 80 as this will now only accept connections from a proxy.

12. This is only a very basic setup of both HAProxy and STunnel. With both of these applications working together you can enable secure communications between yourself and a number of TCP products such IMAP, POP3, SQL etc. this is however, outside the scope of this document. For more information on either of the two OpenSource software titles used in this document please see their relevant websites:
    HAProxy:
    http://haproxy.1wt.eu
    STunnel
    http://stunnel.org