

### What is Reverse Proxy with SSL?

The Blue Coat ProxySG includes the basis for a robust and flexible reverse proxy solution. In addition to web policy management, content filtering, blocking, Web content virus-scanning and network protection, companies can implement a reverse proxy solution (with SSL) front ending their Web applications for greater security and Web performance.

Reverse proxy with the ProxySG provides the following advantages:

- The ProxySG terminates the session with the client and establishes another session with the Web server thereby offloading this process from the Web server.
- The Web server only sees the IP address of the ProxySG
- Granular policies with authentication, authorization and logging can be implemented
- The ProxySG has built in DOS (Denial of Service) protecting the Web servers from these types of attacks
- The server identity can be hidden by the ProxySG
- Increased performance with caching provides an improved Web experience

Moreover with SSL, the ProxySG terminates the SSL session with the client and forwards traffic to origin server via HTTP, offloading the Web server of this task. The ProxySG's flexible advanced forwarding architecture coupled with caching provides organizations a best-of-breed solution to leverage their network infrastructure.

### Why implement Reverse Proxy with SSL with Blue Coat?

Reverse proxy with SSL on the Blue Coat ProxySG provides flexibility to network administrators in defining scalable secured Web services. HTTPS connections are terminated on the ProxySG. The ProxySG obtains content (not currently in cache) from the origin server via HTTP. The following diagram presents this design.



# Implement Reverse Proxy with SSL

## There are four steps to implement Reverse Proxy with SSL on the ProxySG:

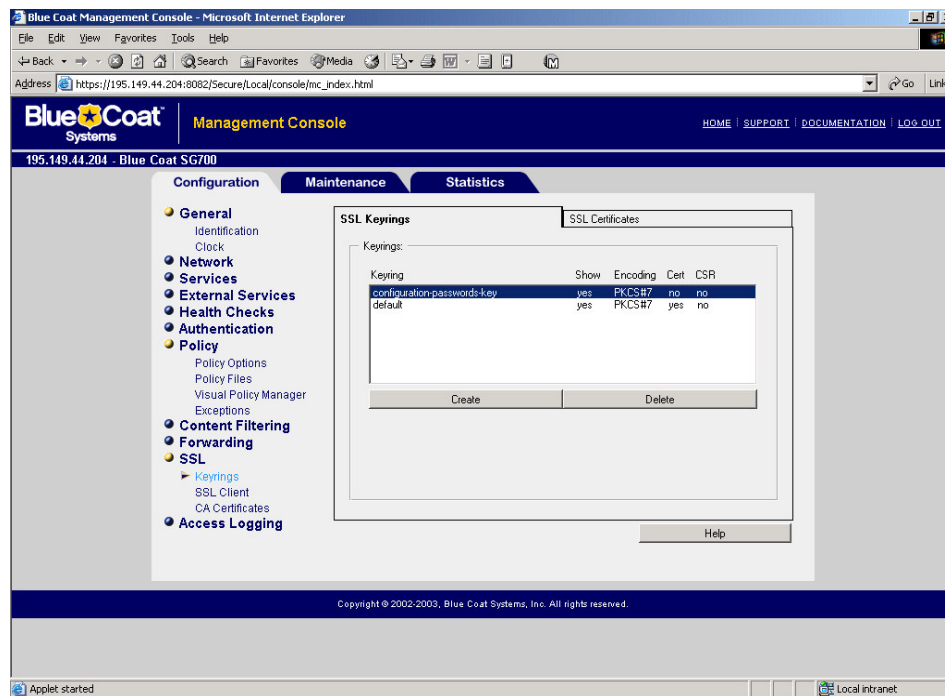
1. Configure the SSL keypair/certificate
2. Configure advanced forwarding hosts
3. Configure advanced forwarding rules
4. Test the configuration

**Note: In order to prevent the ProxySG from being used as an “open proxy” you can also implement these steps:**

1. **The default policy or first layer must be set to DENY**
2. **After DENY there must be a policy layer to explicitly allow the URLs to be forwarded**
3. **Then add the forwarding layer**

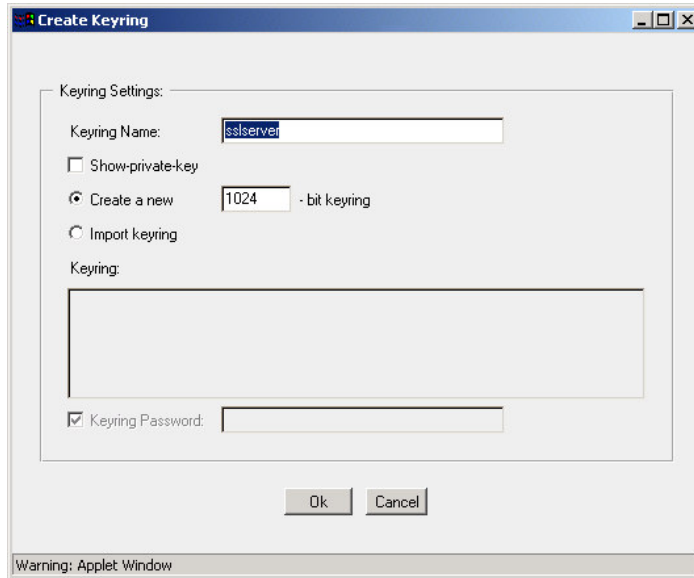
### Step 1 – Configure the SSL keypair/certificate

To configure the SSL keyring/certificate, open the Blue Coat Management interface on the ProxySG. Go to Configuration | SSL | Keyrings as shown in the following example.

