Domain Fronting:

Making Backdoor Access Look Like Google Requests

Alex Johnson
Comp 116
Tufts University
May 5th, 2018

1. Abstract

Domain fronting first gained widespread attention in 2017, when it came out that Cozy Bear (ATP 29) had used the technique to send packets out of a network, evading firewalls. The practice was initially developed to circumvent censorship, and has been deployed for that purpose, most notably by Signal, an encrypted messaging service. It reappeared on front page technology news in April 2018 when Google and Amazon banned domain fronting on their pages. This paper documents this technique in detail, analyzes its use cases, and finally discusses some of these current controversies.

2. Introduction

The Russian hacker group Cozy Bear is well known for having broken into the systems of many nation states across the Western Hemisphere, as well as the Democratic National Committee (DNC) in 2016. They often use phishing techniques to gain access to a system to steal classified information.[1] In one attack that was investigated by FireEye, their command and control malware sent this classified data out over the network using domain fronting, the topic of this paper. The malware acted particularly stealthily, exhibiting what appeared to be fairly normal behavior on disk as well as when sending packets out over a network, as domain fronting obfuscated the outbound information.[2] This technique was first developed by researchers in 2014 to help people affected by censorship access blocked resources and be able to communicate freely.[3] Domain fronting allowed the network traffic of Cozy Bear's malware to look like it was a regular employee visiting Google, when it was really sending confidential information out over the TOR network.

3. To The Community

Fifield et al. showed that domain fronting is an effective way to circumvent censorship, so services like Signal have adopted domain fronting to reach customers in areas with repressive internet regulations. Domain fronting could be used more widely in products trying to serve people in such countries, especially products like Signal, that target journalists and others who require intense secrecy and strong encryption. However, in April 2018, Google and Amazon both took steps to block domain fronting on their services. Google called it "not a supported feature at

[1] Eric Lipton, David E. Sanger, Scott Shane, "The Perfect Weapon: How Russian Cyberpower Invaded the U.S.," *The New York Times* (New York, NY), Dec. 13, 2016.
[2] https://www.youtube.com/watch?v=Ldzr0bfGtHc&t=2822s
[3] Matthew Dunwoody, "APT29 Domain Fronting With TOR," FireEye, March 27, 2017, https://www.fireeye.com/blog/threat-research/2017/03/apt29_domain_frontin.html.

Google", but a "quirk".[4] Amazon also justified the move by referencing its use in malware.[5] Still, internet advocacy groups like Access Now urged Google to reevaluate its decision, since it harms the freedoms of many users living in countries with oppressive internet policies.[6] Domain fronting is an active topic of debate: It might infringe on cloud providers' user agreements, but it also helps provide access to crucial web services for users living under oppressive internet regulations. This paper provides the background and discussion that is crucial to develop an informed opinion on this double-edged sword.

4. Technical Background

In order to understand how domain fronting works, we first have to understand the underlying structure of web services that are exploited by this technique. So this paper will begin by explaining some of the basics behind how the web works, including its protocols, CDN architecture of webservers, and the TOR network.

a. HTTP and HTTPS

HTTP is the basic protocol used to access resources on the internet. When a client wishes to access a certain resource contained on a server, the client sends a "request" to the server. The server will perform some internal computations and send a "response" back to the client. This request-response model is the Hyper Text Transfer Protocol (HTTP), and represents the most basic means of web interaction. Any HTTP request contains a message body and many headers that describe the communication, such as encoding or credentials. Importantly for domain fronting, these headers include a "Host" header, which specifies the final host that should receive the request. All communication in the HTTP model is sent in plaintext, meaning that anyone who intercepts this communication has unrestricted access to this interaction. [7]

The TLS protocol was created to mitigate this obvious security flaw. This protocol is completely encrypted, so even if someone intercepts TLS communication, they will not be able to view any of it beyond the destination address. TLS begins with the client and server establishing a secure connection through what is known as the "Diffie-Hellman" exchange. This exchange allows the two computers to create an encryption key that nobody else can access or

---

[4] Russell Brandom, "A Google update just created a big problem for anti-censorship tools," *The Verge*, April 18, 2018, https://www.theverge.com/2018/4/18/17253784/google-domain-fronting-discontinued-signal-tor-vpn.

[5] Russell Brandom, "Amazon Web Services starts blocking domain-fronting, following Google's lead," *The Verge*, April 30, 2018, https://www.theverge.com/2018/4/30/17304782/amazon-domain-fronting-google-discontinued.

[6] "Google ends 'domain fronting,' a crucial way for tools to evade censors," Access Now, April 18, 2018, https://www.accessnow.org/google-ends-domain-fronting-a-crucial-way-for-tools-to-evade-censors/.

