May 2022

Content Delivery Networks (CDNs) Overview and Illicit Material Considerations

Author

Wes Hardaker
Information Sciences Institute
hardaker@isi.edu

Acknowledgements

This report was prepared in coordination with Thorn as part of a strategic partnership with USC Dornsife Public Exchange and USC Viterbi Information Sciences Institute to study internet infrastructure and the flow of online Child Sexual Abuse Materials. We are grateful for the expertise and research provided by Wes Hardaker of USC Viterbi's Information Sciences Institute as well as the collaboration of Thorn's staff and leadership, particularly Melissa Stroebel and Tim O'Gorman. This project was supported by the USC Dornsife Public Exchange Project Fund. Lastly, thank you to Doug Messer and Rachel Wald from the Public Exchange team for supporting the project and the production of this report.

Disclaimer

The views expressed herein are those of the author and not necessarily those of Thorn, the USC Dornsife College of Letters, Arts and Sciences, or the University of Southern California as a whole.

For more information, contact: publicexchange@usc.edu

Introduction

The purpose of this report is to highlight the techniques and methodologies that companies, organizations and individuals use with *Content Delivery Networks (CDNs)* to speed up access to deployed websites and how illicit content publishers might also make use of CDN technologies. This document first describes caching technologies such as proxies and CDNs. With this knowledge, we then analyze points of intervention when addressing issues with illicit material, and what information should be collected to improve the chances of a content take-down request.

In alignment with Thorn's mission to end child sex trafficking and the sexual exploitation of children, the goal of this report is to assist Thorn in understanding data movement across the internet and inform the development of tools used to identify potential opportunities to stop trafficking online Child Sex Abuse Materials.

This document compliments an accompanying presentation that also highlights many of the facts in this report and augments the discussion with visual representations and figures. Please note, this report includes a glossary of terms located on the final page (p. 11).