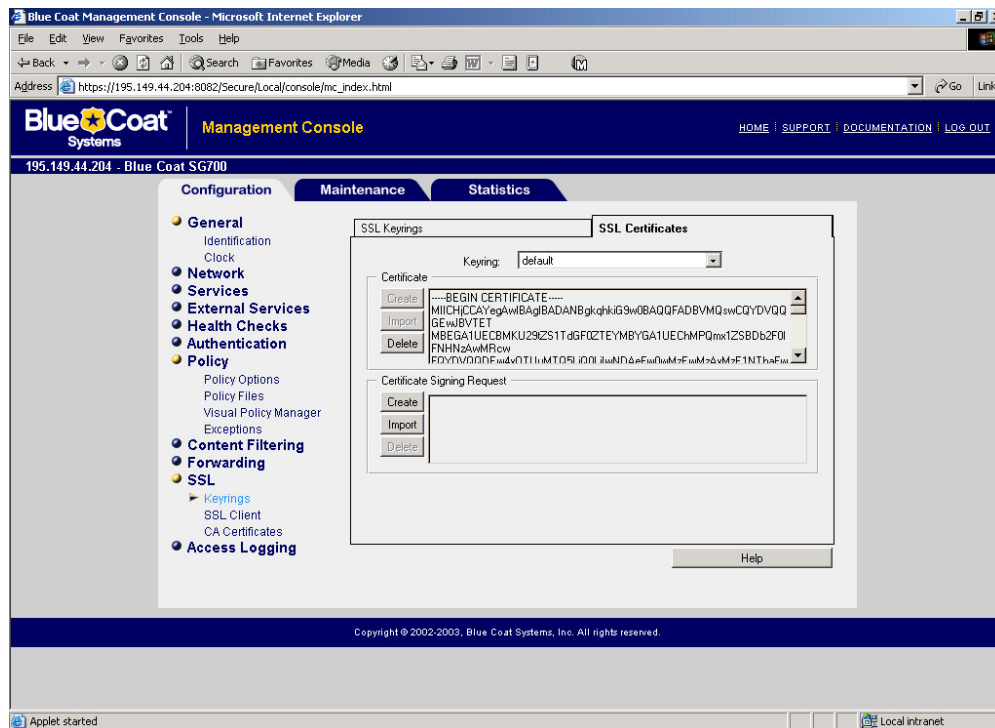


Click on Create.

In this example the keyring is called “sslserver”.



Click on OK, this will create the public/private key for this new keying. Now, a certificate needs to be created or imported for this keying. You could use a self signed certificate (which may cause warnings in the browser) or request a CertificateSigningRequest to a valid CA authority that will return you a signed certificate understood by browsers. To perform these tasks go into the SSL certificates tab. Select the keying we have just created (sslserver).



Click on Create to create the certificate signing request. If you select the certificate signing request, you will need to send that information to a CA authority that will return you a signed certificate to import into the ProxySG. In our example, we will create a self-signed certificate. Click on Create under certificate:

Fill in the corresponding information.

Create Certificate

State/Province: London Country Code: GB

City/Locality: London

Organization: Bluecoat

Unit: IT Department

Common Name: www.foo.com Challenge: abd3z2f

E-mail Address: yogi@bluecoat.com

Company: Bluecoat Systems

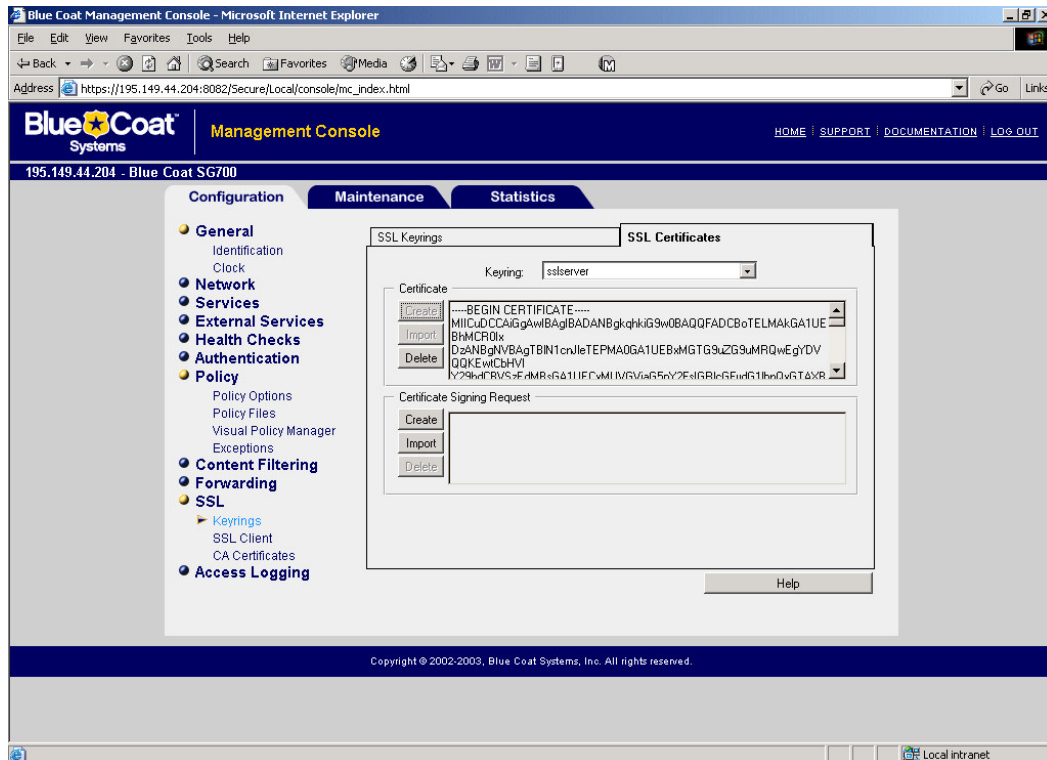
Ok Cancel

Warning: Applet Window

**Note: the common name is the URL that will be entered by the users – this is the name of the hostname of the SSL service**

