Using CDN for Entire Website and Country Blocking - Part 3

Posted At : November 29, 2018 5:14 PM | Posted By : Wil Genovese
Related Categories: ColdFusion, Coldfusion Security, Hosting and Networking, Security, AWS

This is Part 3 in a short series of articles about blocking entire countries from a website. Parts one and two cover **CloudFlare** and **CloudFront**.

CF Webtools has been asked numerous times to block an entire country or countries by many clients. The issue is that there's a lot of hacker activity from certain identified countries and the client(s) does not do any business with those countries. Typically it's entire server hacking attempts, but more recently it's to use the client's shopping cart to "test" stolen credit cards. This is a very serious problem and as such clients are asking us to help them prevent this from happening. One potential solution is to block the IP addresses that these attacks are coming from. I refer to this as the Whack-A-Mole method because it's just like that arcade game. As soon as you block one IP they switch to another IP address.

We need a better solution. I looked into what we could do and how reasonable and feasible the various options are in terms of technology and cost. In my previous two articles I wrote about using CloudFlare and AWS CloudFront. In this article I'm writing about using a slightly better hammer in the Whack-A-Mole method to block entire countries. This is one of the simplest but also least effective methods.

The option many of us have traditionally done is blocking problematic IP's on a case by case basis and in extreme cases blocking entire IP ranges. I've often referred to this as the Whack-A-Mole method. It's reactive and not proactive. A real hacker would not use their own personal IP and there is no guarantee that the IP will always remain with an unscrupulous user. Normally I do not block an IP because bad stuff happened from that IP once. However, I have noticed the same IP or IP ranges launching attacks on multiple unrelated, hosted at different locations, and different client's servers. That's when I start pounding the IP with the ol' Ban Hammer! Also, blocking and entire country with this method would mean being able to know all the possible IP addresses or address blocks assigned to a particular country. This is knowable!

I did some research on this and found a few very helpful resources. Resources like this **http://ipdeny.com/ipblocks/** and this **https://www.sitepoint.com/how-to-block-entire-countries-from-accessing-website/**. These sites keep an updated list of IP addresses assigned to every country in the world. These are made available in the form of individual text files per country. And in the case of the SitePoint page, you can download a pre-scripted config file for many versions of web servers and firewalls. Hammer Time!

In the case of the country our client wants to block there are over 130 IP entries. These are in the form of CIDR IP ranges. This is the good news. The harder part here is that means there would have to be 130 plus entries manually added into IIS or a firewall. And this is for a smaller country. Larger countries, including countries that are known for hacking, have many thousands of CIDR IP ranges. But at least I can

download the scripts for Apache and IIS from the SitePoint page and paste them into the respective config files.

What are the downsides to this method? First off I do not know if there would be any performance hit to IIS or Apache if we were to start entering thousands of IP restrictions. I do know that AWS restricts Network ACL's to an absolute max of 40 rules in their VPC's due to "performance issues" if more were added. We're still whacking at moles. IP assignments for countries can change thus you would need to update your static list of IP bans in your web server.

This is an example of how Apache 2.4 is configured.

```
<RequireAll>
Require all granted
Require not ip 5.11.40.0/21
Require not ip 5.34.160.0/21
Require not ip 5.43.192.0/19
Require not ip 5.102.96.0/19
.....
Require not ip 217.78.48.0/20
</RequireAll>
```

This is an example of how the IIS XML web.config is configured. The CIRD notation needs to be converted to IP and network mask format.

```
<?xml version="1.0"?>
<configuration>
<system.webServer>
<security>
<ipSecurity allowUnlisted="true">
<clear/>
<add ipAddress="5.11.40.0" subnetMask="255.255.248.0"/>
<add ipAddress="5.34.160.0" subnetMask="255.255.248.0"/>
<add ipAddress="5.43.192.0" subnetMask="255.255.224.0"/>
<add ipAddress="5.102.96.0" subnetMask="255.255.224.0"/>
.....
<add ipAddress="217.78.48.0" subnetMask="255.255.240.0"/>
</ipSecurity>
</security>
<modules runAllManagedModulesForAllRequests="true"/>
</system.webServer>
</configuration>
```

In conclusion each option; CloudFlare, CloudFront, and IP Banning, each have their benefits and costs. CloudFront was the easiest of the three to setup and if the downsides of the IP address masking isn't an issue then it is likely the most viable solution. The AWS CloudFront solution may be best if you are already on AWS and you have an understanding of AWS Solutions Architecting. Both CDN options have country restrictions (and rate limiting) that will help in preventing potential credit card scammers from misusing your shopping carts. IP Banning is simplistic, it has no additional dollar costs. But it may be a performance hit to your web server if you have a very large number of IP restrictions. You may also have to update the IP lists if IP assignments to a country change. It's also worth noting that all methods can be bypassed via proxies.

CF Webtools is an Amazon Web Services Partner. Our Operations Group can build,

manage, and maintain your AWS services. We also handle migration of physical servers into AWS Cloud services. If you are looking for professional AWS management our operations group is standing by 24/7 - give us a call at 402-408-3733, or send a note to operations at cfwebtools.com.