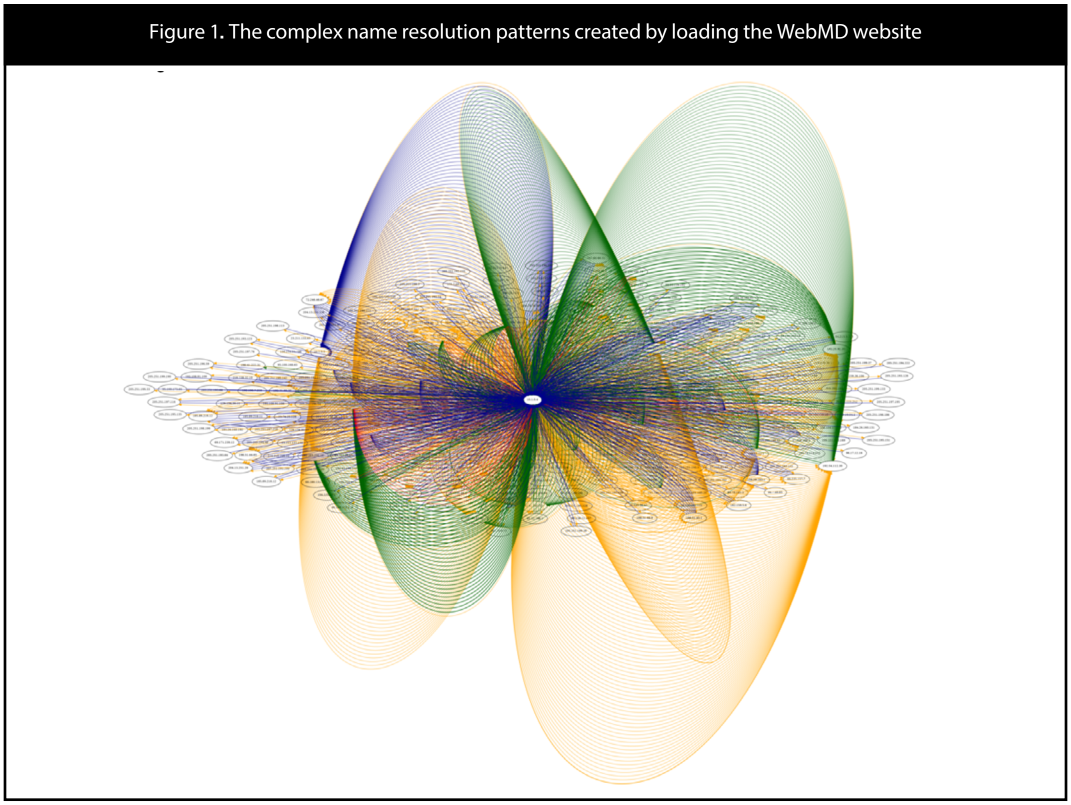
1.1. Complexity of web pages

Most novice end-users believe their web browser simply connects to the named website, gets the results and then displays it. Though not entirely inaccurate, the architecture of internet website deployment is significantly more complex. Instead of a simple, single request, web browsers must start by translating a domain name such as ***www.example.com*** into an IP address, followed by establishing a *(Secure) Hyper Text Transfer Protocol HTTP(S)* connection to that IP address and then recursively fetching all of the other data that the website needs. Other data required to properly display a website includes images, videos, Javascript website code, *cascading style sheet* information (CSS), etc. This process results in tens of Internet connections for even the simplest websites, and amounts to hundreds and thousands of connections for more complex websites.

Complex website contents are frequently hosted from multiple servers, even without CDNs in use, as the primary page's content is often pulled from one server while fonts, images, other media, and advertisements are pulled from other servers. Each of these extra servers triggers a ***Domain Name System (DNS)*** query to translate each name into IP addresses. Figure 1 below shows an example of **just the DNS requests** made when loading the WebMD web page. Adding in the HTTP(S) requests to this graph would add even more requests.

In 2007, Amazon found that a 0.5 second delay in page load times caused traffic to drop by 20%[1] . In 2017, Akamai showed 100ms delays hurt conversion rates by 7% and in 2018 Google showed similar losses of viewers based on a sliding set of delays. The result of these and other realizations have shown a direct commercial advantage to cache as much information as possible close to the end-user to speed up users' page load times.

Figure 1. The complex name resolution patterns created by loading the WebMD website



[1] https://www.gigaspaces.com/blog/amazon-found-every-100ms-of-latency-cost-them-1-in-sales

1.2. Proxies for speeding up the web

A *proxy* is a generic term for multiple internet technologies that act as a middle-agent between a client and a remote server. They are heavily used in web services to speed up client access to web resources, but also introduce additional complexities that can bring about their own problems[2].

1.2.1 ISP-centric proxies

Internet Service Providers (ISPs) deployed early versions of proxies when high speed bandwidth to cities was expensive. These early proxies would auto-cache commonly requested data for users in their geographical region. The first user to request a new image or other content would receive it slowly, after which the proxy would save the image in case other users also requested it. Today, these types of proxies are less common as the commercial industry's deployment of Content Delivery Networks (CDNs) have alleviated the need for every ISP to solve this problem themselves.

These traditional proxies are frequently in control of either the user themselves or the user's ISP. Similar to how a web-browser caches images on a user's computer, traditional proxies cache information regionally to the user.

1.2.2 Content Delivery Networks (CDNs)

CDNs are often referred to as a "reverse proxy," which is often misleading to readers. Usage and configuration of CDNs, unlike traditional proxies, are in control of the company running the CDN and their business customers – the end result being that users and even ISPs frequently do not realize that a CDN is in use. The trickier part of CDN deployment is convincing the user's web browser to connect to the nearest CDN instance when a given image, video or other resource is needed. There are a few techniques used to achieve this, which are discussed later in Section 2.

One important distinction between regular proxies and CDNs is the number of instances deployed near a given user. Because different content publishers may outsource their CDN requirements to different CDN providers, users will rotate among fetching content from multiple CDNs based on the web resources they currently require for the web page they are viewing. This can even result in multiple CDNs in use for a given page.

1.2.2 Differences between traditional proxies and CDNs

There are a few important, but subtle, differences between these two different caching strategies.

Sphere of control

The largest difference, and the most important with respect to battling illicit content, is what entity controls the caching system and its data. Regular proxies are deployed by ISPs or sometimes end-users, and thus every ISP and/or user likely has its own cache of data. There is one proxy cache likely in use for a given user regardless of what web servers they visit. CDNs, on the other hand, are hired by companies and organizations to provide caching servers for their content. The net effect is that a given web server has just the CDN(s) that it has chosen to use, regardless of where its users are. Thus, a given user may be making use of multiple CDNs but a typical content publisher will have only one or sometimes two to three CDNs used for their content.

When battling illicit content, it is important to recognize where the sphere of control is for a given content issue. Note that content publishers can, and some do, make use of multiple CDNs for a given service (discussed further in Section 1.3).

Reactionary vs pre-publishing availability

Local proxies controlled by users are entirely reactionary: when a user requests a file that it does not have a copy of, the proxy must first pull the file from the content publisher. CDNs can operate in this fashion, but it is also possible for client publishers to "pre-publish" content by pushing it to CDN caches ahead of a release. This is common for large video content creators, OS updates, video game releases, etc in order to ensure the information is available ASAP when release announcements are made.