[7] Network Working Group, "Request for Comments: 2616", June 1999, https://www.ietf.org/rfc/rfc2616.txt.

recreate, even if they intercept this communication. HTTPS is regular HTTP that runs in a TLS session and is therefore secure and unreadable for sniffers.[8]

b. CDN Servers

Real-life servers for large web services, such as google.com, do not run on just one server, since no single server can handle that many requests. Instead, behind any such entry point there lies a vast network of servers that share the combined load of all requests. Such networks are referred to as Content Delivery Networks (CDN). Any request to such a service will enter the network and then be rerouted to an appropriate and available server within the CDN, which responds to the request. One common way for CDNs to reroute their traffic in this way is to edit the "Host" header in the HTTP request. As mentioned above, this header describes the endpoint which should receive the request. Usually the host is simply the hostname of the URL the request is being sent to, for example "Host: www.google.com". But web services can use this header to their advantage for their CDN architecture: The host does not need to be the same as the destination hostname in the URL. Instead, the host header can be the name of another server in the same CDN network which receives the request, and the request is internally forwarded to that server. This technique, referred to as "origin pull", is commonly used for CDN services.[9]

c. The Onion Router

Finally, to understand domain fronting, we need to understand The Onion Router (TOR). TOR allows users to stay anonymous on the web, even in the face of surveillance. Where internet traffic usually travels a predictable and straightforward route, TOR traffic is encrypted and routed through an unpredictable path of TOR nodes. If two computers establish a connection through one route, that route will only stay valid for approximately 10 minutes, after which a new random circuit will be created to route this traffic.[10] This makes TOR traffic practically impossible to trace. TOR is heavily funded by the US government and has been hailed for its contribution to protect journalists from "tech-savvy tyrants", but as this paper will show, the intense privacy it provides can be used for both good and bad.[11]

5. Domain Fronting in Detail

Now to get to the technical specifics of domain fronting. This section describes domain fronting as APT 29 used it. Firstly, to be used as in the APT 29 example, the attacker must initially gain access to the system by means of malware. Once that has occurred, domain fronting is a highly stealthy technique the attacker can use to send out information without getting noticed

---

[8] Network Working Group, "Request for Comments: 5246," August 2008, https://tools.ietf.org/html/rfc5246.
[9] David Fifield et al., "Blocking-resistant communication through domain fronting," *Proceedings on Privacy Enhancing Technologies* 2 (2015): 1.
[10] TOR, "TOR: Overview," TOR Project, accessed May 1, 2018, https://www.torproject.org/about/overview.html.en.
[11] "100 Top Global Thinkers of 2012," *Foreign Policy* 197 (2012).

by any firewall. It was first documented by researchers in a 2015 paper and ATP 29 abused the server those researchers created. Domain fronting exploits the CDN architecture that was discussed above: A request is sent to an inconspicuous URL, such as www.google.com, but its host header is set to another address, say meek-reflect.appspot.com, the researchers' server. Since all HTTPS communication is encrypted, the value of the host header cannot be seen by any firewall or censor listening in on these requests.[12] This request contains a nested request including the actual desired destination, beyond the reflection server. When www.google.com receives the request, this request is unencrypted and the server recognizes that the host header specifies another URL. Since appspot domains are in the Google cloud, the initial server will simply redirect the traffic to go to that URL specified in the host header. At this point the reflection server meek-reflect.appspot.com receives the request and responds to that request however it is configured to respond. In the case of the meek-reflect.appspot.com reflection server, it forwarded the nested request on to a destination specified in the request through the TOR network. This means the request becomes obfuscated and the endpoint of the request cannot be deciphered.[13]

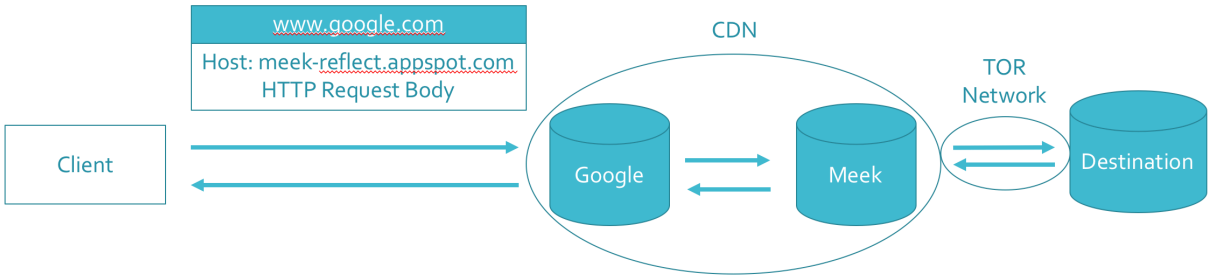

Figure 1: Request forwarding through a CDN and TOR network

Since many CDNs allows anyone to buy space on their cloud, anyone can host such a service and communicate with it through traffic that looks like it is going to an unrelated service.[14] Although meek as been disabled, an attacker could create their own reflection server, or find other servers that perform the same function. This could be done with any popular CDN cloud provider that does not block domain fronting.[15]