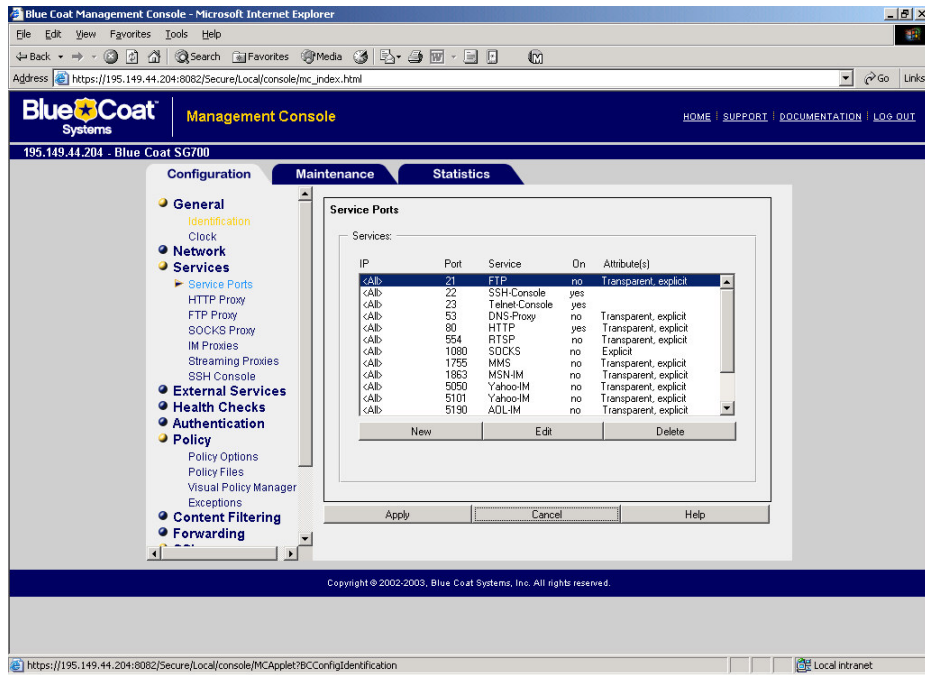
Click OK

You will see the following:



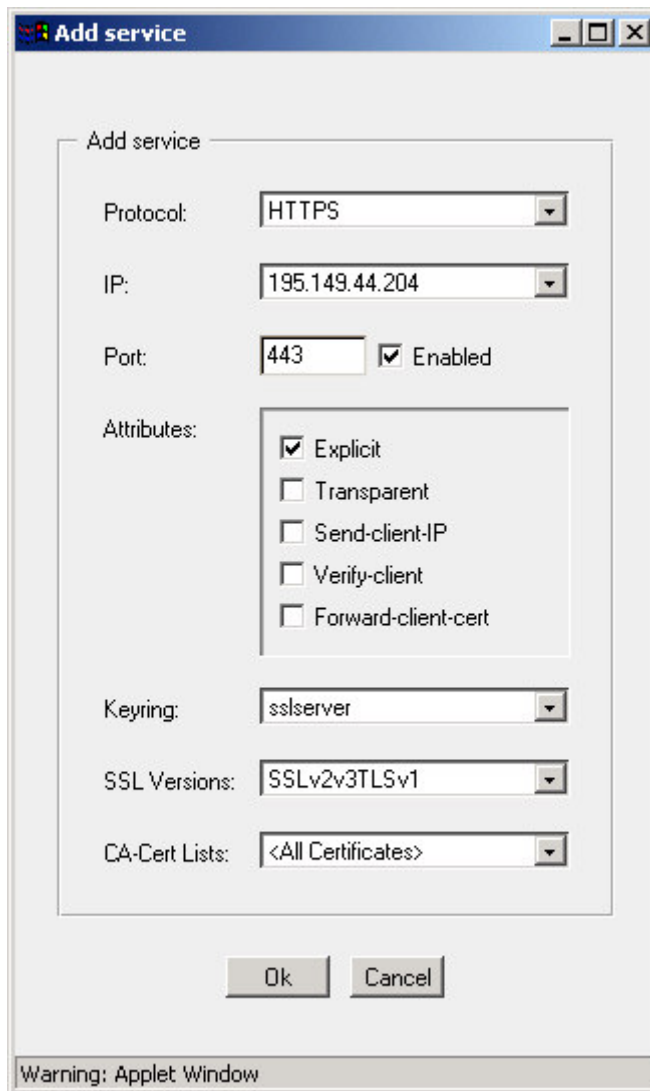
The next step is to assign this keying to the HTTPS service.