An important take-away when battling illicit content is that content being distributed from a CDN may be cached in just a handful of locations around the Internet, or at every instance of a CDN that content has been pushed to. This, in turn, may impact legal actions where political boundaries are being crossed and illicit content is not being hosted from the country where the legal system is being invoked. This is further complicated when CDN instances are deployed in multiple countries with different legal jurisdictions. Thus, the easiest path forward is likely to work with the CDN di-

[2] https://www.rfc-editor.org/rfc/rfc3143.html

rectly if the content publisher is violating the CDN's Terms of Service (TOS). Unfortunately, some hosting providers that specialize in refusing abuse actions have been known to deliberately remove content only from instances in the region where legal enforcement is being exercised. This is further discussed in the next section.

1.3. Motivation for CDNs

CDNs are used for a variety of reasons from large enterprises to individual websites. Although the obvious reason is to "serve content faster to consumers" (also known as a "happy eyeballs" problem), other reasons include transmission cost savings, protecting sites against Distributed Denial of Service (DDoS) attacks, preserving publication privacy, and to simply outsource content delivery to experts.

Serving content quickly is critical for both legitimate businesses and illegal ones. As mentioned previously, of the most famous studies was performed by Amazon in 2007, which found that a half-second delay in serving a web page to end-users resulted in a 20% drop in traffic rates. It is worth noting that the speed of light limits communication to opposite points in the world to a half-second round trip, and that multiple round trips are needed to fully serve large content such as images and videos.

These and other results highlight the reason for everyone depending on online income derived from serving web content to users to have a high incentive to use the best possible, widest distribution mechanisms. Anything else would mean losing clients, regardless of whether they were selling products or distributing illicit material for profit. However, there are significant prioritization differences between legitimate and illegitimate content publishers, which is further discussed in Section 1.3.1.

Serving content in a reliable fashion that accounts for CDN unavailability is important as well. Highly popular sites are known to use high-end distribution architectures that can make use of multiple CDNs so that if one becomes unavailable or underperforms, others are positioned to take over. For example, Hulu was observed in 2012 to rotate their traffic evenly across the Akamai, Level3 and Limelight CDNs [3].

Some CDNs also include privacy preserving features: they can hide the true origin of the content being produced, effectively shielding the content publisher from revealing their true identity. If a CDN is hosting all of a client's content, including the main web-page contents itself, then the only connections seen from the user or network analyst will be to the CDN itself – the true source of the content is shielded from view. CloudFlare is particularly well known for its policies in protecting free speech (including hate speech). CloudFlare claims that the courts, not the technical industry, are the right place to make content censoring decisions. Other CDNs take similar approaches as well. This approach is often advertised as a policy that benefits society as a whole, as it shields voices from journalists and oppressed citizens, for example. However, it also shields suppliers of illicit online material such as DDoS-as-a-service and pornography vendors as well.

1.3.1 Motivational differences for commercial vs illicit activities

Although both legitimate commercial businesses and illegitimate businesses have a similar primary requirement to "provide excellent service to our end-users," the tangible sub-requirements may be prioritized differently when achieving this goal.

Commercial customers heavily prioritize speed and robustness to downtime, as discussed above. This combination of principle desires translates to preferring a widely distributed CDN with instances in as many countries as possible to provide high speed transfers and make take-downs from a DDoS attack nearly impossible. Importantly, it is less likely that commercial customers producing their legitimate content will be subject to legal attacks requesting the take-down of their published material.

Illicit material distributors, on the other hand, may prioritize other features of a distribution network higher. Specifically, the desire for privacy-protection and known abuse tolerance may be significantly more important than speed and robustness. Thus, a CDN offering better protection from legal actions may have a reduced global footprint and avoid placing instances in countries with stricter legislation.

Interestingly, this does not necessarily correspond with an intuitive list of countries. For example, although Russia is known to only weakly enforce many illicit activities (e.g. gambling, hacking as a service, black market drugs, etc), they are also known to take a much stricter stance against pornography. Canada, on the other hand, which is a country with a robust western legal system and associated protective laws, turns out to be desired by illicit content distributors since any content abuse complaints require formal legal actions to be filed with the court system, which can be slow, thus offering a time win-

[3] https://ieeexplore.ieee.org/document/6193524

dow of operational up-time protection.

