Botnet Sizes: When Maths Meet Myths

Elisa Chiapponi[1(✉)], Marc Dacier[1], Massimiliano Todisco[1], Onur Catakoglu[2], and Olivier Thonnard[2]

[1] Eurecom, Biot, France
{elisa.chiapponi,marc.dacier,massimiliano.todisco}@eurecom.fr
[2] Amadeus IT Group, Biot, France
{onur.catakoglu,olivier.thonnard}@amadeus.com

Abstract. This paper proposes a method and empirical pieces of evidence to investigate the claim commonly made that proxy services used by web scraping bots have millions of residential IPs at their disposal. Using a real-world setup, we have had access to the logs of close to 20 heavily targeted websites and have carried out an experiment over a two months period. Based on the gathered empirical pieces of evidence, we propose mathematical models that indicate that the amount of IPs is likely 2 to 3 orders of magnitude smaller than the one claimed. This finding suggests that an IP reputation-based blocking strategy could be effective, contrary to what operators of these websites think today.

1 Introduction

This work has been realised in close collaboration with a major IT provider for the airline industry which hosts several dozens of airline websites. These sites are protected by one of the leading commercial anti-bots services, placed in front of them. This service checks the origin and the fingerprints associated with each request against a large number of "signatures"[1].

Bots have been a plague for the Internet for more than 20 years. Early warnings date back to the 2000s with the early DDoS attacks against major websites [3]. Since then, they have continuously evolved from relatively rudimentary pieces of software to very sophisticated components such as the numerous "all in one sneaker bots" (e.g., aiobot.com) that automate the buying process of luxury goods in high demands. To increase their resilience, the bots take advantage of proxy services publicly available on the web, for a fee. Thanks to these services, the bots use temporarily IP addresses that are owned and used by legit users. There are, supposedly, tens of millions of such IPs made available to bots. Would the targeted websites decide to block each IP which is considered to behave like a bot, they would quickly deny access to millions of IPs, some of them belonging to potential customers. Clearly, an IP blocking solution does not appear to be a viable approach due to the, supposedly, sheer volume of IPs, available all over the world.

[1] This is a simplistic explanation. We refer the interested reader to [23] for more information on such existing commercial offerings.

H. Hacid et al. (Eds.): ICSOC 2020 Workshops, LNCS 12632, pp. 596–611, 2021.
https://doi.org/10.1007/978-3-030-76352-7_52

In this paper, we use empirical evidence to investigate the conjecture that such IP blocking strategy will always fail. We reach the conclusion that the situation might not be as bleak as it might seem.

In order to present our findings, the paper is structured as follows. In Sect. 2, we outline the problem faced and our contributions. Section 3 presents the state of the art on web scraping bots prevention. Section 4 describes the experimental setup and the data it produced. Section 5 briefly describes the raw results obtained over a period of 56 days. Section 6 assesses the credibility associated with the belief that these botnets have millions of IP addresses at their disposal. Mathematical analysis confronted with the empirical pieces of evidence leads us to adjudicate against that belief. In Sect. 7, we gather additional information about the IPs observed in order to consolidate the ideas developed in Sect. 6. In Sect. 8, we discuss the lessons learned thanks to our experiment and analysis. A conclusion as well as thoughts for future work are offered in Sect. 9.

2 Problem Definition and Contributions

A 2019 Imperva report [6] describes how the airlines industry is heavily impacted by large armies of bots. In 2017, according to that report, the proportion of bad bots traffic to airline websites was 43.9%. Almost all these bots are used to gather free information from the airlines' sites about flights and ticket prices. It is commonly agreed that the actors behind these bots activities are unauthorized business intelligence companies, online travel agencies and data aggregators. Indeed, a large part of their business relies on web scraping and using bots instead of having a paying agreement with the targeted websites is much more profitable for them. They harness information, increasing dramatically the amount of requests to be served by airlines websites. Responding to these requests, due to the price ticketing process, is an expensive task which well-behaving organisations normally pay for. The bots aim at getting the same service *for free*. By doing so, they misuse the service provided by airlines companies to individual users.

An arms race exists between bot makers and anti-bot providers. The bot detection relies on a number of different fingerprinting techniques to recognize malicious agents [23]. As soon as a family of bots is identified and blocked, their bot masters replace them with new ones. Blocking all the IP addresses of identified bots is usually not seen as a viable option because it is well known that the real IP addresses of the bots remain hidden behind a large amount of proxy IP addresses provided by professional services. These services claim to offer to their customers millions of residential IP addresses, leaving any IP blocking solution doomed to potentially block a large amount of legit customers.