In the Blue Coat Management console, go into Services | Service Ports as shown here:



Click on create.

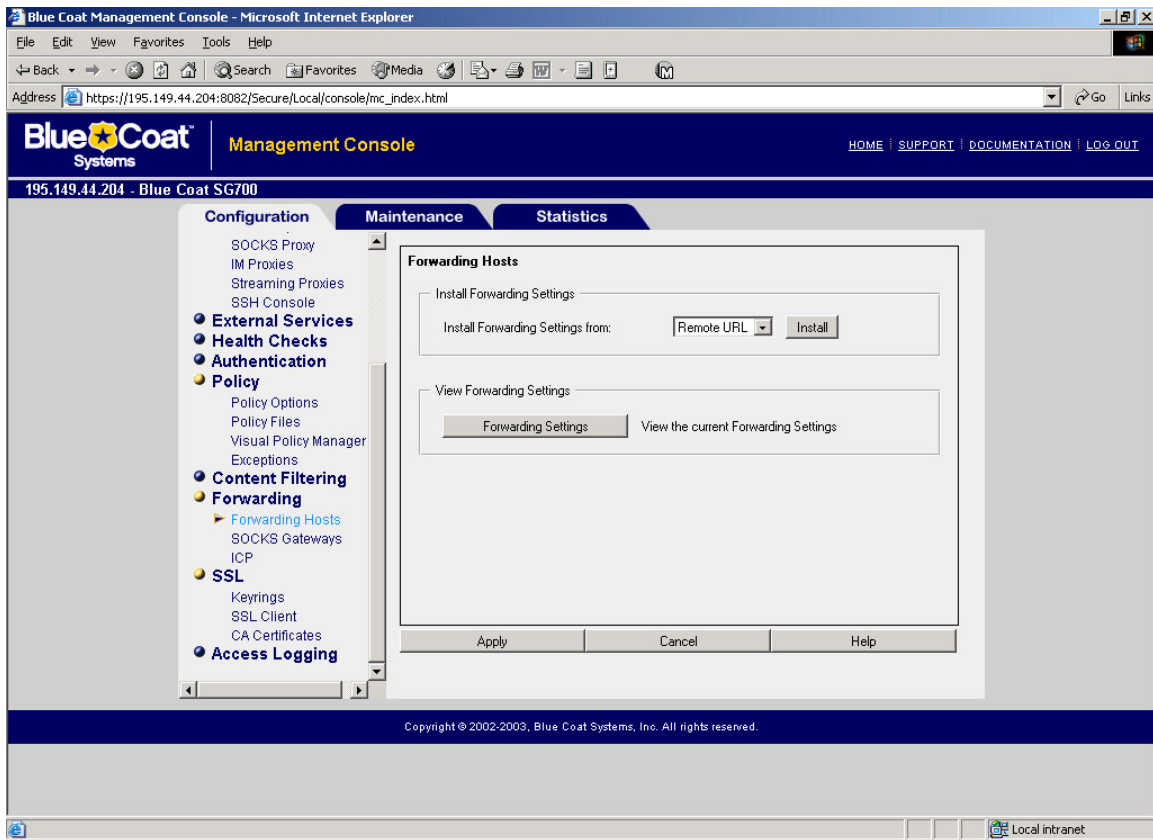
Select a HTTPS service; choose the port 443 as explicit and select the certificate we have created.



Click on OK and Apply.

## **Step 2 – Configure Advanced Forwarding hosts**

The second step is to define the back end servers to obtain the content from. This is defined in the forwarding hosts section of the configuration. Using the Blue Coat Management console, go to Forwarding | Forwarding Hosts as shown in the next example.