An important take-away of these motivational differences when considering illicit content distribution is that CDNs desired by legitimate commercial companies may differ significantly in characteristics from the infrastructure desired by illicit content distributors. Specifically, illicit content distributors are more likely to make use of “Bullet Proof Hosting” (BPH) type services instead of CDNs, as they are designed and deployed to be significantly more robust to legal actions by sacrificing global placements. Their narrower deployments instead concentrate on regions where legal take-downs will be difficult to undertake, often by using offshore hosting facilities that are outside the easy reach of any sovereign nation.

As the more commonly used BPH services are beyond the scope of this document, we refer readers to the third publication titled “The Hacker Infrastructure and Underground Hosting: Cybercrime Modi Operandi and OpSec”[4] from an excellent series of publications by Trend Micro.

1.3.2 Operational differences for long-term content providers

Legitimate commercial operations optimize for maximum availability within the above constraints, sometimes employing multiple CDNs for robustness, but otherwise do not expect to have long-term issues with their content distribution provider. Business relationships between large companies can be expected to last decades or more.

Because illicit content producers are more likely to expect their infrastructure accounts, be them at CDNs or BPHs, to be closed frequently from legal or other abuse reporting actions, they must optimize their deployments for frequent service provider interruption and replacement. Most BPH services are either transient entities that exist as businesses that stay up only for short periods of time before changing names, or are themselves illicit providers that make heavy use of stolen accounts and short-term leases.

Finding lists of commercial CDNs and their ratings is quite straight forward (see Section 5). However, although the author of this document searched for CDNs that were known to willingly host illicit content, none were easily found or were very transient in nature. In the end, BPH services are the generally preferred mechanism by both illicit material publishers (providing them significant more control) and by infrastructure operators with a wider tolerance of hosting illicit content (allowing them to claim less responsibility over the content being distributed).

```
    mirror_mod.use_z
elif _operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

    #selection at the end -add back the deselected mirror
mirror_ob.select= 1
modifier_ob.select=1
bpy.context.scene.objects.active = modifier_ob
print("Selected" + str(modifier_ob)) # modifier ob is the
```

[4] https://documents.trendmicro.com/assets/white_papers/wp-the-hacker-infrastructure-and-underground-hosting-cybercrime-modi-operandi-and-opsec.pdf

2. Background on Web/CDN related network technologies

To understand how CDNs operate and front information for a more distant publication source, we first must obtain a basic understanding of how normal web traffic flows. From there, we can examine the mechanisms that CDNs use to direct content consumers to the nearest CDN instance without requiring the user themselves to be involved or even aware that this is happening.

2.1. Typical web flow

From the web-browser's point of view, it performs a very simple set of steps every time it loads a web-page. It begins by taking the domain name of the requested web-page (for example *www.example.com*) and translating it into an IP address (for example, *www.example.com* currently translates to the *93.184.216.34* IPv4 address and the *2606:2800:220:1:248:1893:25c8:1946* IPv6 address). IP addresses are the Internet equivalent of a phone number – i.e., just a name alone is not enough to make a phone call.

Once the IP address has been obtained, it reaches out to a web server at that address and downloads the requested page. Since modern webpages have many other embedded objects that each must be fetched in turn, it starts with the domain name where the resource is located (for example, images may be hosted under at a separate name like *images.example.com*), and then translates to another IP address and fetches the object. Complex sites end up performing many DNS requests and web-page loads during this process.

This process reveals a large number of points where a web-page and its content rely on internet providers to successfully transmit information. In turn, this also reveals points where network operators may be willing to block illicit materials from being distributed.

2.2. Web flows with Content Delivery Networks

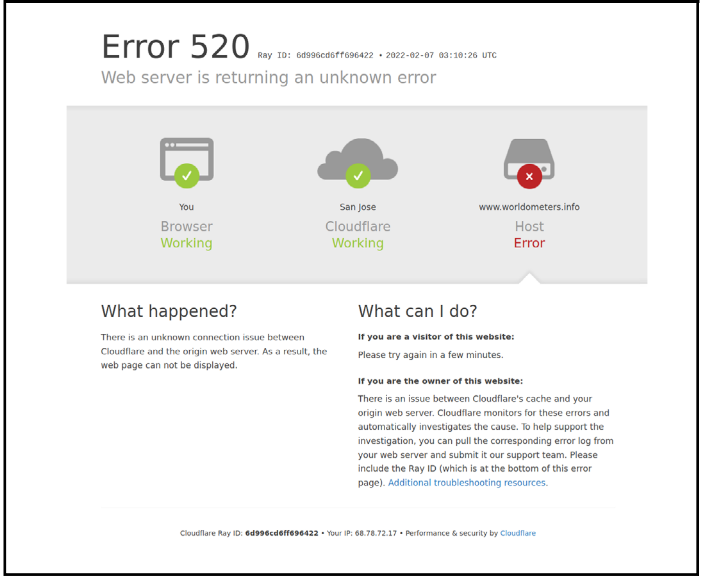
When a website starts outsourcing some or all of its content to a CDN, a number of new transactions happen in the process of a web browser loading a web-page.

