Часто, когда сайт начинает становится более популярным, всякие нехорошие люди (ботов люди создают) пытаются его положить наплывом запросов.

Можно попытаться защититься стандартными решениями на Nginx. В качестве ответов сервера здесь выдается 444 код (закрывает соединение без отправки данных), но можно ставить тот, который посчитаете нужным.

1. Когда в запросах идут левые заголовки, то можно оставить только нужные, а по остальным не отвечать:

```
if ($request_method !~ ^(GET|POST|HEAD)$ ) {
return 444;
}
```

2. Если запросы идут внешне нормальные, но с разными http_referer (например, с взломанных сайтов), то можно вести черные список при помощи map.

2.1 Перед разделом server в vhost создаем map

```
map $http_referer $bad_referer {
default 0;
"~https://example.com" 1;
}
```

Аналогично example.com можно добавлять в новой строке и другие http_referer.

2.2 В разделе server создаем условие на проверку

```
if ($bad_referer) {
return 444;
}
```

Если в http_referer указан https://example.com, то он будет получать 444 от Nginx.

3. Если нужно ограничить количество запросов от IP-адресов, но некоторые адреса надо исключить, то можно применять limit_req со списком исключений (geo+map).

3.1 Формируем список перед разделом server в vhost

```
geo $limited_net {
default 1;
192.168.1.0/24 0;
}
```

Сеть 192.168.1.0/24 будет в исключениях.

3.2 Создаем map с $binary_remote_addr

```
map $limited_net $addr_to_limit {
0 "";
1 $binary_remote_addr;
}
```

$binary_remote_addr — параметр в Nginx, который отвечающий за адрес клиента.

3.3 Перед разделом server создаем условия по запросам. В данном случае создаем зону reqtest, где возможно отправлять с одного IP 10 запросов в секунду.

```
limit_req_zone $addr_to_limit zone=reqtest:10m rate=10r/s;
```

3.4 В location ставим limit_req с возможностью всплеска 50:

```
limit_req zone=reqtest burst=50;
```

Если нужно без задержек обрабатывать всплески, то надо добавить nodelay после burst. Если запросы выше всплеска, то будут отбрасываться.

4. Если надо ограничить количество одновременных запросов от IP-адреса, то можно применить

`limit_conn_zone.`

4.1 Перед разделом server создаем зону.

```
limit_conn_zone $binary_remote_addr zone=conntest:10m;
```

4.2 В разделе server указываем условия одновременных подключений. В данном примере 30.

```
limit_conn conntest 30;
```

Конечно, это всё защита не от серьезного DDOS’а, но в блокировке всяких мутных ботов может помочь. Плюс хорошо бы ещё добавить fail2ban для выявления и блокирования странных запросов в логах.