Domain Fronting

Subjects: Others

Contributor: HandWiki Zheng

Domain fronting is a technique for Internet censorship circumvention that uses different domain names in different communication layers of an HTTPS connection to discreetly connect to a different target domain than is discernable to third parties monitoring the requests and connections. Due to quirks in security certificates, the redirect systems of the content delivery networks (CDNs) used as 'domain fronts', and the protection provided by HTTPS, censors are typically unable to differentiate circumvention ("domain-fronted") traffic from overt non-fronted traffic for any given domain name. As such they are forced to either allow all traffic to the domain front—including circumvention traffic—or block the domain front entirely, which may result in expensive collateral damage and has been likened to "blocking the rest of the Internet".[note 1] Domain fronting does not conform to HTTP standards that require the SNI extension and HTTP Host header to contain the same domain. Large cloud service providers, including Amazon and Google, now actively prohibit domain fronting, which has made it "largely non-viable"[note 1] as a censorship bypass technique.

Keywords: cloud service ; content delivery ; domain names

1. Technical Details

1.1. Basis

The basis for domain fronting is using different domain names at different layers of communication with the servers (that supports multiple target domains; i.e. Subject Alternative Names) of a large hosting providers or a content delivery network (CDN). CDNs are used due to idiosyncrasies in how they route traffic and requests, which is what allows fronting to work.[1][2]

1.2. Obfuscating Requests

In an HTTPS request, the destination domain name appears in three relevant places: the DNS query, the TLS Server Name Indication (SNI) extension, and the HTTPS Host header. Ordinarily the same domain name is listed in all three places.[3]:1

In a domain-fronted HTTPS request, one domain appears on the “outside” of an HTTPS request in plain text—in the DNS request and SNI extension—which will be what the client wants to pretend they are targeting in the connection establishment and is the one that is visible to censors, while a covert domain appears on the “inside”—in the HTTPS Host header, invisible to the censor under HTTPS encryption—which would be the actual target of the connection.[1][3]:2

```
# wget sends a dns query and connects to www.google.com but the http host header
requests # the www.youtube.com webpage, which it is able to fetch and display. Here
www.youtube.com # is essentially domain-fronted by www.google.com; that is, by
blocking www.youtube.com # but allowing www.google.com, a censor may be trivially
bypassed using a domain-fronted request wget -q -O - https://www.google.com/ --header
'Host: www.youtube.com' | grep -o '<title>.*</title>' <title>YouTube</title>
```

Due to encryption of the HTTPS hosts header by the HTTPS protocol, circumvention traffic is indistinguishable from 'legitimate' (non-fronted) traffic. Implementations of domain fronting supplement HTTPS with using large content delivery networks (such as various large CDNs) as their front domains,[3] which are relied on by large parts of the web for functionality.[4] To block the circumvention traffic, a censor will have to outright block the front domain.[3] Blocking popular content delivery networks is economically, politically, and diplomatically infeasible for most censors.[1][4]

When Telegram was blocked in April 2018 following a court ruling in Russia through ISP-blocking of the CDNs Telegram used as a front to evade blocks on its own IP addresses, 15.8 million IP addresses associated with Google and Amazon's CDN were blocked collaterally. This resulted in a large scale network outages for major banks, retail chains, and numerous websites; the manner of blocking was criticised for incompetence.[5]

1.3. Leveraging Request Forwarding

Domain fronting works with CDNs as—when served with two different domains in one request—they are (or historically speaking—they were; see *§Disabling*) configured to automatically fulfil a request to view/access the domain specified in the Hosts header even after finding the SNI extension to have a different domain. This behaviour was and is not universal across hosting providers; there are services that validate if the same domain is used in the different layers of an HTTP request. A variation of the usual domain fronting technique, known as *domainless* fronting may work in this case, which leaves the SNI field blank.[6]