Quoting one of these websites [12], we see that they offer to their customers to "use [their] rotating residential proxies comprised of real user devices, making them undetectable when used correctly". The owners of these real devices, also called exit nodes, "[...] agreed to route [...] traffic through their hosts in exchange for free service" [12]. A quick search on the Internet returns more than a dozen similar proxy service offerings. We prefer not to offer them some additional advertisement by listing them all here. Suffice it to say that, for instance,

both [12,15] claim to offer more than 70 millions of IP addresses whereas [20] supposedly has more than 10 millions IPs. Others have similar claims.

The benefits of hiding behind this very large pool of IP addresses is threefold for the web scraping actors: first, linking a scraping campaign to any known organisation is impossible, thus no attribution and legal recourse; second, the impressive number of frequently changing IP addresses used renders any IP blocking strategy impractical; third, they can run these campaigns with a very limited amount of powerful machines on their back end without the need of any vast and highly distributed infrastructure.

In [2], we describe an experiment designed to analyse the behavior of these bots. That experiment did confirm the existence of these advanced persistent bots (APBs) and the proxies they were relying on. It also raised questions regarding the real amount of IPs put at the disposal of the bots. In this work, we carry out an in-depth investigation of that question. By doing so, we provide the following contributions:

- We provide additional empirical pieces of evidence of the existence of very stealthy APBs and confirm the usage of proxy servers by these bots
- Using two distinct approaches, we show that i) IP addresses provided to the bots are not randomly assigned and that ii) the pool of IPs they are taken from is *two* to *three* orders of magnitude smaller than what is announced by the proxy websites.
- We explain how the idea of IP-blocking could be rejuvenated to defeat such sophisticated bots.

3 State of the Art

Botnets, collections of hosts controlled by a bot, have been used for the years for nefarious activities, such as scraping web pages of different industries [5]. Applying IP reputation to mitigate the threats of web scraping is not a new idea [4]. Moreover, this technique has already been largely used against spam bots [10].

However, as shown in the Imperva Report 2020 [5], recent years have witnessed the rise of traffic produced by Advanced Persistent Bots (APBs). These bots produce few requests per IP staying below the rate limits and protecting their reputation. They rely on professional proxy services that make large numbers of IP addresses available for these activities [19]. These services claim to have access to tens of millions of residential IPs and to be able to rotate them among the different requests of each client. For these reasons, the report [5] asserts that IP blacklisting has become "wholly ineffective". Doubtlessly, millions of different IP cannot be blacklisted all together and e-commerce websites cannot risk to block requests coming from real customers.

In 2019, Mi et al. [13] proposed the first comprehensive study of Residential IP Proxy as a Service. Even if their methodology has been partially criticized for the fingerprinting process of the devices [16,17], they created a successful infiltration

framework that enabled them to study residential proxy services from the inside. They collected 6.18 milions of IPs, of which 95,22% are believed to be residential. Among their findings, it is peculiar to see a discrepancy between the number of IPs claimed by Luminati [12] (30 millions) and the ones collected by them for the same provider (4 millions using 16 million probings). The authors provide no clear explanation for this gap. Furthermore it is noteworthy to mention the discovery of two providers using the same pool of IPs, while another one built its network on top of Luminati [12]. Our paper also aims at better understanding the residential IP proxies ecosystem by providing a different view point.

Nowadays, web site owners usually take advantage of third party anti-bot services to perform bot management. These commercial solutions analyse the incoming requests to the websites. As described in [23], multiple parameters are collected from the environment in which the request is generated, thanks to fingerprinting. This set of parameters can be used to identify the same actor who launches different requests, potentially from different IP addresses. If a signature is recognized as coming from a bot, the corresponding traffic can be blocked or other mitigation actions can be put in place.

Azad et al. [1], propose an empirical analysis of some anti bot services. Unfortunately, their findings indicate that these solutions are mostly efficient against basic bots but not against the truly sophisticated ones. Indeed, an arms race is taking place between anti bot services trying to fingerprint and bots trying to circumvent the detection. This has led the actors behind the bots to perform only small amounts of requests per IP, with the goal of remaining undetected.

4 Experimental Setup

In [2], we have described a honeypot-based experimental setup designed to analyze the behavior of web scraping bots. We offer in this Section a brief recap of that experiment as it is the source of the data we will be analysing in the rest of the paper. The interested reader is referred to [2] for more detailed information.

This experiment was run in close collaboration with a major IT provider for airlines websites. This party handles the calculation of the fares and the booking process for multiple airlines. The airlines companies pay the IT provider an amount proportional to the transactions served. Naturally, an excess of bot traffic dramatically increases the volume of transactions and thus the infrastructure costs for both the airlines and the IT provider. In recent years, bots started to perform an intensive price scraping activity towards airline's websites, producing up to 90% of the requests on some domains [6].

