Project Report: Measuring Facebook's off-net deployments

Elena Frank
Karl Skomski
Leonard von Lojewski

1 INTRODUCTION

Over the last decades, the internet traffic has been heavily consolidated. While it took thousands of Autonomous Systems (ASes) to contribute half the internet traffic in 2007, it was only 150 in 2009 [1]. In 2019 half of the outbound traffic in the internet stemmed from five ASes including Google, Netflix and Facebook[1]. These ASes have been dubbed Hypergiants (HGs). To deliver their content to billions of users around the world, these Hypergiants have to push the cutting edge of internet infrastructure.

To deliver content with very low latency and high speed having some datacenters with high speed connections is not enough for HGs. Pushing that much data through the internet core is not practical. Luckily this type of content (lots of the same information being consumed by many users) lends itself well to caching. These caches are pushed as far to the edge of the internet as possible. It has become the common practice among HGs to put caches in the datacenters of Internet Service Providers (ISPs). These so-called off-net caches are deployed as close to the customer as possible and reduced both latency for the user and upstream traffic for the ISP.

In this project we examine the deployment of off-net caches for Facebook in particular. More specifically we try to replicate the findings of [2]. For this we have collected fresh data ourselves and analyzed it using the methodology from the paper. We have limited our research to only Facebook in order to reduce complexity introduced by possible customer websites being hosted by the HG. However, our approach should be applicable for all big ASes. Our research question is: What is the current AS footprint of Facebooks off-net deployments?

In contrast to the original paper, we have not analyzed existing datasets, but collected our own data to analyze.

2 BACKGROUND

2.1 HTTPS/TLS

Over the last years, the internet has moved away from using plain unencrypted HTTPS for privacy reasons, and instead uses an encrypted version called HTTPS. HTTPS uses the Transport Layer Security (TLS) protocol for the encryption. The TLS handshake includes three simultaneous processes: (1) agree on a common TLS version to use, (2) establish a common session key used for symmetric encryption, and (3) exchange a cryptographically signed certificate. So TLS is not only used for encryption, but also verifies the identity of the server toward the client using this certificate. The TLS certificate contains information about the server, its identity and the organization it belongs to. In the certificate there are multiple fields that contain information about the organization using the certificate (e.g. organization name, city, state, country), along with domain names the certificate is valid for. All of this information is signed by the issuing certificate authority (CA).

During the TLS handshake the certificate is sent from the server to the client and the client can now verify, that the certificate is real, and the domain it is trying to call matches the one in the certificate. Additionally the client could verify that the organization information from the certificate matches its expectations (this is not commonly done however). Therefore, if a server wants to serve content for a specific domain, it will need a certificate to match this domain.

3 METHODOLOGY

Idea. The idea in the paper is to use the fact, that that all servers serving content for a specific service will need a TLS certificate to authenticate themselves toward a client. So once we know which domain names and certificates are used by the HG of interest, we can easily identify which servers serve content for the HG by matching its certificate against the list we know. To find this list of certificates, we can scan the IP addresses announced by the HG for their certificates.

However, multiple of the HGs not only serve content for their own services but also for customers. To filter those hosts, the paper combined two different techniques: First, the information from the certificates can be used to only take into account those certificates whose Organization field contains the name of the HG in question. This alone is not sufficient to match only content served for the HG, as some of them also issue certificates for their customers or serve content directly. So as a secondary filter, they manually identified

[1]For simplicity we will refer to the company Meta (formerly called Facebook) and the service Facebook both only as *Facebook*

HTTP headers that are unique to the server configurations used by each HG.

Parts. Our methodology is heavily influenced by [2], however due to limitations in time and complexity some adaptations had to be made. Our research can be split into three stages: (1) an **on-net scan** to establish fingerprints for hosts serving Facebook content, (2) an **off-net scan** to find hosts matching the fingerprints in the internet, and (3) **AS analysis** to turn the list of hosts into insights about the AS footprint of Facebook.

3.1 On-Net Scan

Before we can scan the internet for hosts serving Facebook content, we need to establish how to identify such a host. For this we will try to find common characteristics of the hosts whose IP addresses are announced by the Facebook AS, as we can be sure, that these are actively serving Facebook content. As in the paper we are interested in all hosts that respond to HTTPS requests on port 443. In this step we seek to find all the different TLS certificates that are used by Facebook. Later we can try to match the certificate of another host against our list and if it is in the list, we assume the host is serving content for facebook.

