Rate Limiting with Nginx

Posted in **Pro Tip**

Nov 18, 2014

Do you manage a website? Does it have a login form? Can somebody brute force attack it with every common username/password combination until they find one that works?

For many small web applications, the answer to all of the above is, "yes". This is a security risk and the solution is rate limiting. Rate limiting allows you to slow down the rate of requests and even deny requests beyond a specific threshold. Unfortunately, for most busy web developers, rate limiting is often tossed into a large pile of "things I know I should do, but don't have time for".

Advanced rate limiting apps such as django-ratelimit exist, but if you use Nginx as a reverse proxy to your application, the solution is almost trivial.

Let's start with a typical Nginx reverse proxy config:

```
upstream myapp {
    server 127.0.0.1:8081;
}

server {
    ...
    location / {
        proxy_pass http://myapp;
```

```
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;
    }
}
```

To enable rate limiting simply add the following line to the top-level of your config file:

```
limit_req_zone $binary_remote_addr zone=login:10m rate=1r/s;
```

This creates a shared memory zone called "login" to store a log of IP addresses that access the rate limited URL(s). 10 MB (`10m`) will give us enough space to store a history of 160k requests. It will only allow 1 request per second (`1r/s`).

Then apply it to a location by adding a stanza to your server block:

```
location /account/login/ {
    # apply rate limiting
    limit_req zone=login burst=5;

    # boilerplate copied from location /
    proxy_pass http://myapp;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header Host $http_host;
}
```

Here we're rate limiting the `/account/login/` URL. The burst argument tells Nginx to start dropping requests if more than 5 queue up from a specific IP. As you may have noticed, there's some duplicated configuration between `location /` and this stanza. Unfortunately this is a necessary evil, but a good configuration management tool can help you avoid error-prone copy/pasting.

Reload Nginx with your new config and you're good to go. To verify it's working, this nasty little bash one-liner can throw out a bunch of HTTP requests quickly:

```
for i in {0..20}; do (curl -Is https://example.com/accounts/login/ |
head -n1 &) 2>/dev/null; done
```

You should only see `200` responses once per second and probably some `503` responses when the queued requests exceeded the burst value. Comparing the results to a regular URL can help you see the difference rate limiting makes.

Here's a full Nginx config with rate limiting enabled. Now go secure your sites!

About **Peter Baumgartner**

Peter is the founder of Lincoln Loop, having built it up from a small freelance operation in 2007 to what it is today. He is constantly learning and is well-versed in many technical disciplines including devops ...

→ View Peter's profile