1 Apache Redirections

Apache provides many ways of redirecting requests to different websites. The two main ones are probably mod_proxy and mod_rewrite, which will be briefly covered here. These modules allow us to access data that may not have been accessible otherwise, for example behind a firewall.

2 mod_rewrite

Mod_rewrite is an incredibly powerful module, and can do much much more than is covered here. This example will show how to redirect requests to another URL, perhaps on another port and/or host, perhaps running under a java servlet engine.

```
<VirtualHost 192.168.1.1>
    ServerName demo.example.com
    DocumentRoot /var/www
    RewriteEngine on
    RewriteRule   ^/demo$ \
       http://java.example.com:8081/servlet/ [P]
    RewriteRule   ^/demo/$ \
       http://java.example.com:8081/servlet/ [P]
</VirtualHost>
```

3 mod_proxy

Mod_proxy can be used in two main ways, either as a proxy server similar to squid, or as a reverse proxy to forward http requests to another host. This example is forwarding requests for /demo off to another host, using the reverse proxy mode.

```
<VirtualHost 192.168.2.1>
    ServerName proxy.example.com
    ProxyPass        /demo/ http://host.example.com:8081/demo/
    ProxyPassReverse /demo/ http://host.example.com:8081/demo/
</VirtualHost>
```

4 Combinations

This example shows how it is possible to use both mod_rewrite and mod_proxy to access a host behind a firewall without having to worry too much about having ports in the URL.

```
<VirtualHost 192.168.1.2>
    ServerName virtual.example.com
    ProxyPass        /internal/ http://firewall.example.com/
    ProxyPassReverse /internal/ http://firewall.example.com/

    RewriteEngine on
    RewriteRule   ^/$ http://virtual.example.com/internal/ [P]
</VirtualHost>
```

On the firewall box, you need something like the following rules. This assumes you are running Linux 2.4, and hence using iptables.

```
# port forward for web from external webserver to internal
/sbin/iptables -t nat -A PREROUTING -i eth1 -p tcp \
       --dport 80 -s web.example.com -j DNAT \
       --to 192.168.1.2
/sbin/iptables -t nat -A PREROUTING -i eth1 -p tcp \
      --dport 8081 -s web.example.com -j DNAT \
      --to 192.168.1.2
```

5 Conclusion

As you can see, the combination of mod_proxy and mod_rewrite allow you to a wealth of things that would otherwise be difficult with standard Apache. The ability to share data that is protected behind a firewall is often very useful, as you can share data without exposing it absolutely more than is required.