First we had to find the on-net IP prefixes announced by the Facebook AS. Here we used a public BGP dump from RIPE. For the scan we used the tool zgrab2 on some of those prefixes[2]. After the zgrab scan, we used a python script to go through all responses and identify all certificates with the organization field matching *"Facebook, Inc."*. This resulted in a list of 15 certificates.

In the paper they used an additional heuristic for their identification: an HTTP header unique to each HG. This was done, to help separate customer hosts with a certificate from the HG from hosts serving actual HG content. However, we focused our analysis on Facebook to limit the scope of the project. As Facebook does not host content for customers, all the hosts with Facebook certificates are serving Facebook content. We did not run our own analysis to find unique HTTP headers, but in the paper they identified the HTTP header *x_fb_debug* as being unique to Facebook, so we did also include it in our analysis. This also serves to make our approach more generalizable.

Off-net cache heuristic. Now we have established a heuristic for a Facebook content serving host. We identify a host as an off-net cache, if and only if all of the following criteria are met:

- the IP address of the host is not announced by the Facebook AS
- the host responds on TCP port 443
- the TLS certificate is identical to at least one of the 15 certificates found in the on-net scan
- the HTTP header *x_fb_debug* is present

[2]The prefixes used for the on-net scan were: *31.13.24.0/21*, and *31.13.64.0/18*

3.2 Off-Net Scan

After establishing a heuristic what constitutes a Facebook content serving host, we can now scan the internet to look for hosts matching this heuristic.

For this internet-wide scan we needed to depart from the methodology. In the paper, they used a complete dataset and analyzed it after the fact. Due to a limited amount of time and storage space on the server running the scan, this was not an option for us. Our first approach was to use zgrab2 to scan the entire internet and use its filtering tools to match hosts immediately during the scan. Then we would only output the IP address of the host, if it matched our heuristic. However, we had to abort this scan quickly, as it was too slow. The issue was that we scanned the entire internet, of which most addresses are not reachable on TCP/443 and therefore zgrab would wait for the timeout to occur.

Iterating on our first approach, we wanted to reduce the number of IP addresses zgrab would hit several orders of magnitude. For this we ran a zmap scan for TCP/443 where our throughput was about 70,000 packets/second. This resulted in a list of approx. 54 million IP addresses responding on TCP/443, on which we then ran the zgrab scan.

Now we could scan the internet for our hosts in a reasonable amount of time, however, the amount of data returned was still too much. Therefore we developed a python script to parse the zgrab output on the fly and identify hosts matching our heuristics. Only if the host matched our heuristics, would we store its response.

To recap, our pipeline consisted of the following steps:

(1) use zmap to find all hosts responding on TCP/443
(2) use zgrab2 to get the certificates and HTTP headers of those hosts
(3) use a python script to identify off-net hosts

4 RESULTS & DISCUSSION

Since this is a proof of concept and not a comprehensive research paper, we did not run all tests to their fullest extent. To reduce both the time needed for scanning as well as the complexity of the analysis, we only ran most scans partially. However, we did ensure that all of the techniques we used would scale reasonably well and work with the full complexity.

4.1 Analyzing the Facebook on-net

TLS certificates. For the on-net scan, we only scanned the two prefixes *31.13.24.0/21*, and *31.13.64.0/18*. Of the total possible 18,432 IP addresses in this range we found 1,073 online

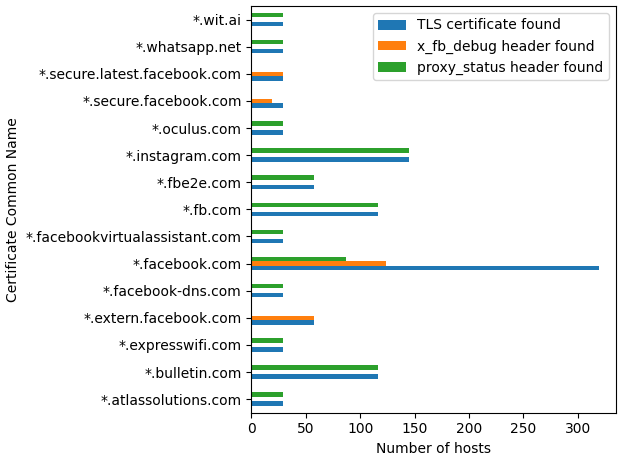

Figure 1: Number of hosts with a Facebook TLS certificate and the headers *x_fb_debug* and *proxy_status*. We can clearly see, that most servers don't serve the *x_fb_debug*

hosts, which corresponds to a hit-rate of approximately 5.82%. Analyzing the responses, we found 15 certificates that had the organization field of the TLS certificate set to *"Facebook, Inc."*. All certificates we found are listed in appendix A.

