High-reputation Redirectors and Domain Fronting

By Raphael Mudge

Published: 2017-02-06 · Archived: 2026-04-05 22:33:38 UTC

Working on Cobalt Strike, I get some insight into what folks are trying to do with it. Recently, the use of domain fronting for redirectors has come on my radar.

A redirector is a server that sits between your malware controller and the target network. Domain fronting is a collection of techniques to make use of other people's domains and infrastructure as redirectors for your controller.

A trivial form of domain fronting is to stand up a node in Amazon's EC2 and configure it as a redirector for your controller. The FQDN of your EC2 instance is an amazonaws.com subdomain. Your payloads may call home to this. While this is beneficial in some cases, this isn't where things get interesting.

Domain fronting becomes interesting when used to appropriate high-reputation domains as redirectors for your controller.

Domain Fronting with Alternate Hosts

How is it possible to use a high-reputation domain that you don't control? Let's use Amazon's CloudFront as an example.

CloudFront is a Content Delivery Network service. It provides its users a globally distributed cache for files hosted on their servers. This reduces load on the customer's servers and allows the CDN to serve cached content from data centers close(r) to the requester. Each CloudFront configuration is called a "distribution".

CloudFront identifies distributions by the FQDN used to request resources. Each CloudFront distribution has a unique cloudfront.net subdomain. CloudFront's users also have the option to serve CloudFront cached objects via their own sub-domain. This is done by creating a DNS record that points to CloudFront and telling CloudFront to associate that DNS record with a specific distribution. Easy enough.

When a client connects to CloudFront, the DNS name that led there is lost information. CloudFront relies on other parts of the request to extract which DNS name the client wants resources from. In an HTTP request, this is the Host header.

One way to domain front is to configure a payload to call home to one host (e.g., media.startupunicorn.com) and set the Host header to something else (e.g., mydistribution.cloudfront.net). If the Host header is set right (and nothing else changes it), your cloudfront.net configuration will dictate what happens next.

Let's examine what this looks like by hand.

The host **malwarec2.losenolove.com** is my Cobalt Strike web server, setup to serve /foo.txt. Naturally a request with wget works. (Note: I change wget's User-Agent with –U because Cobalt Strike always returns a 404 for lynx, wget, and curl useragents).

```
2  I&amp;amp;amp;amp;amp;#039;m a happy little malware controller. :)
```

d16b91n8fagr3u.cloudfront.net is my CloudFront distribution. I've configured it to serve objects from **malwarec2.losenolove.com**. A request to this host works to retrieve our file as well:

```
2  I&amp;amp;amp;amp;amp;#039;m a happy little malware controller. :)
```

What happens if we forge the Host header to a different identity? In this case, we get nothing back.

a0.awsstatic.com is a domain name that points to CloudFront. I know about this domain because other resources on domain fronting use it as an example. If I request /foo.txt from this host, naturally it's not going to give me anything.

Let's modify that slightly. We'll use the **a0.awsstatic.com** domain (it all goes to the same place, right?)—but, we'll forge the Host header to the FQDN of my CloudFront distribution. In this case, I get back the text file.

```
1  root@kali:~# wget -U demo -q -O - http://a0.awsstatic.com/foo.txt --header
   &amp;amp;amp;amp;amp;amp;amp;quot;Host:
   d16b91n8fagr3u.cloudfront.net&amp;amp;amp;amp;amp;amp;amp;quot;
2  I&amp;amp;amp;amp;amp;#039;m a happy little malware controller. :)
```

Here, I've used a0.awsstatic.com as my high-reputation domain name. There are other, far more interesting, options though.

Domain Fronting with Cobalt Strike

Tom Steele and Chris Patten from Optiv's Attack and Penetration Team wrote Escape and Evasion Egressing Restricted Networks. This blog post shows how to setup a CloudFront distribution as a redirector for Cobalt Strike's Beacon payload.

Once this is setup, you'll want to decide which domain(s) you will use as redirectors. Let's say a popular blog service uses CloudFront to serve static images. You may decide it makes sense to use this domain for your C2. Fine!

Next, I recommend you configure a Cobalt Strike Malleable C2 profile that matches something plausible on this domain. Malleable C2 is a Cobalt Strike technology that allows you, the product's user, to shape Cobalt Strike's Beacon traffic to look like other malware or something legitimate. If the domain serves static images, make a

profile that looks like one of these static images. Keep in mind it's possible to use only HTTP GETs (or even HTTP POSTs) for C2.

