Towards Detecting and Geolocalizing Web Scrapers with Round Trip Time Measurements

Elisa Chiapponi
EURECOM
Biot, France
elisa.chiapponi@eurecom.fr

Marc Dacier
RC3, CEMSE, KAUST
Thuwal, Kingdom of Saudi Arabia
marc.dacier@kaust.edu.sa

Olivier Thonnard
Amadeus IT Group
Villeneuve-Loubet, France
olivier.thonnard@amadeus.com

I. Introduction

Web scraping is the activity of continuously extracting data and/or processed output contained in web pages [16]. This process generates large amounts of traffic on the targeted websites, which suffer large losses of money from this phenomenon [3]. For this reason, they engage in a persistent fight against scrapers, trying to detect and mitigate their requests [7].

Lately, scrapers have taken advantage of Residential IP Proxies (RESIP) to perform their requests [5]. These parties give access, for a fee, to a vast network of residential devices that can be used as exit points for requests. Taking advantage of these infrastructures enables scrapers to send requests from IP addresses used by real users. This reduces the confidence of scraped websites in blocking their requests to prevent false positives. Recently, researchers started to study RESIPs, revealing some of their features, how to detect if a device is acting as a RESIP machine and their association with various malicious activities [11], [18], [12], [9], [8], [6], [14], [13].

In our past work [4], we collected a new dataset of RESIP connections. We have used it to validate a new server-side detection method based on network measurements. It can systematically and deterministically detect a RESIP connection by only analyzing a single request, differently from a recently proposed machine learning one [15]. Our approach considers the differences at the transport layer between the connections produced directly by a device and the ones proxied through it. In the first case, the TCP and TLS sessions are built between the same two parties: the device and the server of the target website. In the second case, the TCP session is built between the device and the server while the TLS session takes place between the scraper exploiting the RESIP infrastructure and the server. This difference in the setup can be identified on the server side thanks to distinct Round Trip Time (RTT) measurements.

In a direct connection, all the exchanged packets between the device and the server cover the same distance. In a proxied connection, the RTT_{TLS} is influenced by the distance between the scraper and the server but also by the path of the packets inside the RESIP infrastructure. Moreover, the RTT_{TCP} only reflects the distance between the device and the server. We take advantage of this difference (δ_{RTT}) in the RTTs to detect when a connection passes through a RESIP service.

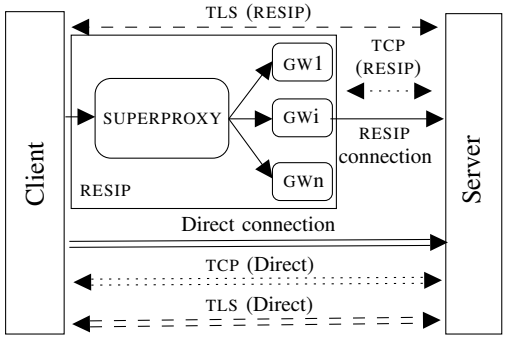


Fig. 1: Experimental Setup.

Besides the detection method, the collected dataset enabled us to gather new insights about RESIP providers, showing the similarities and differences of the associated ecosystems (geographic distribution, types, management and amount of used machines) [2].

In this new work, we share new insights about this dataset. We analyze the mean average speed of packets in the collected connection. We show the high variability of this parameter. This attests that our setup is representative of the real world and that we can apply our technique in any real-world scenario (Section III). We present a study of the impact of the geolocalization of real initiator, server and RESIP machines on our technique. We show that the δ_{RTT} decreases, as expected, in the case where all the machines involved in a connection are in close proximity. However, the difference remains high enough to enable a significant level of detection in such worst-case scenario (Section IV). Moreover, we describe the next steps in our research work (Section V). Thanks to the positive results of our experiment, we managed to implement the technique in front of domains suffering from scraping. The study of these connections is ongoing. Furthermore, we are trying to exploit the δ_{RTT} to geolocalize the position of the real initiator of a connection behind a RESIP, performing triangularization from the used GATEWAYs. We present our idea and how we plan to continue working in this direction.