For nine of the 15 common names found in the certificates, our scan found exactly 29 hosts serving that particular certificate (see Fig. 1).

HTTP headers. Further, an analysis of the headers showed that only 319 of the 1,073 hosts actually sent the header *x_fb_debug* used to identify Facebook servers in the paper. We can also see that this header is specific to hosts serving these three common names: **.secure.facebook.com*, **.extern.facebook.com*, and **.secure.latest.facebook.com*. For the common name **.facebook.com* some servers send the header and some don't. This implies that not all Facebook companies use the same software to serve their content or at least they do not use the same configuration. The former hypothesis is reinforced by the fact that companies such as Instagram and WhatsApp were acquired by Facebook and their technology stack is therefore distinct from the one used by the Facebook service.

We did find another header that is specific for Facebook: *proxy_status*. The value includes sub-keys whose names indicate they are custom for Facebook: *e_fb_vipaddr*, *e_fb_builduser*, *e_fb_binaryversion*, *e_fb_proxy*. This header with the aforementioned sub-keys can be found in 725 of the on-net servers responses. Interestingly, there is no single server that responds with both headers and 29 servers that respond with neither. As can be seen in Figure 1, except for domains of **.facebook.com*, per common name all servers only send either one of the headers. All servers sending neither header have the certificate for **.facebook.com*.

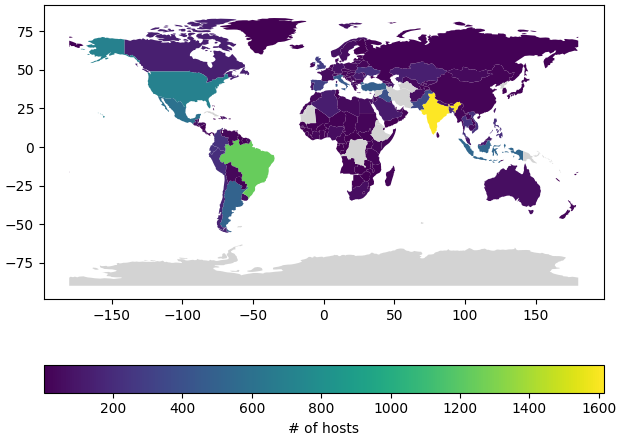

Figure 2: Number of off-net hosts. The country is taken from the AS owning the IP address. The top 5 countries by number of off-net hosts are: India (1614), Brazil (1239), United States (703), Mexico (601), and Indonesia (536).

4.2 Searching for Facebook off-net servers

After running the zmap scan on TCP/443, we found about 54 million hosts. For these we ran the zgrab scan and analyzed the output of zgrab on the fly. To reduce the amount of data we needed to store, for every IP address we only kept the response if it we actually identified the server to be a Facebook server.

From the 54 million hosts we found to be online, we have identified 17,657 hosts to be serving a Facebook certificate[4] (including 1,493 on-net servers). Of these, 432 hosts additionally sent the header *x_fb_debug*. This implies that in contrast to the paper, this header is not a good indicator of a Facebook content serving host at this moment. Eight hosts only sent the header, without a Facebook certificate. After manual inspection it appears that all but one have no connection to Facebook and we have no hypothesis as to why these hosts serve this header[5].

[4]Since we did not scan all Facebook prefixes, we were not confident that we had collected all of their TLS certificates. Therefore we did not match the certificate found in this step to the list of certificates from the on-net scan, but instead just looked for the correct organization. We can assume that this is a good indicator of a valid certificate from Facebook, as we checked the Censys database and found that only a negligible number of certificates with that organization were invalid.

