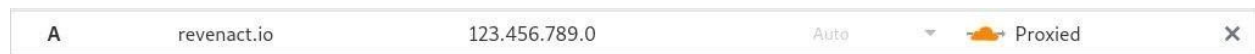


Introduction:

Welcome to the guide. Recently, there has been so many DDOS attacks that I feel compelled to provide these rules for everyone and fight against these nasty people. If you like it, please star my repository <https://github.com/tommytran732/Cloudflare-Firewall-Rules> or check out my website at <https://revenact.io>

The basics:

First of all, you have to make sure the orange cloud is ON. Not having the orange cloud means that your website's IP is exposed and there is nothing that Cloudflare will do to protect you.



You would also want to change the security level in Firewall -> Settings to High. Cloudflare has this set to Medium by default. The full explanation for the security levels can be founded at <https://support.cloudflare.com/hc/en-us/articles/200170056-What-does-Cloudflare-s-Security-Level-mean->, however, we generally want to challenge every IP that can be a potential threat.

DO NOT use the Under Attack Mode (unless you are truly under attack and it is causing you issues). In my experience, the Under Attack Mode is not particularly useful because if the botnet has JavaScript installed it will easily bypass the challenge and hit your website. Not only that, the Under Attack Mode will cause issues with APIs and a whole host of other problems.

Make sure Bot Fight Mode and Browser Integrity Check is on. You do not want botnets faking being legitimate crawlers and access your website.

Bot Fight Mode

Challenge requests matching patterns of known bots before they can access your site.

Requests matching Cloudflare-identified, non-legitimate automated traffic patterns will be challenged and/or blocked by Cloudflare.



[Help ▶](#)

Challenge Passage

Specify the length of time that a visitor, who has successfully completed a Captcha or JavaScript Challenge, can access your website. When the configured timeout expires, the visitor will be issued a new challenge. *Challenge Passage does not apply to Rate Limiting unless the rate limit is configured to issue a challenge.*

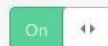
30 minutes ▼

[API ▶](#)

[Help ▶](#)

Browser Integrity Check

Evaluate HTTP headers from your visitors browser for threats. If a threat is found a block page will be delivered.



The firewall

This is the crucial part that a lot of users ignore/do not configure. These will help block a significant chunk of attacks and will keep your server's resource usage to a minimum.

Head to Firewall -> Firewall Rules

You can simply add my premade rules by creating a new rule and edit the expression:

[← Back to rules list](#)

Create Firewall Rule

Rule name

Give your rule a descriptive name

When incoming requests match...

[Use expression builder](#)

Paste my rules here.

Then...

Choose an action

The first rule we are going to add is the rules for known bots. We do not want to block good bots from accessing our website, so we have to make an exception for them.

Add the expression...

```
(cf.client.bot)
```

... and set the action to allow.

Edit Firewall Rule

Rule name

Give your rule a descriptive name

Known Bots

When incoming requests match...

Field

Operator

Value

Known Bots

equals

On

And

Or

Expression Preview

[Edit expression](#)

```
(cf.client.bot)
```

Then...

Choose an action

Allow

The list of known bots can be founded here:

<https://developers.cloudflare.com/firewall/known-issues-and-faq/#how-does-firewall-rules-handle-traffic-from-known-bots>

Now, we need to add a rule for emergency situation where every rule including the under attack mode fails. We create this rule right after known bots to give it high priority.

Give your rule a descriptive name

Emergency

When incoming requests match...

Field

Operator

Value

IP Address ×

is in

0.0.0.0/0 ×

And

Or

Expression Preview

Edit expression

(ip.src in {0.0.0.0/0})

Then...

Choose an action

Challenge (Captcha) ▾

Make sure this rule is disabled unless everything else fails to block the attack. It will force all users except known bots (due to it having lower priority) to solve hcaptcha in order to visit your website.

The next rule we are going to add is the rule for ASNs. We will challenge non residential IPs from accessing your website. Botnets usually contains a large number to servers from OVH, Online.net, etc.. so challenging these ASN will greatly help.