6. Censorship Circumvention

This paper focused on the potential for domain fronting in malware, but the researchers who first documented the technique had less nefarious intentions: censorship circumvention. The

[12] Matthew Dunwoody, "APT29 Domain Fronting With TOR," FireEye, March 27, 2017, https://www.fireeye.com/blog/threat-research/2017/03/apt29_domain_frontin.html.
[13] Matthew Dunwoody, "APT29 Domain Fronting With TOR," FireEye, March 27, 2017, https://www.fireeye.com/blog/threat-research/2017/03/apt29_domain_frontin.html.
[14] Matthew Dunwoody, "No Easy Data Breach," (presentation, DerbyCon, Louisville, KY, September 23-25, 2016), https://www.youtube.com/watch?v=Ldzr0bfGtHc&t=2822s.
[15] David Fifield et al., "Blocking-resistant communication through domain fronting," *Proceedings on Privacy Enhancing Technologies* 2 (2015): 1.

process of domain fronting remains the same, but some initial assumptions change: Instead of a systems administrator, the firewall is a state actor trying to restrict access to the full internet, and instead of malware exposing sensitive information, the user is a regular person trying to access restricted sites. The security derived from domain fronting is very effective, as section 7 will show, but censorship circumvention through domain fronting has a reliability issue. As Fifield et al. explain: "Domain fronting derives its strength from the collateral damage that results from blocking the front domain."[16] But if the state actor accepts this cost, domain fronting can be blocked. As Google has been blocked in China, reflection services hosted on the Google App Engine have always been blocked as well. Additionally, hosting just one reflection server might risk the whole CDN being blocked. If a CDN provider is unwilling to spread that risk over its entire cloud infrastructure, they might block domain fronting internally. Still, some services using domain fronting have already been deployed, such as the Signal messaging app. Signal's servers have been blocked in some countries due to its commitment to secrecy and encryption. So it uses domain fronting in Egypt and the United Arab Emirates to get around that censorship. As Signal put it in a blog post, with domain fronting "disabling Signal starts to look like disabling the internet."[17]

7. Defenses and Detection

The use case outlined in section 5 is explicitly malicious: Although researchers initially intended Domain Fronting to help people living under censorship to evade that firewall, Cozy Bear could use this same technique to keep malware undetected. So naturally, cyber professionals will be interested in how such behavior can be detected and prevented.

As explained above, when a client sends a nested request to the reflection server, only the address of the "front" will be displayed to a firewall or censor, in the case above www.google.com. The address of the reflection server is stored in the host header and the real destination of the request in the request body, both hidden by TLS encryption. So if a censor wanted to block traffic to that reflection server, they would have to block all entry points to the CDN, including www.google.com. This would result in a cost very few administrators would be willing to bare; One notable exception is China.[18] Draconian blacklisting is therefore infeasible for the standard administrator.[19]

Another option would be TLS Interception. TLS is the fundamental technology that HTTPS runs on, and it is indeed possible for administrators to intercept TLS traffic. TLS usually protects

---

[16] David Fifield et al., "Blocking-resistant communication through domain fronting," *Proceedings on Privacy Enhancing Technologies* 2 (2015): 1.
[17] Matthew Rosenfield, "Doodles, stickers, and censorship circumvention for Signal Android," Signal Blog, December 21, 2016, https://signal.org/blog/doodles-stickers-censorship/.
[18] David Fifield et al., "Blocking-resistant communication through domain fronting," *Proceedings on Privacy Enhancing Technologies* 2 (2015): 1.
[19] Matthew Dunwoody, "APT29 Domain Fronting With TOR," FireEye, March 27, 2017, https://www.fireeye.com/blog/threat-research/2017/03/apt29_domain_frontin.html.

