



Vincent Yiu [Follow](#)

Advanced Threat Replication. Simulating real threat actors using bleeding edge techniques.

Mar 15 · 4 min read

Domain Fronting: Who Am I?

My blog has moved: <https://vincentyi.co.uk>

TLDR; Set whatever Host header you want in your Domain Fronting packet when you use CloudFront

Note: It's 1:31 am, I do my independent research to contribute to the community in my evenings. So give me a break if it all sounds like I'm talking to myself. Take what you will and enjoy.

Quick blogpost tonight, after having a play around with Alibaba in my post here. It received pretty good feedback from the community. I learned a whole lot, I'm sure the threat replication experts and incident responders have learned something new too.

Understanding the core of how HTTP routing works, and how the Content Delivery Network system is just a hack on top of a protocol that already exists in order to allow for seamless acceleration of content distribution.

Let's have a quick look at a HTTP packet:

```
GET / HTTP/1.1
Host: domain.com
```

This packet arrives at a target host, the host parses the Host header and performs virtual hosting to route to the right application and know what to show. If we take a content delivery network and place it in front, for example where CloudFront give us xxxxxxxx.cloudfront.net as a instance name, and it connects to the domain.com origin. Great, CloudFront knows how to fetch domain.com, because that's what the host header will be after you

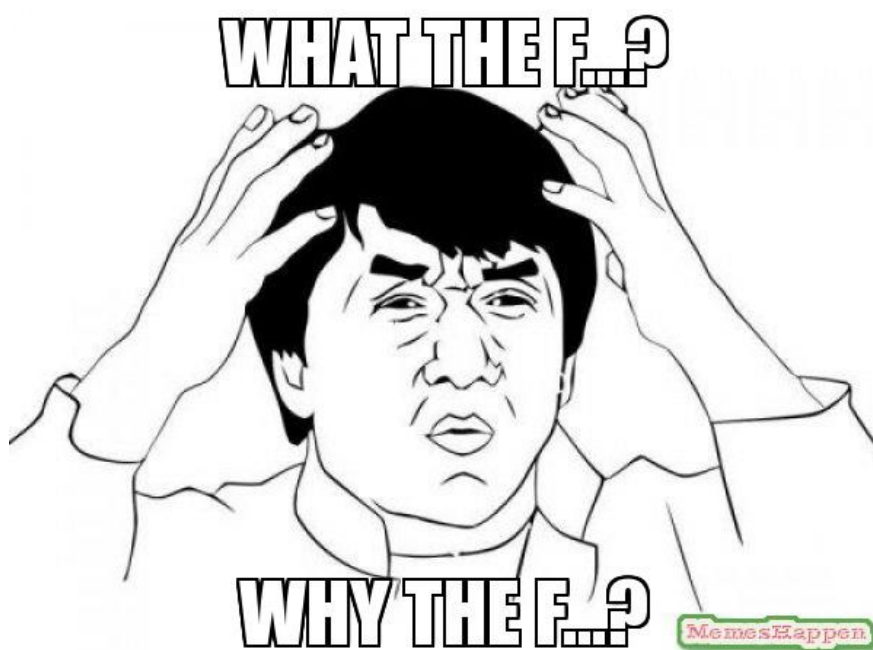
perform the DNS request to find out what server to connect to. However, how does CloudFront know what content you want? and more importantly, from what instance?

When we fetch content through the CDN, we issue the following request:

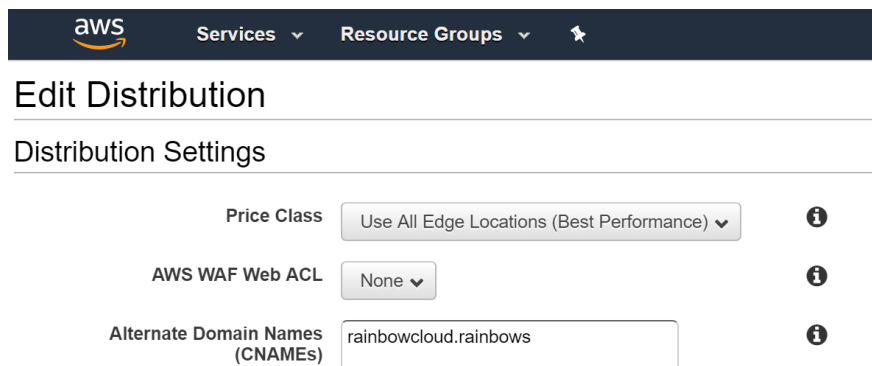
```
GET / HTTP/1.1  
Host: xxxxxxxx.cloudfront.net
```

Great, we resolve this, it gives us a CloudFront edge node belonging to Amazon. We tell it we want xxxxxxxx.cloudfront.net, so it goes ahead and finds out what CDN instance ID xxxxxxxx.cloudfront.net belongs to. Let's call this ID: 123. Now, 123 is configured to do the previous GET request to domain.com. So you might be thinking, what if my company wants to use cdn.domain.com?

Now think about it, if you set cdn.domain.com CNAME xxxxxxxx.cloudfront.net, it shouldn't work right? Your browser is looking for cdn.domain.com, but the DNS record points you at a Amazon edge node. The Amazon edge node doesn't know how to translate cdn.domain.com to ID 123 to know where to fetch the origin.



If we go back to Amazon's guidelines for setting up the CNAMEs, it actually does say that you should mention it in the set-up or in the configuration.



The screenshot shows the AWS 'Edit Distribution' page. The 'Distribution Settings' section includes three main configuration items, each with an information icon (i) to its right:

- Price Class:** Set to 'Use All Edge Locations (Best Performance)'.
- AWS WAF Web ACL:** Set to 'None'.
- Alternate Domain Names (CNAMEs):** A text input field containing 'rainbowcloud.rainbows'.

Great, so that's where we stick the CNAME. So that Amazon knows that rainbowcloud.rainbows should find ID 123 to then be able to fetch origin domain.com. Great.

What? Rainbowcloud.rainbows? :)

No CNAME domain verification

Amazon has no CNAME domain verification, you don't have to own the domain to be able to allocate the CNAME. This was already reported by many bug bounty hunters, and it's not been acknowledged as an issue. Therefore, this is definitely a FEATURE!



Time to set them host headers!

```
http-get {  
  set uri "/_utm.gif";  
  client {  
    header "Host" "rainbowcloud.rainbows";  
  
    parameter "utm_source" "UA-3207684-2";  
    parameter "utm_medium" "1";  
    parameter "utm_campaign" "150-8804-1";  
    parameter "utm_term" "1288x1004";  
    parameter "utm_content" "32-bit";  
  }  
}
```

Them domain fronting host headers

It could be a pretty cool way to put off the blue team, who are looking for cloudfront.net or appspot.com for example. It can also add an extra layer of confusion during an attack simulation. You can set your host header to a government domain, set it to a financial domain that doesn't even use CloudFront, or anything you want. This can put them off in terms of attribution, or even make them:



Take away points

Red

Change host headers, don't want them to fingerprint off cloudfront.net, appspot.com, etc. This was never the way to detect domain fronting, so break that patch and show them why :)

You have a new toy to play with, for a little while.

Blue

Monitor all host header to domain requested discrepancies.

Amazon

This isn't really an issue. It's been reported before by bug bounty hunters and not acknowledged as a cyber security issue.