If you don't feel comfortable with people getting challenged, you can keep these rules off until you get an attack.

```
(ip.geoip.asnum in {3223 3561 3842 4250 4323 4694 5577 6724 6870
6939 7203 7489 7506 7850 7979 8075 8100 8455 8560 8972 9009 9370
10297 10439 11588 11831 11878 12989 13213 13739 14061 14127
14618 15003 15083 15169 15395 15497 15510 15626 15734 16125
16262 16276 16284 16397 16509 16628 17216 18450 18779 18978
19084 19318 19437 19531 19624 19844 19871 19969 20021 20264
20454 20473 20598 20738 20836 20860 21100 21159 21321 21859
22363 22552 22781 23033 23342 23352 24482 24768 24875 24940
24961 24971 25163 25369 25379 25780 25820 27257 28753 29066
29073 29182 29354 29465 29550 29691 29802 29838 29854 30083
30176 30633 30693 30900 30998 31034 31103 32097 32181 32244
```

32475	32489	32613	32780	33083	33302	33330	33387	33438	33480
33724	33785	33891	34305	34971	34989	35017	35366	35415	35470
35662	35908	35916	36024	36114	36290	36351	36352	36666	36873
36887	36920	36970	37018	37088	37153	37170	37209	37230	37248
37269	37280	37308	37347	37377	37472	37506	37521	37540	37643
37661	37692	37714	39020	39326	39351	39392	39572	40156	40244
40676	40824	40861	41653	41665	42160	42473	42708	42730	42831
43146	43289	43317	43350	44050	44066	45102	45470	45671	45815
46261	46430	46475	46562	46664	46805	46816	46844	47328	47447
47588	49349	49367	49453	49532	49544	49981	50297	50613	50673
51159	51167	51191	51395	51430	51731	51765	51852	52048	52173
52219	53013	53340	53559	53597	53667	53755	53850	53889	54104
54455	54489	54540	55225	55286	55536	55933	55967	56322	56630
56934	57043	57169	57230	57858	58073	58305	59253	59349	59432
59504	59729	59764	60011	60068	60118	60404	60485	60505	60558
60567	60781	61157	61317	61440	62217	62240	62282	62370	62471
62540	62567	63008	63018	63119	63199	63473	63949	64245	64484
132816	133296	133480	133752	134451	136258	197155	197328	198310	
199653	199883	200039	201011	201525	202053	202836	203523	203629	
204196	327705	327784	327813	327942	328035	394256	394330	394380	
395089	395111	395978	20248	44901	200904	53057	200532	50968	

135822 55293 57286 201200 24549 39458 200000 14576 54290 206898
60117 20448 201553 54825 31472 8556 29119 60476 25532 49949
51698 42442 11274 57345 54817 200019 53342 33569 201983 132425
197395 42699 31698 42612 29311 54527 63213 27175 13209 29140
27223 31659 49834 49693 30152 19133 198414 45201 31981 62605
61280 53332 61147 51109 19234 40438 58797 26978 29748 35974
262990 43021 42695 39704 62899 53281 59615 55761 52335 16973
196827 32647 14992 198968 196745 62071 132869 56106 32911 24931
57669 48896 45481 132509 39839 63129 53370 25048 28747 46433
55051 18570 13955 16535 22903 9823 46945 263032 36536 50986
199733 48825 35914 33552 52236 28855 198347 40728 18120 53914
12586 55720 27640 62563 202118 9290 45887 51050 20068 49485
40374 14415 46873 14384 54555 263237 20773 53918 4851 32306
133229 28216 36236 42210 51248 49815 34649 41562 33260 24220
52347 45486 33182 53055 51290 132225 133120 42776 55799 48446
263093 56732 42399 47385 40539 42244 29302 10929 47549 200147
393326 198171 57773 47583 43472 32338 9166 62082 198651 24725
29067 197902 42418 29097 196645 56110 23535 29869 62756 26484
25926 15189 20401 24679 25128 39756 32400 9412 9667 51294 23052
28099 45693 17881 17669 17918 50926 201634 22611 54641 61102
132071 10207 45577 132070 262603 29883 24558 38279 199997 50465
14120 11235 50655 17019 31240 199481 16862 47161 56784 59791
59677 202023 199990 50872 54839 58936 11230 62310 38894 47172
262287 46260 14442 133143 197648 39451 58922 27589 42400 133393
201597 28997 60800 33322 38001 199129 197372 57752 201670 14244
22152 34541 196678 43198 47625 42331 62049 35295 42311 53589
59705 36791 14160 34432 41062 59135 201630 25260 23108 40281
31590 10532 22720 27357 33070 45187 7595 26481 29713 13926 54203
62651 63128 62838 30849 14987 47577 54334 63916 50915 21217
59816 23273 59632 29452 59795 60739 15919 49313 57879 56617
62088 45179 27597 201702 32740 58667 12617 199847 25642 14567
35278 197914 41079 1442 43620 197439 198313 42705 44398 13909
34745 24958 17971 47143 59854 57682 3722 13647 20450 30235 47205
23881 198047 14986 17920 32275 50608 199213 262170 201862 43541
24381 10200 14708 27229 48093 42465 7598 30475}})