Once you have that part figured out, you'll want to update your profile to insert your CloudFront distribution's FQDN into the Host header. You can do this with Malleable C2's header keyword:

```
1 http-get {
2 client {
3 header &amp;amp;amp;amp;amp;amp;amp;quot;Host&amp;amp;amp;amp;amp;amp;amp;quot;
&amp;amp;amp;amp;amp;amp;amp;quot;[your
distribution].cloudfront.net&amp;amp;amp;amp;amp;amp;amp;quot;;
```

Make sure you do this in both the **http-get** -> **client** and **http-post** -> **client** Malleable C2 blocks. During a conversation, a friend remarked that they were stuck for awhile because they set the Host header in the http-get block, but not the http-post block. You have to set this in both places.

```
1 http-post {
2 client {
3 header &amp;amp;amp;amp;amp;amp;amp;quot;Host&amp;amp;amp;amp;amp;amp;amp;quot;
&amp;amp;amp;amp;amp;amp;amp;quot;[your
distribution].cloudfront.net&amp;amp;amp;amp;amp;amp;amp;quot;;
```

Once this is setup, you can configure your Beacon payload to call home to different domains that point to CloudFront.

One limitation: there's no option to set the Host header in Cobalt Strike's stagers. You can work with stageless payloads though. Go to **Attacks** -> **Packages** -> **Windows EXE (S)** to export a Cobalt Strike stageless artifact. If you want to stage, I recommend that you configure the stager to connect to your cloudfront.net distribution directly or use another redirector.

This video walks through all of this, end-to-end:

Finding High-reputation Domains for Use

My examples here use a0.awsstatic.com as an alternate host. Think of it as the Hello World of Domain Fronting. Vincent Yiu from MDSec took this a step further. He wrote a script to check likely CDN subdomains from a list of popular websites. His initial work found over three thousand subdomains that point to CloudFront and demonstrated that they work as alternate hosts with the technique discussed here.

A Note About RFC 2616, Section 14.23

So far, this blog post focuses on domain fronting over HTTP. **If the target system goes through a proxy server, you're in trouble.** An RFC-compliant HTTP proxy server will rewrite the Host header in an HTTP request to match the domain in the URL it's asked to retrieve. The Squid proxy documentation talks about this behavior. For ~~many~~ some networks, this means HTTP is a non-option.

Update 7 Feb 2017: This behavior matches my experiments with a Squid proxy locally, but don't take it for granted that your target's appliance(s) work this way. After I made this post live, Vincent Yiu took a look at a commercial secure web appliance and its behavior with these techniques. This appliance didn't rewrite the Host header as expected. If you're curious about how an appliance that enforces site categorization behaves with these techniques, Vincent's latest video is worth a look:

Ett fel inträffade.

Det går inte att köra JavaScript.

You probably want SSL/TLS

What about SSL/TLS? That's an option. This will likely get you through some proxy configurations. A device that intercepts SSL traffic ~~will~~ may make life more difficult. You may find that certain networks will exempt some high-reputation domains from SSL interception.


df_https_withproxy

If you'd like to use the HTTPS Beacon with CloudFront:

1. Be aware, CloudFront requires your web server to have a valid SSL certificate.
2. Consult Cobalt Strike's Malleable C2 documentation. It shows how to use a valid SSL certificate with Beacon.

Other Services

Here, I've given a lot of details on domain fronting with CloudFront. There are other fronting-capable web services where these (and other) techniques apply. The Camouflage at encryption layer: domain fronting blog post demonstrates these concepts with Google App Engine. The documentation for the Meek Pluggable Tor Transport also discusses several domain fronting options. Finally, Blocking-resistant communication through domain fronting describes this concept in detail, for multiple services.

My Thoughts

Domain Fronting is an interesting technique to use high-reputation domains for callbacks. It's not the right tool for all situations though. An RFC compliant proxy will defeat HTTP requests. A proxy server that terminates and inspects SSL/TLS sessions ~~will~~ might handily defeat this as well. There's probably wiggle room using this technique with whitelisted high-reputation domains. This makes locating domain options even more important! Remember, these are not CloudFront-only techniques.

Source: https://www.cobaltstrike.com/blog/high-reputation-redirectors-and-domain-fronting/