If the request to access the Hosts header domain succeeds, to the censor or third parties monitoring connections, it appears that the CDN has internally forwarded the request to an uninteresting page within its network; this is the final connection they typically monitor. In circumvention scenarios, the domain in the Hosts header will be a proxy. The Hosts header domain, being a proxy, would be blocked by the censor if accessed directly; fronting hides its address from the censor and allows parties to evade blocks and access it. No traffic ever reaches the front domain specified in the DNS request and SNI extension; the CDN's frontend server is the only third-party in this interaction that can decrypt the Hosts header and know the true destination of the covert request. It is possible to emulate this same behaviour with host services that don't automatically forward requests, through a "reflector" web application.[3]:2

As a general rule, web services only forward requests to their own customers' domains, not arbitrary ones. It is necessary then for the blocked domains, that use domain fronting, to also be hosted by the same large provider as the innocuous sites they will be using as a front in their HTTPS requests (for DNS and STI).[3]:2

2. Usage

2.1. Internet Censorship Circumvention

Signal

Signal, the secure messaging service, deployed domain fronting in builds of their apps from 2016 to 2018 to bypass blocks of direct connections to their servers from Egypt, Oman, Qatar and the United Arab Emirates.[4][7]

Tor Browser

The Tor anonymity network uses an implementation of domain fronting called 'meek' in its official web browser to bypass blocks to the Tor network.[1][2][4]

Telegram

Telegram used Amazon Web Services as a domain front to resist attempts to block the service in Russia.[8]

GreatFire

GreatFire, a non-profit that assists users in circumventing the Great Firewall, used domain fronting at one point.[4]

2.2. Cyberattacks

Domain fronting has been used by private, and state-sponsored individuals and groups to cover their tracks and discreetly launch cyberattacks and disseminate malware.[1][4]

Cozy Bear

The Russian hacker group Cozy Bear, classed as *APT29*, has been observed to have used domain fronting to discreetly gain unauthorised access to systems by pretending to be legitimate traffic from CDNs. Their technique used the meek plugin—developed by the Tor Project for its anonymity network—to feign detection.[9][10]

3. Disabling

The endurance of domain fronting as a method for censorship circumvention has been likened to the expensive collateral damage that blocking it comes with—as to block domain fronting, one must block all traffic to and from their fronts (CDNs and large providers) which by design are often relied on by countless other web services.[4] The Signal Foundation drew the analogy that to block one domain fronted site you "have to block the rest of the Internet as well."[11]

Cloudflare disabled domain fronting in 2015.[12] In April 2018, Google and Amazon both disabled domain fronting from their content delivery services by removing the idiosyncrasies in redirect schemes that allowed fronting to happen.[13] Google broke domain fronting by removing the ability to use 'google.com' as a front domain by changing how their CDN was structured.[14] When requested to comment they said domain fronting had "never been a supported feature" and that the changes made were long-planned upgrades.[14][15][16] Amazon claimed fronting was "already handled as a breach of AWS Terms of Service" and implemented a set of changes that prohibited the obfuscation that allowed sites to masquerade as and use CloudFront domains of other websites as fronts.[11][17][18]

3.1. Reactions

Various publications speculated that the effort by both Google and Amazon was in part due to pressure from the Russian government and its communications authority Roskomnadzor blocking millions of Google and Amazon domains, in April 2018 as well, due to Telegram using them as fronts.[14][19][20][21]

Digital rights advocates have commented that the move undermines peoples ability to access and transmit information freely and securely in repressive states.[22]

According to Signal's founder, Moxie Marlinspike, Google management came to question whether they wanted to act as a front for sites and services entire nation states wanted to block as domain fronting gained popular attention with apps like Signal implementing it. He called fronting a circumvention tool "now largely non-viable" in the countries it was needed.[11]