Websites typically choose one of two deployment scenarios: 1) they deploy all of their website content on the CDN, including the main home-page content or 2) they deploy only the static content from a website served via the CDN while the dynamic content is served from self-hosted server(s). Static content of this nature includes images, videos, other multimedia as well as general HTML, Cascading Style Sheets (CSS), Javascript and other web-specific data files.

Dynamic sites that change their content every visit or are different for every user, such as social media sites that only show content relating to the viewer's specific friends and followers or news sites that only display content based on viewer interests, must use different techniques for speeding up the main page delivery. Even dynamic web-pages, however, include a lot of static data components that do not change and can be delivered via CDNs.

Content from a publisher can be pushed to a CDN prior to it being requested, or be pulled from a host website by a CDN after a first request from a client has been received.[5][6][7][8] Cloudflare's very popular free service that offers privacy protection services, for example, is known for its simple diagram showing exactly that the source of a failure is outside the scope of their network and the upstream publisher was the source of the problem (see Figure 2.2.1).

Figure 2. A CloudFlare error page identifying that the origin site, not CloudFlare, is the source of a web page's current unavailability.



Error 520 Ray ID: 6d996cd6ff696422 • 2022-02-07 03:10:26 UTC

Web server is returning an unknown error

What happened?

There is an unknown connection issue between Cloudflare and the origin web server. As a result, the web page can not be displayed.

What can I do?

If you are a visitor of this website:

Please try again in a few minutes.

If you are the owner of this website:

There is an issue between Cloudflare's cache and your origin web server. Cloudflare monitors for these errors and automatically investigates the cause. To help support the investigation, you can pull the corresponding error log from your web server and submit it our support team. Please include the Ray ID (which is at the bottom of this error page). Additional troubleshooting resources.

Cloudflare Ray ID: **6d996cd6ff696422** • Your IP: 68.78.72.17 • Performance & security by Cloudflare

[5] https://www.belugacdn.com/push-cdn/#:~:text=In%20push%20CDN%2C%20the%20website,to%20be%20delivered%20to%20visitors

[6] https://cdn.net/push-vs-pull-cdn/

[7] https://www.cachefly.com/push-vs-pull-whats-the-difference/

[8] https://www.quora.com/What-do-most-modern-CDNs-use-push-or-pull

2.3. CDN redirection technologies

Three primary different possibilities exist for how web-browsers are directed to the correct CDN server (aka, “CDN instance”). Web browsers begin by translating the domain name in a URL of a web-resource to an IP address, and CDN-using websites may point directly to an IP address controlled by the CDN (2.3.1), or they may use a CDN that supports on-the-fly web-page rewriting (2.3.3), or they may use domain name redirect records pointing to another name controlled by the outsourced CDN (2.3.2)[9].

2.3.1. Directly mapping website names to CDN IPs

A first solution for directing web clients to the correct DNS instance is to simply give the client domain owner (*example.com*) an IP address controlled by the CDN. The CDN customer should then simply have the domain name point toward that address. Although this works, it is less used in practice as it does not leave flexibility for when the CDN needs to change the IP address to be pointed out without contacting all of their customers that may be using it. When this is used in production, its often when the domain owner is also the CDN owner as ensuring data consistency within a single enterprise is significantly easier. An example of a company deploying this simplistic mapping is google, which returns a simple IPv4 or IPv6 record when a DNS client asks for a name. The resulting address is clearly part of Google’s CDN infrastructure, and rotates over time (see Section 2.3.4), but the record itself is a straight mapping from a domain name to an address record or multiple address records. Table 2.3.1 shows some example records observed for the *www.google.com* domain name.

Name	DNS Answer
www.google.com	142.250.188.228 142.250.72.132

Table 2.3.1 Example address records returned for *www.google.com*

2.3.2. CDN DNS redirects

A more common redirection approach, especially when the CDN is an outsourced entity, is to have the customer map their domain to a DNS record within the CDN’s DNS domains. This allows CDNs to change their infrastructure more easily without disturbing their client’s configuration, as it is far easier to change DNS redirect records than it is to change referred IP addresses. Sometimes, multiple redirects are even used in complex customer/CDN deployment combinations. Table 2.3.2 shows an example of the multi-hop redirection observed when resolving *www.paypal.com*.