978-3-903176-58-4 ©2023 IFIP

II. Background

In this section, we briefly recap how the dataset was collected. We refer to [4] for a complete description. We have acquired 22 machines, running as client and server, distributed worldwide and four RESIP services among the ones most used for scrapers. As displayed in Fig. 1, each client connects to each server through a RESIP provider (RESIP connection) and directly (Direct connection), embedding in the URL information about the involved parties. In the first case, the client sends the requests to the RESIP entry point, the SUPERPROXY, which chooses one of the residential GATEWAYs (GWi) to proxy the request out. In this case, the server builds the TCP session with GWi and the TLS session directly with the client. In a direct connection, the two sessions are both built between client and server. On the server, we take network measurements and we calculate the δ_{RTT} = RTT_{TLS}- RTT_{TCP}. If this difference is greater than 50ms we declare the connection as coming from a RESIP. We have run our collection for 4 months, in which we have gathered 92,712,461 connections. As showed in [4], our technique had an accuracy of 99.01% in classifying when a connection was direct or was passing through a RESIP.

III. Measurements Representativeness

Our detection technique is based on the RTT mirroring the distance between the parties involved in a connection[1]. However, the RTT is a measure of time. To transform it in a distance, we need to consider the speed at which packets move. If this speed had a common mean value, there would be a proportional factor between each RTT value and the corresponding traveled distance of a packet. This factor would be common to all the connections and this would favor the success of our detection technique. However, the real world does not present perfect conditions. As acknowledged by Weinberg et al. [17], an idealized common value for the average speed does not exist in practice for connections across different areas of the world.

This section highlights the great diversity of the hypothetical average speed of packets. This tells us that the results previously obtained with our technique are valid for a wide range of operational network conditions and are thus representative. This enables us to say that our technique can deliver good results in a real-world scenario.

In [1], we used our dataset to test an RTT-based geo-localization algorithm between our GATEWAYs and servers. To evaluate which type of geo-localization algorithm to use, we performed a study of the average speed of packets between the GATEWAYs of all studied providers and our servers. The average speed is defined as the ratio between space and time. The space is the distance between each GATEWAY and a server. The location of each GATEWAY is acquired thanks to the MaxMind GeoLite2 database [10]. We calculated the distance between each GATEWAY and a server thanks to the Haversive distance, which approximates the distance on Earth between two points given their coordinates. We obtained the time to go from a GATEWAY to a server by dividing by half the RTT_{TCP} recorded in the corresponding connection.

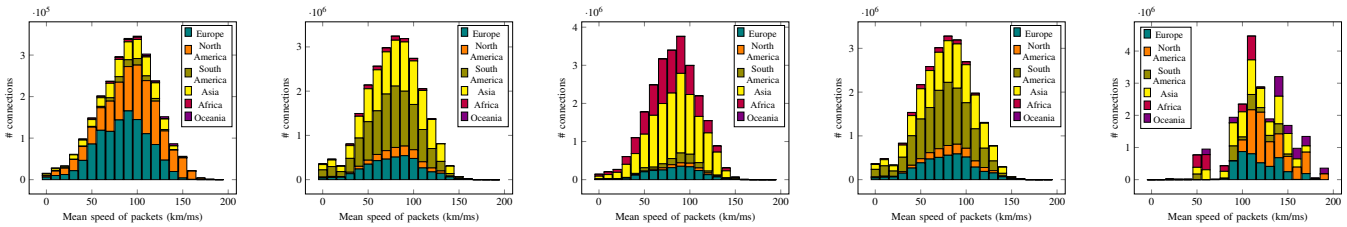
We propose here a study with the same methodology and we perform a new analysis. We analyze individually the connections of each RESIP provider (Ri). For each provider, we consider all the RTT_{TCP} between each server and each GATEWAY that sent requests to it. Moreover, we perform the same analysis for the direct connections between our clients and servers. We exclude the combinations of client and server in which the machines are in the same location. We cannot calculate the distance between them since they share the same coordinates. We use the Haversine formula to obtain the distance between each client and each server. We collect all the direct connections among each couple of machines and we calculate the time as half of the RTT_{TCP} of the connection.

Fig. 2 shows the histograms of the obtained results. On the x-axis, we see the average speed of packets in km/ms. On the y-axis, we find the number of connections showing that speed in the dataset. The average speeds are shown in different colors depending on the continent in which the sender is located. In this Figure, we do not consider the cases where the value of the speed is higher than 200km/ms, for better visualization, or where the continent is not available in the database. These two cases together correspond to 0.08% of all the considered connections and thus we consider their contribution negligible.