Set the action to Challenge (NOT JS Challenge - as we have discussed before, the JS Challenge is not very effective. We would want Google ReCaptcha Challenge). Of course, you can be stricter and set it to block instead.

Then...

Choose an action

Challenge (Captcha)

Because of the 4kb/rule limit, we will need to create a second ASN rule for the next set up ASNs:

```
(ip.geoip.asnum in {55229 7349 33251 52465 52270 45152 8477  
198153 52925 61412 262978 53225 41427 53101 41369 35467 59554  
52674 24611 48812 40715 201449 52321 29331 201709 53221 198432  
51241 19969 56799 26277 58113 28333 42120 6718 20692 17439  
132717 9925 132779 42622 6188 40819 24997 38107 36408 57363  
46177 62026 61107 58325 19604 23470 39287 12876 60033 13830  
60094 37963 31624 26496 48287 132839 204641 35913 39572 32592  
23969 9371}))
```

We should also challenge countries that your website does not get a lot of traffic from and is usually responsible for large attacks. Russia, China, and Tor is almost always the case. (Yes, Tor establishes itself as a “Country”. You can use this to challenge access from the Tor Network).

Create a new rule for country challenge:

```
(ip.geoip.country in {"T1" "CN" "RU" "ID" "BD" "KH" "BR" "SA"  
"LY" "ZA" "TH" "IN" "AR"}))
```

Give your rule a descriptive name

Country & Tor

When incoming requests match...

Field	Operator	Value	
Country ×	is in	Tor × China × Russian Federation × Indonesia × Bangladesh × Cambodia × Brazil × Saudi Arabia × Libya × South Africa × Thailand × India × Argentina ×	And Or

Expression Preview

Edit expression

```
(ip.geoip.country in {"T1" "CN" "RU" "ID" "BD" "KH" "BR" "SA" "LY" "ZA" "TH" "IN" "AR"})
```

Then...

Choose an action

Challenge (Captcha)

Of course, you should change this rule so that it fits your website.

Also, please keep in mind the order of the rules. The rules on top will have higher priority than the rules at the bottom. Known Bots should always have the highest priority.

If you use something like the Pterodactyl Panel, you will need to head to Firewall -> Tools and whitelist your nodes' IPs as well.

Managed Rules:

This is a really nice feature that Cloudflare Pro plan above has. I highly recommend that you check it out if you have a paid plan as it can turn out to be very useful.

Rate Limiting:

Head to Firewall -> Tools -> Rate Limiting and create a custom rule. Be sure to use * in your expressions.

The limit is highly dependant on your website. Some recommended values are: Plain

website: 100 connections/minute -> Block for 1 hour.

Xenforo: 250-350 connections/minute -> Block for 1 hour.

WHMCS: 150 connections/minute -> Block for 1 hour. Pterodactyl:

100 connections/10 seconds -> Block for 1 minute.