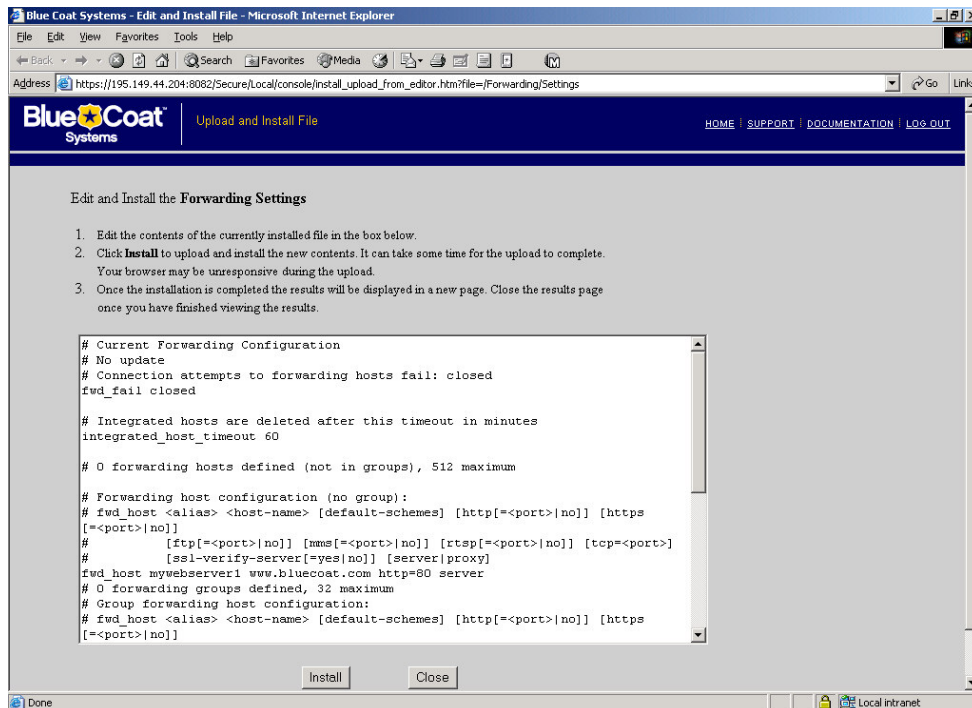


Open the text editor and add the following to the forwarding file:

```

fwd_host mywebserver1 <ipaddressofhtwebserver> http=80 server

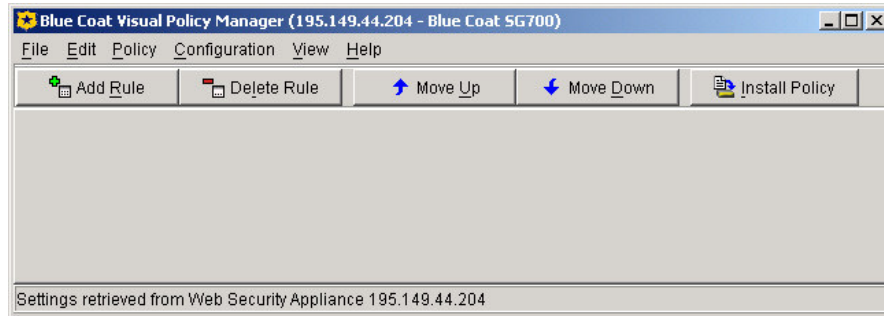
```



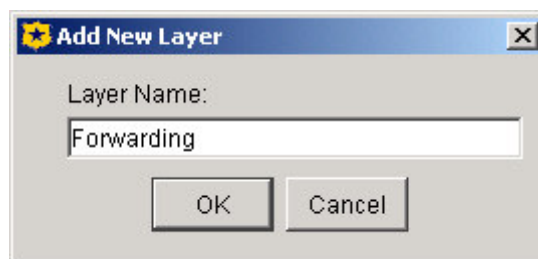
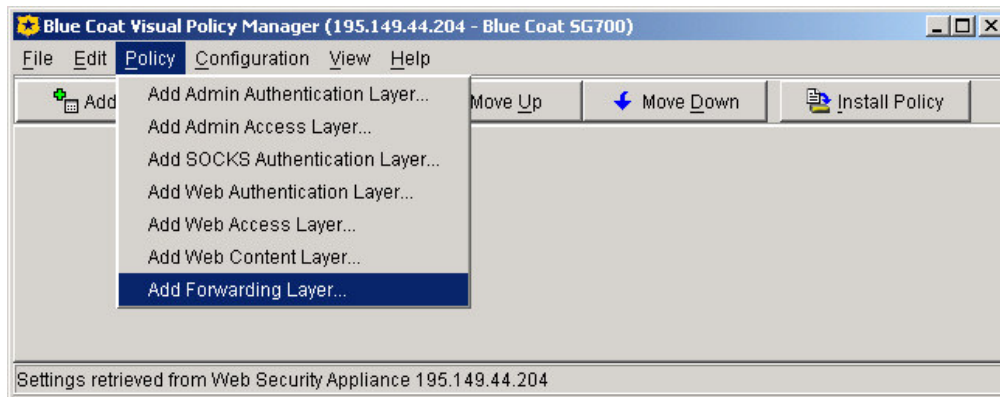
Click on Install.

### **Step 3 – Configure Advanced Forwarding rules**

The advanced forwarding rules are implemented via the Blue Coat Visual Policy Manager. Open the Visual Policy Manager on the ProxySG as shown here.