We can see how the distribution of the speed for each RESIP provider (Fig. 2(a)-2(d)) ranges from 0 to 150km/ms. Moreover, we see that the distribution for the connections from GATEWAYs of each continent follows a shape comparable to the global one. These curves highlight the great variability of the average speed of packets in our monitored connections. Our dataset is clearly representative of the real conditions of the Internet in which multiple factors affect the time it takes for a packet to reach its destination.

Fig. 2(e) illustrates the results for the direct connections. We can see that the mean speed value ranges between different values as in the previous cases. However, the values in the range are higher in this second study. Connections coming from machines in North America only present values above 80km/ms. Connections from European locations have mostly values between 75 and 150 km/ms. The mean speed values have more variability for the connections coming from other continents. This analysis tells us that the connections between our client and servers have better connectivity levels than the ones between GATEWAYs and servers. We expected these results since the machines are located in well-connected data centers, as opposed to GATEWAYs machines, which are in unknown conditions. However, even in this case, the mean speeds do not end up with a single value. This enables us to say that our setup had no apriori bias that could compromise our experiment and thus the analysis performed on it are representative of a real-world scenario.

[1]We assume the processing time at the client to be negligible with respect to the transmission time.



(a) R1 (b) R2 (c) R3 (d) R4 (e) Direct Connections

Fig. 2: Distribution of the mean speed of packets for each RESIP provider and direct connections.

TABLE I: Connections where there is machine proximity.

Provider	$\delta_{RTT} > 50$ms	Total Connections
R1	64.79%	22,582
R2	86.92%	12,887
R3	62.03%	79
R4	87.08%	14,314
Total	76.90%	49,862

IV. Machines Proximity Analysis

In our detection method, we assume that the RTT_{TLS} is higher than the RTT_{TCP} in case of RESIP connection because the TCP packets sent by the server stop at the GATEWAY while the TLS ones, after reaching this machine, traverse the RESIP infrastructure and are then forwarded to the client.

However, we can consider a scenario in which client, server, GATEWAY and SUPERPROXY machines are in proximity to each other. In this situation, the overhead given by the physical distance of the machines taking part in the connection is small. Naturally, we could think that this influences the values of δ_{RTT} and thus our detection.

We have analyzed the distribution of the δ_{RTT} of such connections. For each connection, we collect the GATEWAY and the SUPERPROXY IP addresses. Thanks to the MaxMind GeoLite2 database [10], we retrieve the latitude and longitude of each of these IPs. We study those connections where client and server are in the same location and in which GATEWAY and SUPERPROXY are not further than 1000km (i.e. 10ms apart at a speed of 100km/s) from it and from each other, considering their Haversine distances.

Table. I shows the results of our analysis. We can see that the total number of connections satisfying our requirements is low, especially for R3. In total, they account for 49,862 connections, which is 0.07% of the total amount of proxied connections. This information tells us that, even when clients and servers are close to each other, it is not common that the SUPERPROXY is close to that location and/or the assigned GATEWAY is in near proximity to the other machines.

We can see that 76.90% of the considered connections have a δ_{RTT} higher than our chosen threshold (50ms). This tells us that, even in an unlikely event where all the machines are in near proximity, our technique works relatively well. The exchanges between two additional machines (GATEWAY and SUPERPROXY) increase the δ_{RTT} enough to detect the presence of a RESIP in more than 3 out 4 cases.

The number of false negatives increases with respect to the global data (23.09% vs 1.93%). However, only 3.07% of these values present a δ_{RTT} lower than 20ms. This is the value under which we can find 97% of the δ_{RTT} of the direct connections. Hence, this shows that there is still a significant difference between direct and proxied connections.

V. Future Works and Conclusion

The results presented in this work, confirm that our detection technique can work in any real-world environment and is accurate enough even in the unlikely event where client, SUPERPROXY, GATEWAY and server are all in close proximity.

Thanks to the positive results obtained with our technique, we convinced one of the leading technology companies for the travel industry to implement our technique in front of their domains victims of scraping. Early results show that the δ_{RTT} is a strong parameter that analysts can use in detecting when a connection passes through a RESIP. In two representative months, it was used as a parameter in 74,32% of the investigations. We are studying these connections flagged with these parameters to assess the impact of our detection.