against such a man-in-the-middle attack by exchanging certificates with the destination server, but administrators can have their organization's machines trust their own certificate. Thus, any internet connection from these machines goes through a web-proxy, which can unencrypt and inspect the data, before it sends it off to the actual destination.[20] In this case, the proxy could detect that the real destination of the nested packet and recognize it as illegitimate traffic. However, introducing such a proxy has been found to lead to its own security flaws. de Carné de Carnavalet and Mannan found that all TLS proxies they analyzed had weaknesses in certificate management or security issues in the proxy-server connection. Since these supposed security products introduce more problems than they solve, they advise against the use of proxies for antivirus purposes.[21] Finally, FireEye, who detected domain fronting in Cozy Bear's attack, recommends not looking for these domain filtering techniques, but instead finding the underlying malware on the infected machines or systems. This is also how FireEye discovered Cozy Bear's malware.[22]

8. Conclusion

Domain fronting can circumvent firewalls and censors extremely effectively by exploiting CDN architectures. Malware can send confidential information out of networks while not appearing any more dangerous than a Google search. The only effective way to block unwanted internet traffic would be to block all large cloud providers, including Google, Amazon, and more. The important takeaway from this for system administrators is that just because a system's outgoing network traffic looks clean, does not mean it actually is clean. Similarly, users in countries with repressive internet regulations can use domain fronting to evade state-imposed firewalls. However, the decisions by Google and Amazon to ban domain fronting in April 2018 shows its reliability can be jeopardized by nation states as well as large cloud providers. Other cloud providers can still be used for the same purposes, so it is unlikely that the technique becomes infeasible. But recent developments banning domain fronting show services like Signal might be at risk in some countries. This risk comes not from repressive regimes taking effective action against them, but from large cloud providers like Google and Amazon that are disallowing the fundamental technique they rely on.

---

[20] Zakir Durumeric et al., "The Security Impact of HTTPS Interception," accessed May 1, 2018, http://mdbailey.ece.illinois.edu/publications/ndss17_interception.pdf.
[21] de Carné de Carnavalet et al., "Killed by Proxy: Analyzing Client-end TLS Interception Software" accessed May 1, 2018, https://users.encs.concordia.ca/~mmannan/publications/ssl-interception-ndss2016.pdf.
[22] Matthew Dunwoody, "APT29 Domain Fronting With TOR," FireEye, March 27, 2017, https://www.fireeye.com/blog/threat-research/2017/03/apt29_domain_frontin.html.

References

Brandom, Russell. "A Google update just created a big problem for anti-censorship tools." *The Verge*. April 18, 2018. https://www.theverge.com/2018/4/18/17253784/google-domain-fronting-discontinued-signal-tor-vpn.

Brandom, Russell. "Amazon Web Services starts blocking domain-fronting, following Google's lead." *The Verge*. April 30, 2018. https://www.theverge.com/2018/4/30/17304782/amazon-domain-fronting-google-discontinued.

de Carné de Carnavalet et al., "Killed by Proxy: Analyzing Client-end TLS Interception Software." Accessed May 1, 2018. https://users.encs.concordia.ca/~mmannan/publications/ssl-interception-ndss2016.pdf.

Dunwoody, Matthew. "APT29 Domain Fronting With TOR." FireEye. March 27, 2017. https://www.fireeye.com/blog/threat-research/2017/03/apt29_domain_frontin.html.

Dunwoody, Matthew. "No Easy Data Breach." Presentation at DerbyCon, Louisville, KY, September 23-25, 2016. https://www.youtube.com/watch?v=Ldzr0bfGtHc&t=2822s.

Durumeric, Zakir, et al. "The Security Impact of HTTPS Interception." Accessed May 1, 2018. http://mdbailey.ece.illinois.edu/publications/ndss17_interception.pdf.

Fifield, David, et al. "Blocking-resistant communication through domain fronting." *Proceedings on Privacy Enhancing Technologies* 2 (2015): 1.

"Google ends 'domain fronting,' a crucial way for tools to evade censors." Access Now. April 18, 2018. https://www.accessnow.org/google-ends-domain-fronting-a-crucial-way-for-tools-to-evade-censors/.

Lipton, Eric, Sanger, David E., Shane, Scott. "The Perfect Weapon: How Russian Cyberpower Invaded the U.S." *The New York Times* (New York, NY). Dec. 13, 2016.

Network Working Group, "Request for Comments: 2616", June 1999, https://www.ietf.org/rfc/rfc2616.

Network Working Group, "Request for Comments: 5246", August 2008, https://tools.ietf.org/html/rfc5246.

Rosenfield, Matthew. "Doodles, stickers, and censorship circumvention for Signal Android." Signal Blog. December 21, 2016. https://signal.org/blog/doodles-stickers-censorship/.

TOR. "TOR: Overview." TOR Project. Accessed May 1, 2018. https://www.torproject.org/about/overview.html.en.

"100 Top Global Thinkers of 2012." *Foreign Policy* 197 (2012).