Edit Rate Limiting Rule

Rule Settings

Rule Name

Rate Limiting

If Traffic Matching the URL

http & https

revenact.io/*

from the same IP address exceeds

10

requests per

1 minute

Then

Block

matching traffic from that visitor for

1 hour

When "Block" is set, when the threshold is exceeded, the Client will receive a "429" error page until the Block time has expired.

Cancel

Save

Other mentions:

If possible, configure your firewall so that it only accepts connections from Cloudflare. The Layer 7 attacks will become less of a problem even if the attacker finds out the origin IPs because you only accept connections from Cloudflare anyways. The list of Cloudflare's IPs can be founded here:

<https://www.cloudflare.com/ips/>

If you can afford Cloudflare Argo, I highly recommend that you use it with an Nginx server binded to 127.0.0.1. That way, your website is completely inaccessible unless the traffic goes through cloudflare.

As a alternative, you can use my script here:

<https://github.com/tommytran732/Cloudflare-IPWhitelist>

Also, be sure to have no MX records pointing to the same address as your main website. Even with the orange cloud on, the MX record WILL Expose your Origin IP and that will lead to a whole host of other problems. Check the email header of the emails sent by your website as

well, you need to make sure that the Mail Server does not reference the IP of your website there.

Restore visitor's IPs:

As we are proxying behind cloudflare, all connections seen by the web server are from cloudflare. This will undoubtedly create problems with access logging. Make sure you set your web server up to restore the visitor's IP:

<https://support.cloudflare.com/hc/en-us/articles/200170786-Restoring-original-visitor-IPs-Logging-visitor-IP-addresses-with-mod-cloudflare->

Bonus rule - Referrer Check:

Sometimes, some attacks will go through all of the rules above (except the emergency rule), but you don't want to challenge everyone yet. You can check your web server's access log and see if all of the connections share anything in common.

For my case, I am using NGINX on RHEL/CentOS 8. I can watch the live logs using `tail -f /var/log/nginx/access.log`

