Using CDN for Entire Website and Country Blocking - Part 2

Posted At : November 29, 2018 3:22 PM | Posted By : Wil Genovese
Related Categories: ColdFusion, Coldfusion Security, Hosting and Networking, Security, AWS

This is Part 2 in a short series of articles about blocking entire countries from a website. See Part 1.

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 this article I'm writing about using Amazon Web Services CloudFront to block entire countries.

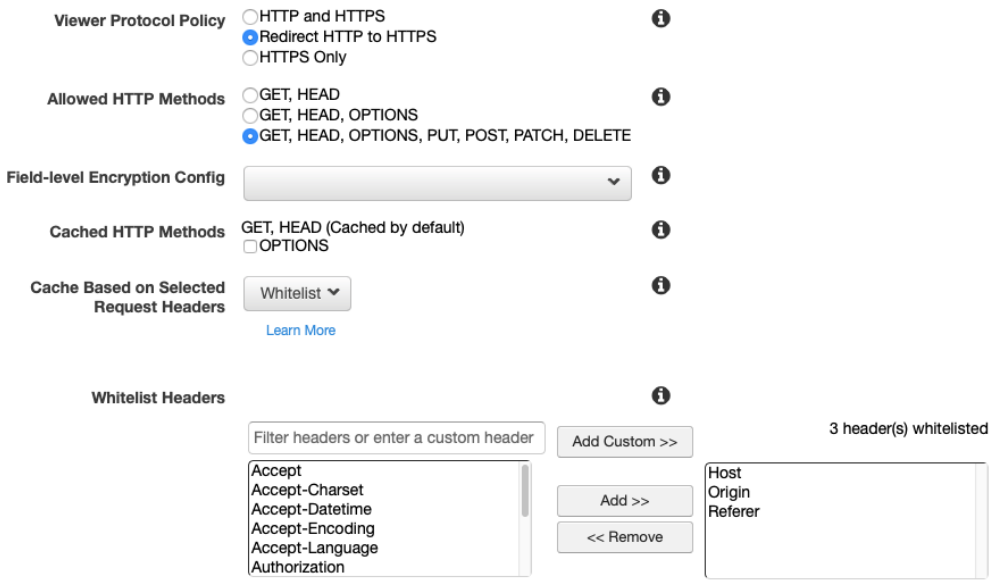
Amazon AWS CloudFront
AWS CloudFront does offer country blocking. I thought this would be an easy setup, but it isn't. When I tried to setup AWS CloudFront to 'front' an entire website I found there are many pieces that needed to be in place in order for CloudFront to handle the entire website.

Route 53 is needed or any other DNS that allows an ALIAS record for the Zone Apex record. This is because the Zone Apex record (root record) will be set to the URL provided by CloudFront and not an IP address.

Elastic Load Balancing is needed. The CloudFront origin (EC2 server) needs to be behind an TCP Elastic Load balancer. If there is only one site then the ELB target can be the instance itself. If the EC2 instance hosts multiple different sites, then we need to add multiple internal IP addresses to the instance and configure the origin site to be on it's own IP. Then the ELB should be configured to that internal IP address and not instance. If you are passing host headers in the CloudFront 'Behavior' section then you can have a single IP on the web server with multiple sites per usual for virtual name hosting. You have to setup the TCP ELB as TCP port 80 passthrough in order to pass the original IP addresses to the web server.

AWS Certificate Manager is needed to create a new free SSL for the domain name being setup in CloudFront. (I say it's needed because all sites should be using TLS protocols these days.) I found a wild card certificate works well.

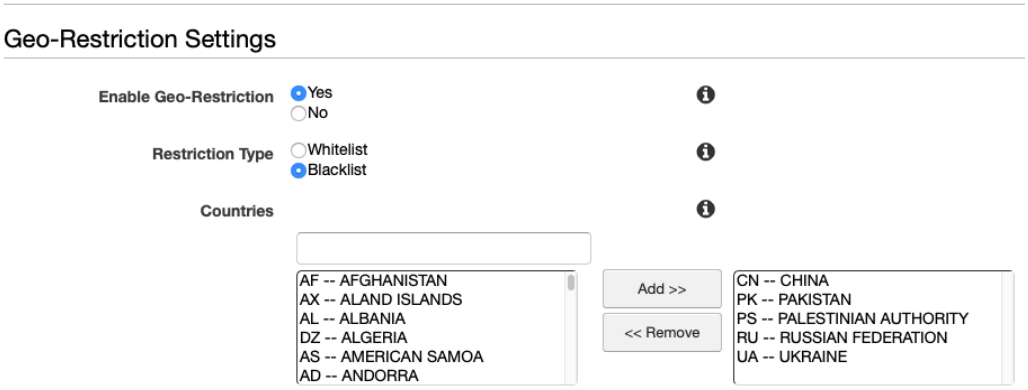
Then lastly **AWS CloudFront** itself can be setup. The settings are a bit tricky. The Origin will be the ELB which will then pass requests to the EC2 instance. If you want or need forms to be posted to the website then you need to select "GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE" option for Allowed HTTP Methods. If you need to allow logins then you have to choose "All" for Forward Cookies.



There are costs to each part. Route 53 charges by zone and number of requests. Elastic Load Balancing charges by the hour and by data transfer amounts. Then Cloud Front charges by data transfer amount.

There are downsides to this method as well. In addition to the AWS method being harder and more complex to setup there are more costs involved. I can pass the original requesting IP address through to the web server, it still comes through in the X-Forwarded-For custom header. In Apache it's easy to globally capture and place this value into log files or the CGI scope. IIS does not allow this to be done at a global level meaning each IIS site must be configured for the custom headers. Additionally, you may need to custom code the web application to read X-Forwarded-For no matter which web server you are using.

After you have all of that setup, configured, and working you can now start blocking countries. This is done in the AWS CloudFront Restrictions section. You can add a Geo-Restriction blacklist or whitelist by country.



Part 3 will cover using IIS and Apache and a slightly better hammer in the Whack-A-Mole method.

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.