DELIVER SITESPRING SECURELY WITH APACHE

Use simple reverse proxy technology to secure Macromedia Sitespring

January 31, 2002
Revision: 1.2

Brian Ghidinelli
Principal, Technical
vFive Digital Services
http://www.vfive.com
brian@vfive.com

Deliver Sitespring sites using Apache on Unix or Windows

Providing versioning control through integration with Microsoft Windows NTFS, Sitespring only supports installation on Microsoft Windows NT/2000/XP. For many system and network administrators, Windows servers may not be accessed from external networks for security reasons. This white paper describes a simple Apache solution that allows secure access to your Sitespring server from any web server capable of running the freely available Apache, such as Linux, Solaris or Windows.

Macromedia currently has a technote

http://www.macromedia.com/support/sitespring/ts/documents/sitn_usingiis.htm

explaining how to deliver Sitespring sites via an IIS connector acting as a stub from one Windows server to another. Unfortunately, this does not cover Unix-based networks. Our use of simple Apache reverse-proxy technology makes available a more universal solution using freely available software and minimal configuration time.

What is a Reverse Proxy?

Proxies and reverse proxies act as middlemen in a conversation between a client and a server. A proxy typically behaves as a cache for many users on a network to share resources to another network (such as the Internet). A reverse proxy however, acts as a buffer between the client and server to protect against unauthorized or unmonitored access to your servers. We will show you how to use an Apache reverse proxy to allow secure access from the Internet to an internal Sitespring server.

Reviewing Our Network Architecture

Before we set up Apache, let's review the network architecture at vFive's office in San Francisco and how we configured our Sitespring server. Your values will likely vary, but it should be easy to relate our setup to yours.

Development Firewall / Web Server:	66.125.189.161-165
Internal Sitespring Server:	192.168.1.20
Sitespring Intranet:	http://192.168.1.20:8400/
Sitespring Project Sites:	http://192.168.1.20:80/

192.168.1.20 is a private IP address, inaccessible to the general Internet. We decided to place project sites on port 80 to simplify access for clients who may not be savvy enough to remember a port number. We also set up a DNS record, *extranet.vfive.com* to point to 66.125.189.164, for convenience.

Installing and Configuring Apache

Apache is available from *http://httpd.apache.org* and can be downloaded and used free of charge. The latest version as of this writing is 1.3.23. The Windows version is not officially considered as stable as its Unix counterpart, but we have used it with great success. Since your Sitespring installation is probably not overloaded with requests, we feel that using Apache on a secured Windows server is acceptable. If you are concerned about stability, you may wish to wait until Apache 2.0 is released in the near future.

Since a solution already exists for Windows/IIS based servers, we will write this procedure from the viewpoint of a Unix administrator. If you are installing Apache on Windows, consult the official Apache documentation. The first step is to unpack the Apache distribution archive and switch to the Apache directory (we have removed the output of these commands for clarity):

```
[root@vfive]# tar zxvf apache_1.3.23.tar.gz
[root@vfive]# cd apache_1.3.23/
```

Configure Apache to include the module **mod_proxy**. We install our Apache at the default location and enable shared modules. If you have other needs such as **mod_rewrite**, **mod_usertrack** or a custom install directory, append them to the following configure command:

```
[root@vfive apache_1.3.23]# ./configure --prefix=/usr/local/apache
--enable-module=so --enable-module=proxy
```

Now compile and install Apache:

```
[root@vfive apache_1.3.23]# make
[root@vfive apache_1.3.23]# make install
```

You might be surprised if this is your first foray with open-source software, but Apache is now installed! All that is left is to modify the configuration file and specify what ports we should pass (using **mod_proxy**) to our Sitespring server:

```
[root@vfive apache_1.3.23]# cd /usr/local/apache/conf
```

Open *httpd.conf* in your favorite text editor and scroll to the bottom. This is where your virtual host configurations go and where you will set up the reverse proxy. We added the following configuration directives to our *httpd.conf*:

```
Listen 66.125.189.164:8400
<VirtualHost 66.125.189.164:8400>
        ServerName extranet.vfive.com

        ProxyRequests On
        ProxyPass / http://192.168.1.20:8400/
        ProxyPassReverse / http://192.168.1.20:8400/
        NoCache *

        CustomLog /usr/local/apache/logs/rproxy.access combined
        ErrorLog /usr/local/apache/logs/rproxy.errors
</VirtualHost>

Listen 66.125.189.164:80
<VirtualHost 66.125.189.164:80>
        ServerName extranet.vfive.com

        ProxyRequests On
        ProxyPass / http://192.168.1.20/
        ProxyPassReverse / http://192.168.1.20/
        NoCache *

        CustomLog /usr/local/apache/logs/rproxy.access combined
        ErrorLog /usr/local/apache/logs/rproxy.errors
</VirtualHost>
```

A brief recap of the setup: we told our web server to `Listen` on the IP address for extranet.vfive.com. For simplicity, Apache will listen on the same ports that we proxy to Sitespring: 80 and 8400. For each `VirtualHost`, we have enabled mod_proxy through the `ProxyRequests` directive.

The magic belongs to the `ProxyPass` and `ProxyPassReverse` directives. `ProxyPass` and `ProxyPassReverse` act as the middlemen in sending requests to and from the public Internet to our private network. `ProxyPass` forwards the initial request from the client to our internal Sitespring server and `ProxyPassReverse` receives the response and then sends it back to the client. Working together, they transparently provide access to a private network server from anywhere on the Internet.

The `NoCache` directive ensures that content is not cached between requests.

Now start Apache:

```
[root@vfive apache_1.3.23]# /usr/local/apache/bin/apachectl start
```

That's it–you're now providing more secure access to your Sitespring server from the Internet.

Firewalls and Application Security

If you are running a firewall (such as Linux **IPChains** or **IPTables**) on or in front of your reverse proxy server, you may need to configure access to your external IP address and ports before the above setup will work properly.

In our case, we configured our firewall to accept connections on port 80 from the Internet and our private 192.168.* network. However, on port 8400, we decided to only allow connections from our private 192.168.* network. Clients anywhere will need access to the Sitespring project sites, but only internal staff will access the Sitespring intranet. As a result, we configured a more restrictive policy that prevents outsiders from reaching port 8400.

Unless you have off-site employees or contractors who must access the Sitespring intranet application, we recommend closing off remote access to that port.

Conclusion

If you can't or don't want to provide global, unmonitored access to your Windows Sitespring server, Apache's **mod_proxy** implements a simple and secure reverse proxy for your network with a minimum of installation and configuration effort.

About

A full-service firm founded in 1996, vFive provides strategy, design and technology consulting and services. Based in San Francisco, our designers and developers work side-by-side on tightly integrated media projects and web applications. Experienced working with both established brands and startups; vFive has completed projects for companies including Epson, Oxygen Media, AutoDesk/Discreet, Playboy and Burson-Marsteller.