Name	DNS Answer
www.paypal.com	www.glb.paypal.com
www.glb.paypal.com	www-fastly.glb.paypal.com
www-fastly.glb.paypal.com	151.101.129.21

Table 2.3.2 Example redirection and address records returned for *www.paypal.com*

2.3.3. On the fly web-page rewriting

Some CDNs support the ability to rewrite web-pages on the fly, such that when a user requests a containing page from their CDN (or through their CDN), the URLs for the other content within that page (images, etc) will be rewritten on the fly to point the requesting web browser to the best available CDN instance based on proprietary selection logic. This technique does require an additional computational burden at the CDN server, however. This also has the advantage of directing users to the right image or video resolution supported by estimating a client’s bandwidth availability.

2.3.4. Loading balancing by rotating IP addresses

Regardless of how a web client is redirected to a particular IP address, two different techniques are used in order to both provide users with a copy of information as close as possible to them and to load-balance instances so customer requests are spread across servers that can handle the results.

The first technique is to give different IP address answers to different users depending on where the DNS request is coming from. A user requesting an address for *www.example.com* in Europe may get an entirely different answer from the United States. Information about the source of the user comes either from IP-to-geographical-location databases using the DNS request’s source address or the EDNS Client Subnet field in the request . Additionally, the answers may also rotate over time in order to distribute users to different CDN instances. Google uses both regional IP address responses to DNS requests and IP address rotation in its global infrastructure.

The second technique is called “IP Anycast”[11], and is an Internet traffic routing technique (using BGP[12]) that allows multiple servers around the globe to advertise the same IP address to users. Because the Internet’s routing system, when it sees duplicate announcements, operates by sending communication to the topologically closest server. This has become a proven method for creating sets of globally distributed servers capable of responding quickly to local client requests. At one point, Cloudflare’s CDN implementation made use of IP Anycast to achieve its global coverage.

[9] https://www.rfc-editor.org/rfc/rfc3568.html
[10] https://www.rfc-editor.org/rfc/rfc7871.html
[11] https://www.rfc-editor.org/info/rfc1546
[12] https://www.rfc-editor.org/info/rfc4271

3. Illicit material distribution contact points

Removing illicit content from the Internet is a daunting and difficult task. To succeed, it is best to consider multiple networking points where distribution disruption can occur. Below, we consider each point in turn ranging from the easiest and most effective to the most difficult.

3.1. The site owner

Whether or not a website is using a CDN, it's worth considering contacting the site owner about hosted illicit content if it looks like it's out of place with respect to the rest of the site. Criminal activity of all types often uses compromised sites and accounts to host material. Frequently, the real site operator has no knowledge that their system has been compromised and is distributing illicit material. However, a balance must be considered about the legitimacy of the rest of the site – accidentally contacting a site with a fake cover may indicate to them that their otherwise illegal activity has been caught and will cause them to retreat to other sites, hosting companies, compromised accounts and CDNs.

When contacting a site owner, it is important to consider checking both the owner of the domain name through the WHOIS or RDAP protocols[13], and the owner of the IP address serving the content through WHOIS or other reverse lookup protocols and databases[14]. If, however, the domain name redirects to either a name and/or IP addresses owned by a CDN, then the real site owner has effectively hidden themselves from easy identification and there may be no choice other than contacting the fronting CDN instead.

3.2. CDNs

Most CDNs have Terms of Service (TOSs) or Terms of Use (TOUs) that prohibit the distribution of illicit content[15] [16] [17] [18] [19] [20]. Thus, a key aspect of effectively removing CSAM and other material hosted at CDNs will be to establish contacts at CDNs that can be reached quickly and efficiently.

Unfortunately, CDN abuse requests are acted on at different speeds based on the size and complexity of the CDN, the importance of the request, and who is making the take-down request. Requests to law enforcement for the removal of malicious content has similar issues with the volume of requests they receive, as they typically concentrate on the cases with the highest impact. Unfortunately, this means there may be a delay between identifying and reporting any illicit material and when it is removed. Well-exercised contact databases and good relationships are the best key for battling problems with slow responses.

After illicit content is taken down from a particular service, however, one should expect the publisher to immediately replace the affected CDN or hosting service with another. Even legitimate commercial companies understand the importance of using multiple CDNs to ensure their content will always be accessible even when one hosting CDN has a catastrophic failure.

For example, as mentioned above, Hulu balances their traffic among three different CDNs. There is also no reason why illicit sites will not use the same balancing techniques, and there is reasonable evidence that many already do. As an example, the Internet forum *8chan*, which hosts extreme opinions and content posts with little to any filtering, has been kicked off of both a network access point (Clearnet) and a CDN (CloudFlare) in the past. This did not defer the site owner, however, which simply moved to using Voxility and the Epik web hosting company which are more resilient to abuse requests.