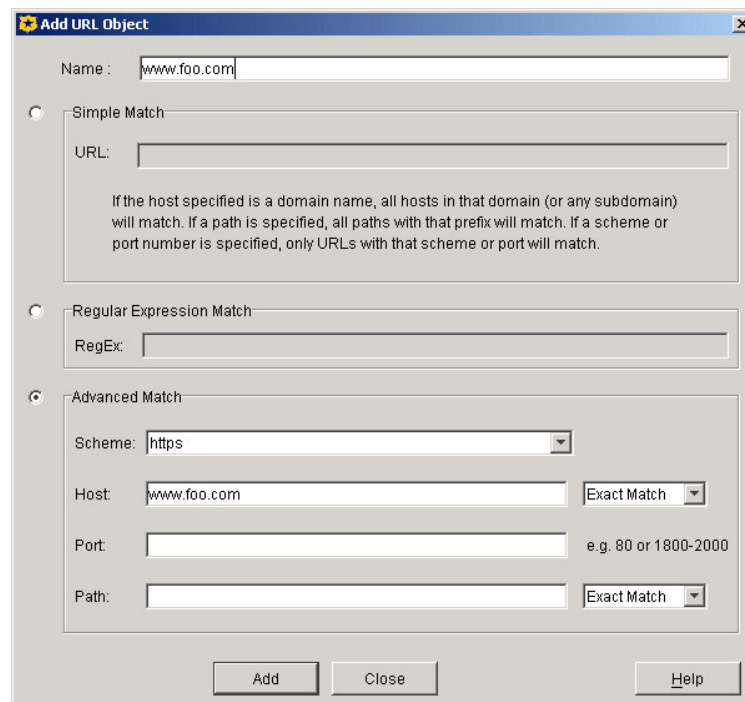
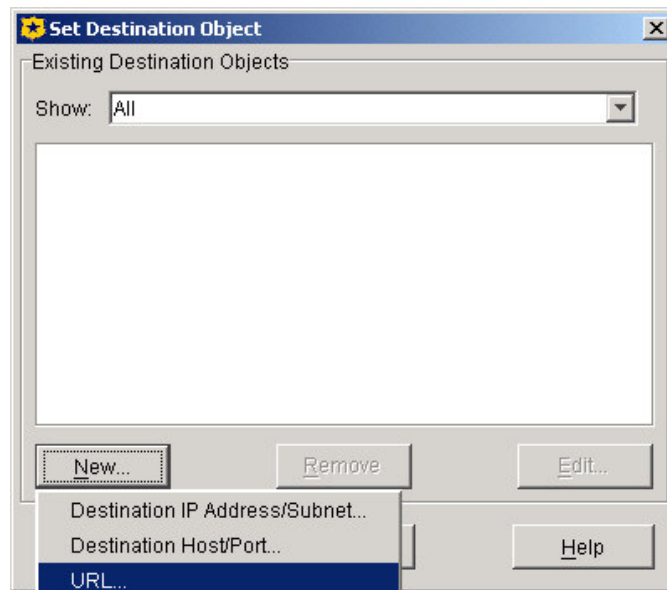
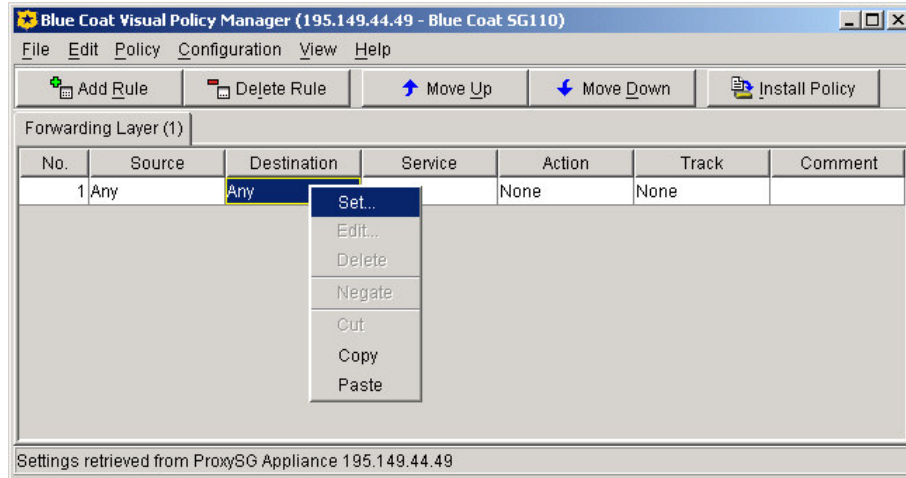


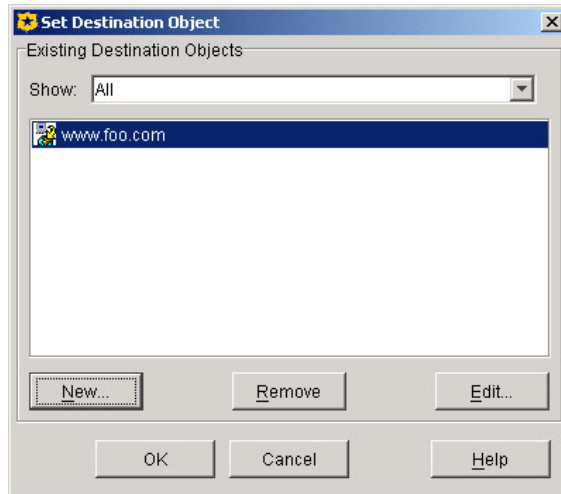
Create a Forwarding Layer called "forwarding"



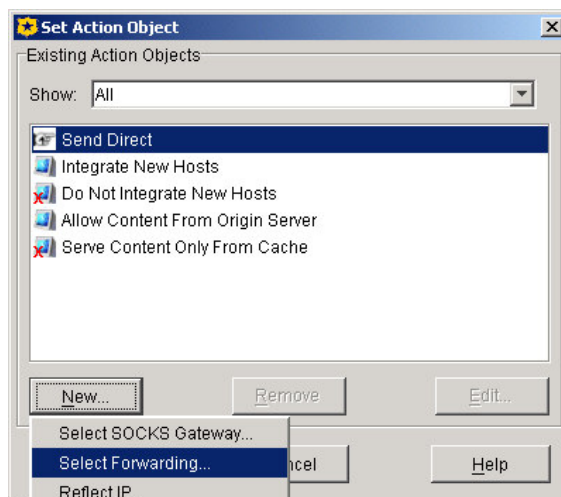
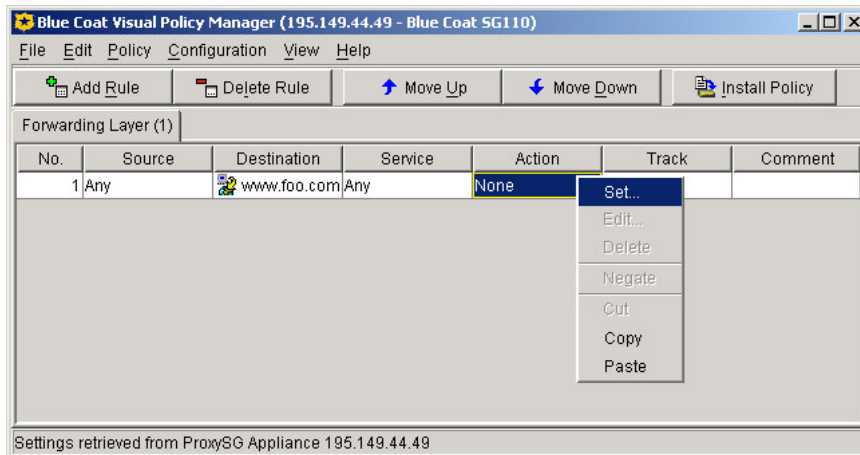
Next, create a forwarding rule with the following policy criteria. Examples of each step of this configuration are shown in subsequent screens.