As you can see, sometimes, you might see something like this (yes, this is a screenshot from an actual attack on my server):

```
92.245.142.215 - - [27/May/2020:13:26:44 -0400] "GET /tags?=SUCKITTHEBIGCOCK HTTP/1.1" 200 4281 "https://xnxx.com" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.130 Safari/537.36 OPR/66.0.3515.103" "148.251.36.51, 148.233.13.3, 148.251.34.37, 148.233.62.12,92.245.142.215"
95.79.99.148 - - [27/May/2020:13:26:44 -0400] "GET /tags?=SUCKITTHEBIGCOCK HTTP/1.1" 200 4283 "https://xnxx.com" "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3599.0 Safari/537.36" "148.233.48.53, 148.233.10.49, 148.233.41.66, 167.233.19.62,95.79.99.148"
110.39.187.50 - - [27/May/2020:13:26:44 -0400] "GET /tags?=SUCKITTHEBIGCOCK HTTP/1.1" 200 4282 "https://xnxx.com" "Mozilla/5.0 (Linux; Android 9; SM-G950F) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.116 Mobile Safari/537.36" "148.233.47.65, 167.233.60.0, 167.251.46.42, 167.251.24.27,110.39.187.50"
103.86.187.242 - - [27/May/2020:13:26:44 -0400] "GET /tags?=SUCKITTHEBIGCOCK HTTP/1.1" 200 4282 "https://xnxx.com" "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3599.0 Safari/537.36" "148.233.39.49, 167.251.8.41, 148.233.27.31, 167.251.21.57,103.86.187.242"
45.170.129.53 - - [27/May/2020:13:26:44 -0400] "GET /tags?=SUCKITTHEBIGCOCK HTTP/1.1" 200 4282 "https://xnxx.com" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36 Edge/18.18247" "148.251.57.57, 167.251.63.21, 167.251.57.2, 148.233.9.36,45.170.129.53"
78.140.7.239 - - [27/May/2020:13:26:44 -0400] "GET /tags?=SUCKITTHEBIGCOCK HTTP/1.1" 200 4281 "https://xnxx.com" "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3950.1 Safari/537.36" "167.251.27.48, 167.251.35.1, 148.251.2.16, 148.233.13.21,78.140.7.239"
92.245.142.215 - - [27/May/2020:13:26:44 -0400] "GET /tags?=SUCKITTHEBIGCOCK HTTP/1.1" 200 4281 "https://xnxx.com" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.130 Safari/537.36 OPR/66.0.3515.103" "148.251.36.51, 148.233.13.3, 148.251.34.37, 148.233.62.12,92.245.142.215"
59.221.54.114 - - [27/May/2020:13:26:44 -0400] "GET /tags?=SUCKITTHEBIGCOCK HTTP/1.1" 200 4282 "https://xnxx.com" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.87 Safari/537.36" "148.251.8.37, 167.233.48.60, 148.233.4.28, 148.233.27.21,89.221.54.114"
61.29.96.146 - - [27/May/2020:13:26:44 -0400] "GET /tags?=SUCKITTHEBIGCOCK HTTP/1.1" 200 4282 "https://xnxx.com" "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/68.0.3599.0 Safari/537.36" "167.233.15.38, 167.251.54.14, 148.233.22.34, 167.233.3.3,61.29.96.146"
124.41.211.251 - - [27/May/2020:13:26:44 -0400] "GET /tags?=SUCKITTHEBIGCOCK HTTP/1.1" 200 4280 "https://xnxx.com" "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3599.0 Safari/537.36" "148.251.59.38, 148.251.29.41, 167.233.68.49,124.41.211.251"
81.200.63.108 - - [27/May/2020:13:26:44 -0400] "GET /tags?=SUCKITTHEBIGCOCK HTTP/1.1" 200 4283 "https://xnxx.com" "Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.145 Safari/537.36 Vivaldi/2.6.1566.49" "167.233.54.20, 148.251.54.29, 148.251.11.33, 148.233.47.15,81.200.63.108"
94.103.12.43 - - [27/May/2020:13:26:44 -0400] "GET /tags?=SUCKITTHEBIGCOCK HTTP/1.1" 200 4280 "https://xnxx.com" "Mozilla/5.0 (iPhone; CPU iPhone OS 13_2 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/13.0.3 Mobile/15E148 Safari/604.1" "167.233.9.43, 167.251.10.54, 167.233.36.33, 148.251.39.15,94.103.12.43"
43.252.145.50 - - [27/May/2020:13:26:44 -0400] "GET /tags?=SUCKITTHEBIGCOCK HTTP/1.1" 200 4283 "https://xnxx.com" "Mozilla/5.0 (Linux; Android 9; SM-G950F) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.116 Mobile Safari/537.36" "148.233.19.35, 167.233.20.42, 167.251.39.13, 167.233.3.69,43.252.145.50"
36.89.181.161 - - [27/May/2020:13:26:44 -0400] "GET /tags?=SUCKITTHEBIGCOCK HTTP/1.1" 500 572 "https://xnxx.com" "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3599.0 Safari/537.36" "167.233.17.64, 148.233.5.55, 148.233.34.63, 148.233.34.41,36.89.181.161"
181.57.198.102 - - [27/May/2020:13:26:44 -0400] "GET /tags?=SUCKITTHEBIGCOCK HTTP/1.1" 200 4285 "https://xnxx.com" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.102 Safari/537.36 Edge/18.19008" "148.233.36.70, 167.251.46.2, 148.233.7.60, 148.233.37.59,181.57.198.102"
```

In this case, you can easily see that the attacker is using "<https://xnxx.com>" (Warning: NSFW) as the referrer. All we need to do is to block (or challenge) referrers that have some certain words that the attacker is using:

```
(http.referer contains "what-risks-are-introduced") or
(http.referer contains "suck") or (http.referer contains "boner")
or (http.referer contains "gay") or (http.referer contains "porn")
or (http.referer contains "xnxx") or (http.referer contains
"corona") or (http.referer contains "sex") or (http.referer
contains "fuck") or (http.referer contains "chin")
```