3.3. DNS Services

Almost all websites and content URLs contain a domain name that is used to direct web browsers to a given Internet IP address. Thus, the web's content is heavily reliant on DNS registrations and resolution.

Therefore, one source of illicit content intervention is with the DNS service components that enable this name for IP address translation. The original web content publisher has a domain name associated with their primary site, and the

[13] https://lookup.icann.org/en
[14] https://iptoasn.com/
[15] https://www.cloudflare.com/website-terms/
[16] https://www.keycdn.com/terms
[17] https://www.stackpath.com/legal/privacy-statement
[18] https://docs.microsoft.com/en-us/legal/intune/acceptable-use-policy-for-microsoft-intune
[19] https://bunny.net/acceptable-use
[20] https://www.fastly.com/acceptable-use/

CDN service also has a domain name associated with their service. As discussed in Section 2.3.2, it is also common for principle domains to use domain name redirection to refer to the CDN in question. Simply put, if any of this chain fails, then the content will no longer be accessible.

Multiple parties are involved in the registration of domain names, each referred to by a different term. *Registrants* are the users that wish to register (or renew) a domain name. A *Registrar* is the company that accepts the registration and payment, and transmits the request to the company or organization (*Registries*) that runs the parent part of the DNS tree. This model was created to allow competition among the many Registrars even when each Top Level Domain (e.g. *.com*, *.net* and *.org*) could only be operationally run from a single company. Finally, most domain owners outsource the operational aspects of running a DNS server to a third party. More often than not, users will select the Registrar to also run their DNS service, as most registrars offer both the registrar and DNS operational services under a single price. Keeping services together simplifies customer billing, configuration management and operations.

Both the Registrar and the Registry can be points of contact if the customer is violating either of their Terms of Service. The company responsible for running the registered zones, DNS services, are likely to be the best place to start, as they are more likely to have stricter Terms of Service than DNS registrars and registries.

3.4. ISPs

All network elements discussed above require connections to the Internet through some sort of Internet Service Provider (ISP). The easiest ISP to contact when trying to remove illicit content is the upstream ISP used by the content publisher. This point of contact will only be beneficial when the source publisher can be identified and is not behind a CDN providing privacy protection.

Any other ISP fronting the rest of the infrastructure (e.g. DNS services and CDN services) are less likely to be persuaded to turn off Internet access for the service, as it will affect all the service's clients and not just the ones hosting illicit materials. Though still worth considering as a contact point, unless that service can be shown to be mostly malicious or illicit content, it is unlikely that an ISP would be willing to turn off access to an otherwise significant internet resource.

Also note that very large CDNs and DNS infrastructure servers functionally are ISPs themselves, but still must be connected to other Internet backbones to achieve maximum connectivity. The architecture of these arrangements is beyond the scope of this document.

4. Identifying and gathering information to report illicit material

Regardless of which point of contact might be reached, gathering needed and accurate information to pass along will be critical in assuring the most rapid action possible. Specifically, *What* content was observed and why it was offensive, *Where* the offensive media was located (both the source page and the URL of the media itself), *Where* the viewer was located since the regionality of the viewer may be important, and *When* the content was seen (with as precise timing as possible, including time zone information).

Unfortunately, collecting the source URL of an image can be challenging. Although web browsers support a right-click (or control-click on a mac) option to bring up a context menu allowing an image's source URL to be copied, this is not always available. Modern web user interface frameworks frequently load images in Cascading Style Sheets and Javascript in ways that disable the ability for web browsers to show this option in the context menu. Furthermore, other sites completely disable context menus to prevent users from easily examining the content's source. (This is done for beneficial reasons too, such as when implementing copyright protection for commercial digital content.)

None of these problems are daunting to an experienced web-developer who understands how these techniques function. As a result, it is critical to ensure that a staff member has the technical knowledge enabling them to study the underlying HTML, CSS and Javascript code, which will always reveal the content's source URL.

5. CDN popularity

There are now over 100 different CDN companies. Note that some companies have built an internal CDN that they use for themselves, while also offering one to customers (Amazon and Google fall into this category). Many cloud hosting providers also dual-perform as a CDN, which includes the popular Amazon Web Services and Microsoft Azure platforms.