References

1. "Privacy 2019: Tor, Meek & The Rise And Fall Of Domain Fronting". 2019-04-15. https://www.sentinelone.com/blog/privacy-2019-tor-meek-rise-fall-domain-fronting/.
2. "doc/meek – Tor Bug Tracker & Wiki". https://trac.torproject.org/projects/tor/wiki/doc/meek.
3. Fifield, David; Lan, Chang; Hynes, Rod; Wegmann, Percy; Paxson, Vern (15 February 2015). "Blocking-resistant communication through domain fronting". Proceedings on Privacy Enhancing Technologies 2015 (2): 46–64. doi:10.1515/popets-2015-0009. ISSN 2299-0984. http://www.icir.org/vern/papers/meek-PETS-2015.pdf. Retrieved 2017-01-03.
4. "The Death of Domain Fronting | What Lies Ahead?" (in en-US). 2018-06-11. https://blog.finjan.com/what-is-domain-fronting/.
5. Savov, Vlad (2018-04-17). "Russia's Telegram ban is a big, convoluted mess" (in en). https://www.theverge.com/2018/4/17/17246150/telegram-russia-ban.
6. "Proxy: Domain Fronting, Sub-technique T1090.004 - Enterprise | MITRE ATT&CK®". https://attack.mitre.org/techniques/T1090/004/.
7. > Blog >> Doodles, stickers, and censorship circumvention for Signal Android". https://whispersystems.org/blog/doodles-stickers-censorship/. " id="ref_7">"Open Whisper Systems >> Blog >> Doodles, stickers, and censorship circumvention for Signal Android". https://whispersystems.org/blog/doodles-stickers-censorship/.
8. Brandom, Russell (2018-04-30). "Amazon Web Services starts blocking domain-fronting, following Google's lead" (in en). https://www.theverge.com/2018/4/30/17304782/amazon-domain-fronting-google-discontinued.
9. "APT29 Domain Fronting With TOR" (in en). https://www.fireeye.com/blog/threat-research/2017/03/apt29_domain_frontin.html.
10. "Domain Fronting, Phishing Attacks, and What CISOs Need to Know" (in en-US). 2018-12-13. https://cofense.com/domain-fronting-phishing-attacks-cisos-need-know/.
11. Marlinspike, Moxie (2018-05-01). "A letter from Amazon". https://signal.org/blog/looking-back-on-the-front/.
12. "#14256 (Clarify whether Cloudflare's Universal SSL thing works with meek) – Tor Bug Tracker & Wiki". https://trac.torproject.org/projects/tor/ticket/14256#comment:2.
13. "Domain fronting: pros and cons | NordVPN" (in en). 2019-07-12. https://nordvpn.com/blog/domain-fronting/.
14. Gallagher, Sean (2018-05-02). "Amazon blocks domain fronting, threatens to shut down Signal's account" (in en-us). https://arstechnica.com/information-technology/2018/05/amazon-blocks-domain-fronting-threatens-to-shut-down-signals-account/.

15. Brandom, Russell. "A Google update just created a big problem for anti-censorship tools" (in en-US). https://www.theverge.com/2018/4/18/17253784/google-domain-fronting-discontinued-signal-tor-vpn.
16. "Google ends "domain fronting," a crucial way for tools to evade censors - Access Now". 18 April 2018. https://www.accessnow.org/google-ends-domain-fronting-a-crucial-way-for-tools-to-evade-censors/.
17. "Enhanced Domain Protections for Amazon CloudFront Requests". 2018-04-27. https://aws.amazon.com/blogs/security/enhanced-domain-protections-for-amazon-cloudfront-requests/.
18. "Amazon Web Services starts blocking domain-fronting, following Google's lead". 2018-04-30. https://www.theverge.com/2018/4/30/17304782/amazon-domain-fronting-google-discontinued.
19. "Amazon and Google bow to Russian censors in Telegram battle" (in en-US). Fast Company. 2018-05-04. https://www.fastcompany.com/40568177/amazon-and-google-bow-to-russian-censors-in-telegram-battle.
20. Bershidsky, Leonid (May 3, 2018). "Russian Censor Gets Help From Amazon and Google". https://www.bloomberg.com/view/articles/2018-05-03/telegram-block-gets-help-from-google-and-amazon.
21. "Info". Tass.ru. http://tass.ru/pmef-2018/articles/5231399.
22. Dahir, Abdi Latif. "Google and Amazon's move to block domain fronting will hurt activists under repressive regimes" (in en). https://qz.com/africa/1268974/google-and-amazon-end-domain-fronting-to-bypass-internet-censorship-in-africa/.