[5]The common names in the certificates served by these hosts are: *protected.cudadps.com* (twice, NXDOMAIN, *cudadps.com* owned by Barracuda

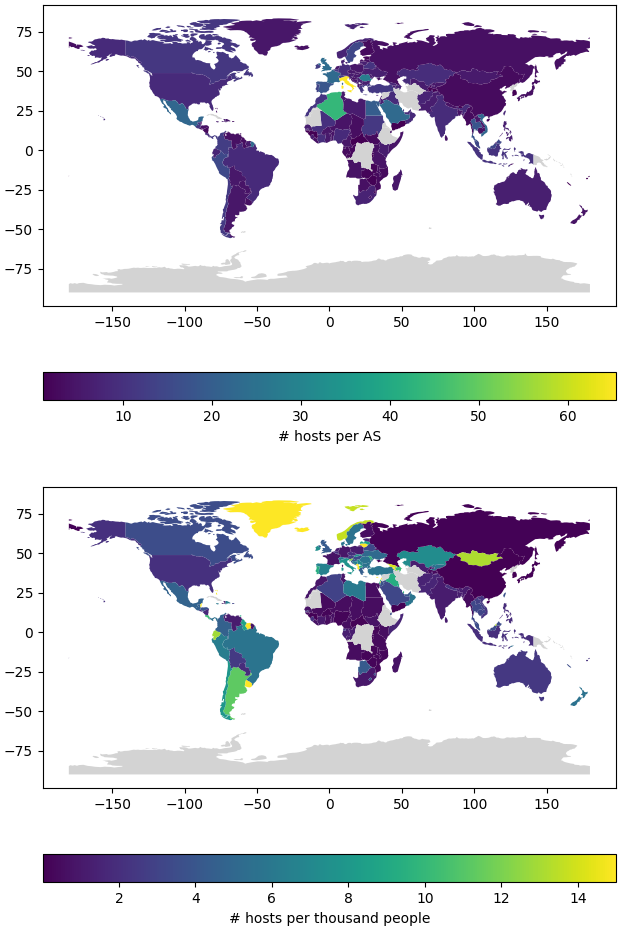

Figure 3: Number of off-net hosts per AS (top) and off-net hosts per thousand people[3] (bottom). The country is taken from the AS owning the IP address. The number of hosts per thousand people is capped at 15 to remove outliers (e.g. very small countries with very small population). We can see that the density of hosts is highest in Latin America and there is a general trend of more hosts per inhabitant for countries of higher populations. It is noticeable that Kazakhstan and Mongolia (both directly bordering China) have a very high off-net deployment density even though they do not have large populations.

Using the IANA whois service we have mapped all hosts to their ASes. In total the we have found 16,164 hosts in 2,339 ASes (excluding the Facebook on-net). By mapping the

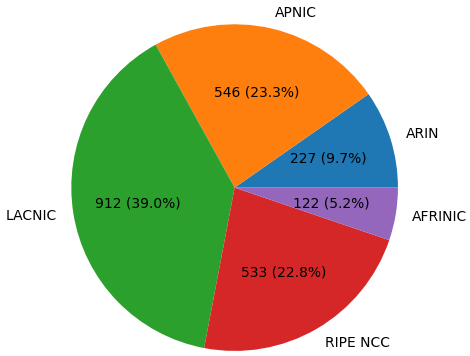

Figure 4: Number of hosts per Regional Internet Registry (RIR). A host is mapped to a RIR by the AS announcing the hosts IP address. The AS number was mapped to a RIR using [4].

ASes to countries (see Fig. 2) and Regional Internet Registries (RIRs) (see Fig. 4) we can see, that the largest portion of hosts is in Latin America. However, the largest number of hosts can be found in India and Brazil which together have about 18 percent of all off-net servers. This observation is comparable to the one made in the original paper.

Analyzing the data geographically we can observe, that the countries with by far the most off-net hosts are India (1614 hosts; 10.0%) and Brazil (1239 hosts; 7.6%). This seems logical as both are large countries with large populations and geographically distant from one another. When looking at the number of hosts per person (see Fig. 3) however, India is now among the average. Here Latin America ranks much higher. We can speculate that this has to do with a very large user base living in urban areas. In order to lower bandwidth, many servers might be needed to serve urban communities. In contrast, India is more rural and therefore less servers are required as their primary purpose might be latency reduction. Unfortunately this is just speculation and from just the data we cannot assert its validity.

5 CONCLUSION

In this research project we have developed a complete process for finding the footprint of a HGs off-net deployments. The methodology was heavily inspired by [2]. Not only did we analyze existing data to find this footprint, but we also collected all of the necessary data on our own. To achieve this we needed to reduce the complexity for some of the steps. However, we did produce outputs for all of the steps and ensured, that the techniques we used were applicable

Networks, hosted by Microsoft), **.jawaker.com* (twice, hosted by CloudFlare), *gdongbear.com* (hosted by Alibaba), **.polri.go.id* (owned/hosted by the Indonesian National Police), *butterfly-hk.stillwater.cn* (hosted by Alibaba), and **.carriersignal.info* (apparently owned by Facebook)