CDN customers are typically divided into two groups: large enterprises that have more complex needs, and smaller businesses and personal or organizational groups. Some companies primarily target the larger enterprises, while others allow for easier small business uses. CloudFlare dominates the market in terms of number of customers overall, but much of their customer base consists of small businesses or personal users making use of their free account. When examining CDN popularity among the larger enterprises, CloudFlare, Amazon Web Services and Akamai are nearly neck-and-neck in terms of large, enterprise customers.

In general, the list of CDNs below are well-respected as the top CDN providers today [21] [22] [23] [24] [25] [26]. (Note that we list these in alphabetical order, since ranking order differs depending on what metric is being measured and/or prioritized. For example, some rankings are based on latency, while others are on number of customers, etc).

- Akamai
- Alibaba Cloud
- Amazon CloudFront
- BelugaCDN
- Bunny CDN
- CDN77
- CDNsun
- CacheFly
- China Cache
- China Net Center
- Cloudflare (free service)
- Cloudinary (free service)
- Fastly
- G-Core CDN (free service)
- Google Cloud
- Hostry Free CDN (free service)
- Imperva
- KeyCDN
- Leaseweb
- Limelight
- Lumen
- MetaCDN
- Microsoft Azure
- NetDNA
- StackPath
- Tata Communications CDN
- Tencent Cloud
- Verizon Edgecast CDN

The above list includes some markings about the service having a free version, which will be particularly attractive to smaller and personal distributors that are looking for identity protection when hosting illicit materials. Thus, those are potentially more important to develop contact relationships with when looking to interrupt distributors that are not trying to profit from redistribution.

Wikipedia also has a constantly updating list of CDNs, along with their intended audience and purposes (including other Free CDNs)[27].

[21] https://websitesetup.org/best-cdn-providers/

[22] https://www.techradar.com/news/best-cdn-providers

[23] https://firstsiteguide.com/best-cdn-providers/

[24] https://www.enterprisenetworkingplanet.com/guides/cdn-providers/

[25] https://haydenjames.io/best-cdn-providers/

[26] https://www.gartner.com/reviews/market/global-cdn

[27] https://en.wikipedia.org/wiki/Content_delivery_network#Notable_content_delivery_service_providers

6. Glossary

Content Publisher

A term used in this document to refer to the entity that publishes material on the web, regardless of the original source of the content. Note that they may not be the original source of the content. Multiple content publishers may also be publishing the same material.

Web Proxy[28]

A server application that acts as an intermediary between a client requesting a resource and the server providing that resource.

CDN instance

CDNs are most effective when multiple web servers are placed around the world at different locations for both load sharing and providing the fastest speed for regional clients. Each hosted site, in this document, is referred to as a *CDN instance*.

Domain Name System (DNS)[29, 30]

The Domain Name System (DNS) is the hierarchical and decentralized naming system used to identify computers reachable through the Internet or other Internet Protocol (IP) networks.

Domain Name[31]

A name accessible within the DNS hierarchy. Domain Names are typically thought of as the registration point that an owner will register (for example *example.com*) underneath a Top Level Domain (*.com*). In addition, each individual name, including any prefixes, are each technically considered individual domain names. For example, *example.com* may be a delegation point, but *www.example.com* and *images.example.com* are also considered domain names even though they may be part of the larger *example.com* DNS zone.

DNS Registry[32]

The administrative operation of a zone that allows registration of names within that zone. People often use this term to refer only to those organizations that perform registration in large delegation-centric zones (such as Top Level Domains (TLDs)); but formally, whoever decides what data goes into a zone is the registry for that zone. This definition of “registry” is from a DNS point of view; for some zones, the policies that determine what can go in the zone are decided by zones that are superordinate and not the registry operator.

DNS Registrant[33]

An individual or organization on whose behalf a name in a zone is registered by the registry. In many zones, the registry and the registrant may be the same entity, but in TLDs they often are not.

DNS Registrar[34]

A service provider that acts as a go-between for registrants and registries. Not all registrations require a registrar, though it is common to have registrars involved in registrations in TLDs.

Bulletproof Hosting (BPH)[35]

BulletProof Hosting services provide criminal actors with technical infrastructure that is resilient to complaints of illicit activities, which serves as a basic building block for streamlining numerous types of attacks.

[28] https://en.wikipedia.org/wiki/Proxy_server
[29] https://www.rfc-editor.org/rfc/rfc1034.html
[30] https://en.wikipedia.org/wiki/Domain_Name_System
[31] https://en.wikipedia.org/wiki/Domain_name
[32] https://www.rfc-editor.org/rfc/rfc8499.html
[33] https://www.rfc-editor.org/rfc/rfc8499.html
[34] https://www.rfc-editor.org/rfc/rfc8499.html
[35] https://ieeexplore.ieee.org/document/7958611