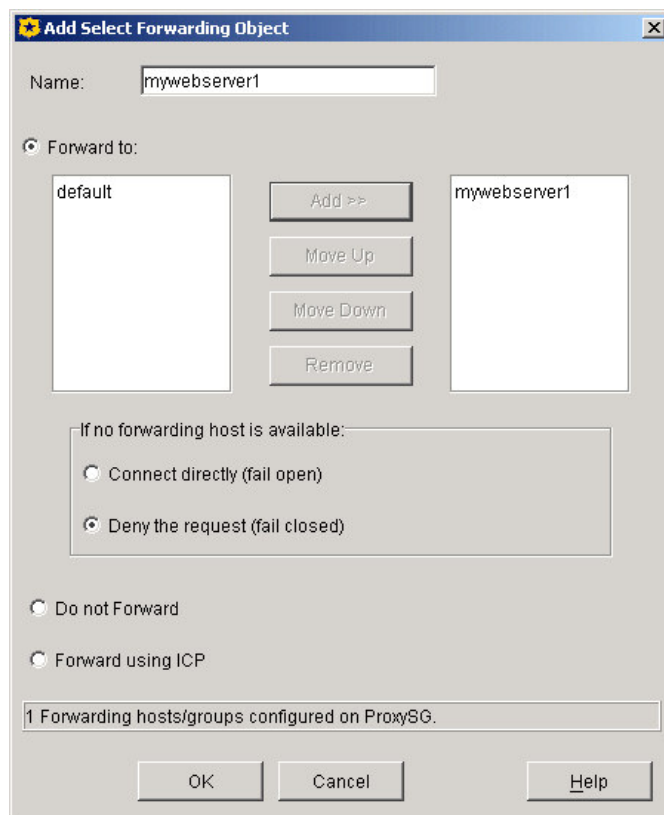
```
Source= any
Destination = protocol scheme = https and host : www.foo.com (what
users will type in their browsers ie url seen by users)
Service = any
Action = Select Forwarding
Time = any
Tracking = none
```



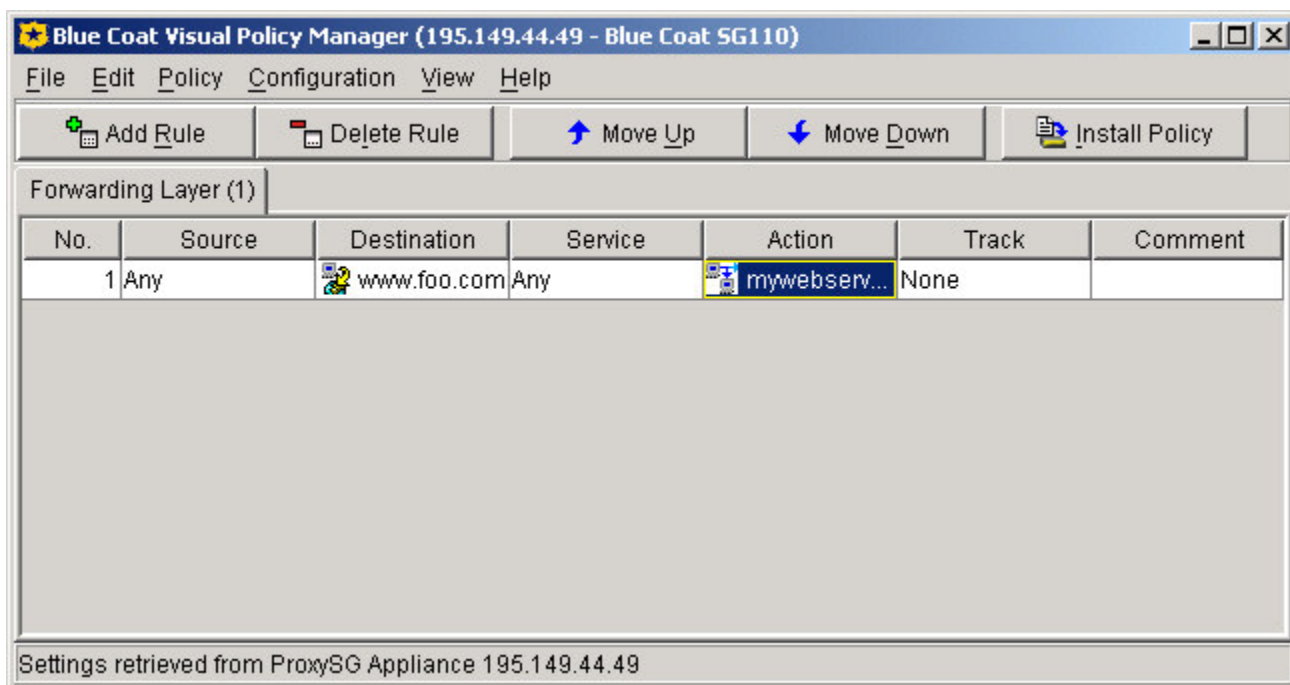


After you have completed the entry of this information, click on OK. Next, select the Action tab to set the parameters for this step.