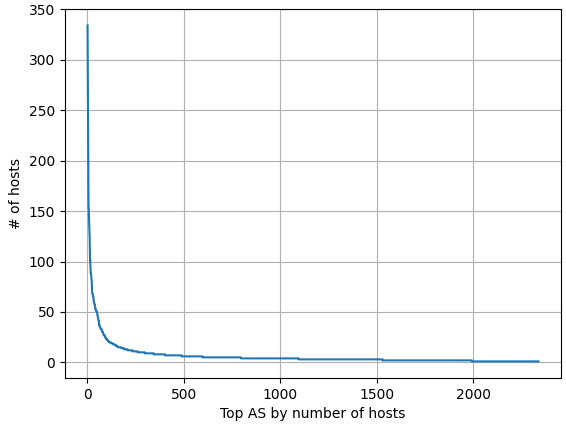

Figure 5: Number of hosts per AS. This clearly shows, that there are a limited number of ASes that make up the majority of off-net hosts. The median number of hosts per AS is just 3.

and scalable for a comprehensive research paper. In our analysis we identified 16,164 off-net hosts in 2339 ASes serving content for Facebook users.

We did show that the idea behind the methodology of the original paper does work. Our results are comparable to the results shown in the paper. But we did also show, that not all steps taken in the paper are always needed or useful for all Hypergiants. For Facebook we saw, that the header-based filtering would have reduced the number of identified off-net deployments massively.

An area for improvement is validation of our methods. We only used our analysis methods on the data we collected ourselves. The paper mostly analyzed data collected by third parties (Censys and Rapid7). Comparing the results from our methods on third-party data to their results would be very interesting. Additionally the methods from the paper could be used on our data. This would give a good insight into how well our data collection works and how well we replicated their methodology.

6 DECLARATION

This research project was done for the Internet Measurement module at Freie Universität Berlin during the summer term 2022. The code is available at https://github.com/elenaf9/hypergiants-offnet-footprint.

REFERENCES

[1] C. Labovitz, "Internet Traffic 2009-2019," 2019.

[2] P. Gigis, M. Calder, L. Manassakis, G. Nomikos, V. Kotronis, X. Dimitropoulos, E. Katz-Bassett, and G. Smaragdakis, "Seven years in the life of Hypergiants' off-nets," in *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, ser. SIGCOMM '21. New York, NY, USA: Association for Computing Machinery, Aug. 2021, pp. 516–533. [Online]. Available: https://doi.org/10.1145/3452296.3472928

[3] United Nations, "World Population Prospects - Population Division - United Nations." [Online]. Available: https://population.un.org/wpp/Download/Standard/CSV/

[4] IANA, "Autonomous System (AS) Numbers." [Online]. Available: https://www.iana.org/assignments/as-numbers/as-numbers.xhtml

A UNIQUE CERTIFICATES

C=US, ST=California, L=Menlo Park, O=Facebook, Inc.

- CN=*.facebook.com
- CN=*.instagram.com
- CN=*.facebookvirtualassistant.com
- CN=*.bulletin.com
- CN=*.atlassolutions.com
- CN=*.wit.ai
- CN=*.fb.com
- CN=*.expresswifi.com
- CN=*.whatsapp.net
- CN=*.facebook-dns.com
- CN=*.fbe2e.com
- CN=*.oculus.com
- CN=*.extern.facebook.com
- CN=*.secure.latest.facebook.com
- CN=*.secure.facebook.com

B TOP 15 ASES BY NUMBER OF OFF-NET HOSTS

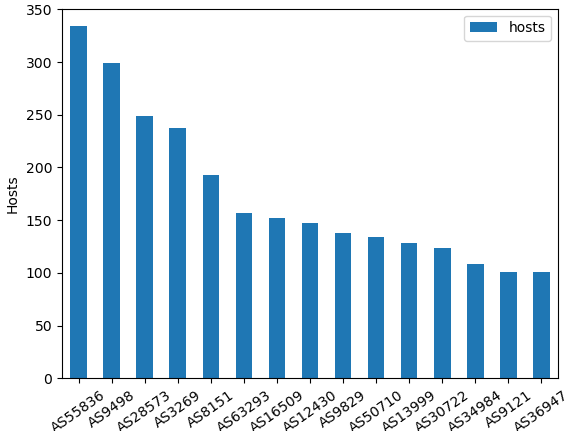

Figure 6: Top 15 ASes by number of hosts. These make up about 16% of the total Facebook off-net hosts.