Furthermore, RESIP are just instruments in the hands of scrapers. These are the real actors that we wish to block. Our intuition is to use the δ_{RTT} to obtain their geolocalization. Indeed, the δ_{RTT} gives information about the "distance" between client and GATEWAY. If the same client uses multiple GATEWAYs, we can find the intersection of the circles whose centers are the GATEWAY locations and whose radii are half of the δ_{RTT} multiplied by the packet. This intersection is the location of the scraper initiating the connection. If scraping campaigns starting from distinct clients take place at the same time, there are multiple intersections and we divide the dataset to geolocalize the machine behind each campaign.

However, there are challenges in achieving our goal. As seen in Section III, the packet speed has no average value. Furthermore, the current geolocalization algorithms are not able to correctly put into practice our theoretical idea [1]. We are thus working to implement a new algorithm that overcomes previous limitations and enables us to localize the client behind the RESIP.

REFERENCES

[1] M. Champion, M. Dacier, and E. Chiapponi. Immune: Improved multilateration in noisy environments. In *2022 IEEE 11th International Conference on Cloud Networking (CloudNet)*, pages 1–6, 2022.

[2] E. Chiapponi, M. Dacier, and O. Thonnard. Inside residential ip proxies: Lessons learned from large measurement campaigns. In *8th International Workshop on Traffic Measurements for Cybersecurity (WTMC 2023), IEEE European Symposium on Security and Privacy Workshops*, 2023.

[3] E. Chiapponi, M. Dacier, O. Thonnard, M. Fangar, M. Mattsson, and V. Rigal. An industrial perspective on web scraping characteristics and open issues. In *2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume (DSN-S)*, pages 5–8, 2022.

[4] E. Chiapponi, M. Dacier, O. Thonnard, M. Fangar, and V. Rigal. BADPASS: Bots Taking ADvantage of Proxy as a Service. In *Information Security Practice and Experience: 17th International Conference (ISPEC 2022)*, page 327–344, Berlin, Heidelberg, 2022. Springer-Verlag.

[5] DataDome. Bot IP addresses: 1/3 of bad bots use residential IPs. Here is how to stop them., 2022.

[6] A. Hanzawa and H. Kikuchi. Analysis on Malicious Residential Hosts Activities Exploited by Residential IP Proxy Services. In *Information Security Applications*, pages 349–361. Springer International Publishing, 2020.

[7] Imperva. 2022 Bad Bot Report, Evasive Bots Drive Online Fraud. Technical report, Imperva, 2022.

[8] B. Krebs. The rise of "bulletproof" residential networks, 2019. https://krebsonsecurity.com/2019/08/the-rise-of-bulletproof-residential-networks/.

[9] F. M, P. Plante, and G. Joly. Illegitimate residential proxy services: the case of 911.re and its iocs, 2022. https://gric.recherche.usherbrooke.ca/rpaas/.

[10] MaxMind. Geolite2 free geolocation data. https://dev.maxmind.com/geoip/geolite2-free-geolocation-data.

[11] X. Mi, X. Feng, X. Liao, B. Liu, X. Wang, F. Qian, Z. Li, S. Alrwais, L. Sun, and Y. Liu. Resident evil: Understanding residential ip proxy as a dark service. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 1185–1201, 2019.

[12] X. Mi, S. Tang, Z. Li, X. Liao, F. Qian, and X. Wang. Your Phone is My Proxy: Detecting and Understanding Mobile Proxy Networks. In *Proc. of NDSS 2021*, 2021.

[13] M. Ryuichi, K. Takumi, F. Hikari, and K. Hiroaki. Investigating potential malicious activities via residential ip proxy services. *Research Report Computer Security (CSEC)*, 2023-CSEC-100(59):1–8, Feb. 2023.

[14] A. Tosun, M. De Donno, N. Dragoni, and X. Fafoutis. RESIP Host Detection: Identification of Malicious Residential IP Proxy Flows. In *2021 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–6, 2021.

[15] A. Vastel. How to Use Machine Learning to Detect Residential Proxies, 2022.

[16] C. Watson and T. Zaw. *OWASP Automated Threat Handbook - Web Applications*. OWASP Foundation, USA, 2018.

[17] Z. Weinberg, S. Cho, N. Christin, V. Sekar, and P. Gill. How to catch when proxies lie: Verifying the physical locations of network proxies with active geolocation. In *Proceedings of the Internet Measurement Conference 2018*, pages 203–217, 2018.

[18] M. Yang, Y. Yu, X. Mi, S. Tang, Y. Guo, S. Li, X. Zheng, and H. Duan. An Extensive Study of Residential Proxies in China. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022.