Click OK twice.

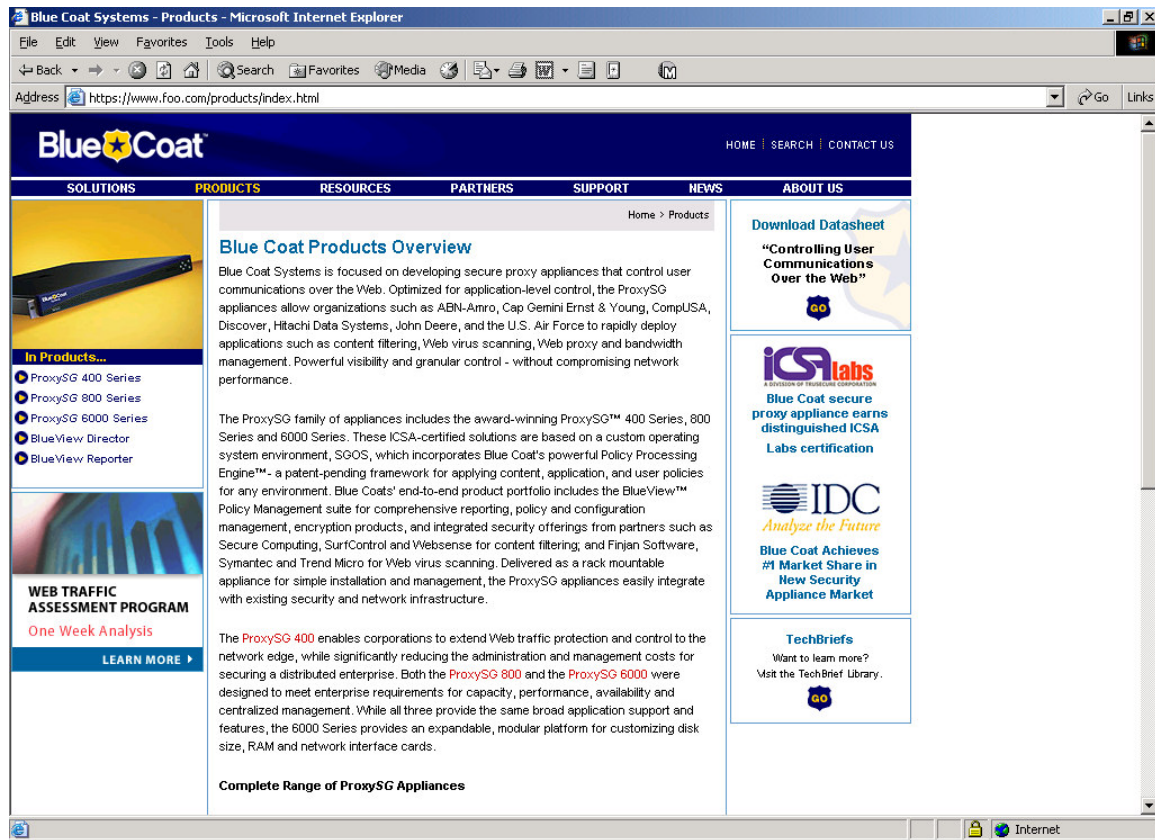


Finally, click on Install Policy to install the policy to the ProxySG. In this example users requesting <https://www.foo.com> will get content from <http://www.bluecoat.com>

**Note:** you will need to make sure that [www.foo.com](http://www.foo.com) resolves to the IP address of your ProxySG.

## Step 4 – Test your configuration

The last step is to validate that reverse proxy with SSL is working. Be sure to resolve [www.foo.com](http://www.foo.com) to the IP address of your ProxySG and request with a Web browser <https://www.foo.com>. You will see the [www.bluecoat.com](http://www.bluecoat.com) site secured.



## Caveats

It is possible that the origin Web server references links to HTTP instead of relative links. This may cause certain pages to reference the origin Web server. In this case you will need to install the following local policy that uses the **TwoWayURLRewrite** function:

```
<Proxy>
url.host=www.foo.com action.bluecoat_server_portal(yes)
; This transformation provides server portaling
define url_rewrite bluecoat_portal
caseless subst_embedded "https://www.foo.com/" "http://www.bluecoat.com/"
end
; This action runs the transform for bluecoat server portaling for http
; content
; Note that the action is responsible for rewriting related headers
define action bluecoat_server_portal
; request rewriting
```

```
rewrite( url, "^https://www\.foo\.com/(.*)", "http://www.bluecoat.com/$(1)",
cache, server )

rewrite( request.header.Referer, "^https://www\.foo\.com/(.*)",
"http://www.bluecoat.com/$(1)" )

; response rewriting
transform bluecoat_portal
end
```

## Conclusion

The ProxySG provides powerful reverse proxy capabilities allowing an organization to terminate SSL on the ProxySG and make content requests in behalf of the user to the origin server. An enterprise can achieve greater performance and security benefits by implementing a reverse proxy with SSL solution on their network.