ProxyAbuse

Why do I see requests for foreign sites appearing in my log files?

An `access_log` entry showing this situation could look like this:

```
63.251.56.142 - - [25/Jul/2002:12:48:04 -0700] "GET http://www.yahoo.com/ HTTP/1.0" 200 1456
```

For this log line, the *200* code (second to last field in this example) indicates that the request was successful – but see below for an explanation of what "success" means in this situation.

This is usually the result of malicious clients trying to exploit open proxy servers to access a website without revealing their true location. They could be doing this to manipulate pay-per-click ad systems, to add comment or link-spam to someone else's site, or just to do something nasty without being detected.

It is important to prevent your server from being used as an open proxy to abuse other sites.

How can I prevent these requests from accessing the foreign server through my server?

First, if you don't need to run a proxy server, disable mod_proxy by commenting out its `LoadModule` line or setting `ProxyRequests off` in `httpd.conf`. Remember that disabling `ProxyRequests` does not prevent you from using a reverse proxy with the `ProxyPass` directive.

If you do need to have Apache act as a proxy server, be sure to secure your server by restricting access with a `<Proxy>` section in `httpd.conf`.

My server is properly configured not to proxy, so why is Apache returning a 200 (Success) status code?

That status code indicates that Apache successfully sent a response to the client, but not necessarily that the response was retrieved from the foreign website.

RFC2616 section 5.1.2 mandates that Apache must accept requests with absolute URLs in the request-URI, even for non-proxy requests. This means that even when proxying is turned off, Apache will accept requests that look like proxy requests. But instead of retrieving the content from the foreign site, Apache will serve the content at the corresponding location on your website. Since the hostname probably doesn't match a name for your site, Apache will look for the content on your default host.

In the above example, since `www.yahoo.com` is obviously not a valid virtual host on your system, Apache will serve the homepage content from your default (virtual) host. The size of the response (1456 in the above example) can be compared to the size of the corresponding page on your default site to confirm that the response was served locally and no proxying was involved.

But how can I be really sure that I am not allowing the abuse of other sites

You can try yourself to use your server as a proxy to access other sites and make sure that you get either a failure, or local content from your site. Among the ways to do this:

1. Configure your browser to use your web server as its default proxy server and then try to request foreign sites. You should get only your own website content back in reply.
2. Manually construct requests using telnet:

```
telnet yoursite.example.com 80
GET http://www.yahoo.com/ HTTP/1.1
Host: www.yahoo.com
```

Then press enter twice. If your server is properly configured, you should receive content from your own site and not Yahoo.

What about these strange CONNECT requests?

A variant of this problem is an `access_log` entry that looks like

```
63.251.56.142 - - [25/Jul/2002:12:48:04 -0700] "CONNECT smtp.example.com:25 HTTP/1.0" 200 1456
```

The `CONNECT` method is usually used to tunnel SSL requests through proxies. But in this case, the port 25 on the target shows us that someone is attempting to use our HTTP proxy to send mail (probably spam) to a foreign site.

Everything mentioned above applies equally to this case. But normally, as long as the proxy is disabled, Apache would respond to such requests with status code `405 (Method not allowed)`. The fact that a success status code is returned indicates that a third-party module is processing the `CONNECT` requests. The most likely culprit is php, which in its default configuration will accept all methods and treat them identically.

This isn't inherently a problem since php will handle the request locally and will not proxy to the foreign host. But it is still a good idea to configure php to accept only specific methods (using the php configuration setting `http.allowed_methods`) or have your php script reject requests for non-standard methods.

I don't like the idea of my server responding to requests for random hostnames, even if it serves local content. How can I deny these requests?

You can configure Apache to deny access to any host that isn't specifically configured by setting up a default virtual host:

```
NameVirtualHost *:80

<VirtualHost *:80>
  ServerName default.only
  <Location />
    Order allow,deny
    Deny from all
  </Location>
</VirtualHost>

<VirtualHost *:80>
  ServerName realhost1.example.com
  ServerAlias alias1.example.com alias2.example.com
  DocumentRoot /path/to/site1
</VirtualHost>
```

See also the CanonicalHostNames recipe.

Can't I just drop these requests entirely?

Apache is an HTTP server and responds to HTTP requests with HTTP responses. It does not simply drop requests on the floor, since this would make it difficult to debug problems with client-server interactions.

If you really want to send no response at all, the third-party module `mod_security` is able to drop requests. But the savings in resource usage will be minuscule.

Unfortunately, even if your server is properly configured, you may see this type of exploit attempt recur. Since the offending client is usually itself a compromised computer (or a botnet), there is little that can be done to stop them beyond assuring that your site does not act as an open proxy.