To mitigate this phenomenon, the IT provider with which we collaborate, is using a commercial bot detection service provider. A box is put in front of the provider's booking domains and it detects bots thanks to browser fingerprinting and machine learning. Every request is studied and a signature is assigned to it. If the signature matches the one of a bot, an action is taken such as blocking, serving a CAPTCHA [24] or a JavaScript challenge. However, sophisticated bots, dubbed APBs for Advanced Persistent Bots [5], can overcome these countermeasures and/or change their parameters to avoid detection [14].

This solution works but has a major drawback, which is to provide feedback to the bots when they are identified. They use this information to morph as soon as they detect that they have been unmasked. By doing so, they defeat the mitigation process provided by the anti-bot solution. To overcome this problem, we have decided to create a new action associated with a signature match: requests coming from identified bots would now be redirected to a real-looking, yet fake, web page. This web page, which can be seen as an application layer honeypot, serves two distinct objectives: i) reduce the workload of the production servers, ii) study the behaviour of the bots.

This honeypot is able to produce responses that are, syntactically, indistinguishable from the real ones. However, semantically, they differ because we use cached values or, sometimes, modified values for the tickets. Cached values dramatically reduce the cost of computing the responses. Modified values enable us to analyse to what extent the bots are capable of detecting erroneous information provided to them.

We have designed and implemented such a platform in collaboration with the IT provider and a specific airline company. We chose a company whose traffic was highly impacted by bots. At the time of our experiment, that company was receiving, on average, 1 million requests per day, of which 40% were detected as bot traffic by the anti bot solution. Unfortunately, the anti-bot solution is not capable of blocking all bots. Each signature is associated with a confidence level indicating the uncertainty whether the request comes, or not, from an ill behaving bot. Depending on that value, that IP will be blocked, challenged (e.g. with a CAPTCHA) or simply put under scrutiny (e.g. to be blocked later if it sends a suspiciously large number of requests). We focused on that last category and found, for that airline, a signature that was matched every day, always in the same small time window of 40 min by, roughly, the same amount of IPs. Last but not least, almost none of these IPs ever booked a ticket. All these elements gave us great confidence that that signature, while not blocked by the anti bot solution, was reliably identifying members of a specific botnet. We have then configured the anti-bot solution to redirect all requests matching that signature to our honeypot.

For this publication to be self-contained, we offer in the next Section a synthetic presentation of the raw results obtained and some statistical results.

5 Experimental Results

The experiment ran for 56 days, between 7th January and 2nd March 2020. We have had no match for our signature after that date. We believe the reason has to do with the business needs of the actor behind these bots. Indeed, that date coincides with the beginning of the worldwide pandemic. Furthermore, the airline, subject of our experiment, is the main one for a country whose government issued its first major travel restriction on the 2nd of March, practically shutting down airline travel to and from that country. Without any customer interested in buying tickets to/from that country, there was no incentive for the

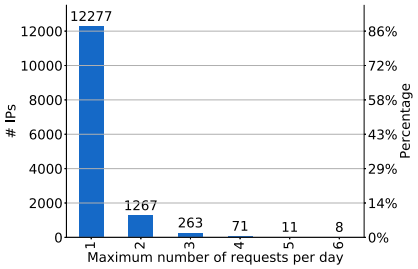

Fig. 1. Left (resp. right) Y axis: absolute (resp. relative) amount of IP addresses which have made at most X requests per day

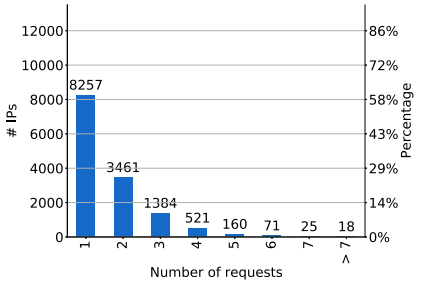

Fig. 2. Left (resp. right) Y axis: absolute (resp. relative) amount of IP addresses which have made at most a grand total of X requests during the whole period of the experiment.

malicious actor to keep collecting ticket prices from that company. This most likely explains the disappearance of these bots.

Over the duration of the experiment, the honeypot has received 22,991 requests. The daily average amount was 410 with a standard deviation of 33 queries. All requests arrived at the same time of the day. The signatures were only seen during a small time window of 38.18 min, on average. The amount and the timing of the requests were in line with those of that bot signature before the beginning of the experiment.

The 22,991 requests were issued by 13,897 unique IPs. Figure 1 shows that most of the IPs (97% of the total) made at most two requests per day, with the vast majority (88%) making only one request per day. Figure 2 shows the total amount of requests made per distinct IP over the whole experiment. Here, we see that 8,257 IPs have sent only one request. This value is to be compared with 12,277 of Fig. 1. It highlights the fact that a large amount of IPs have shown up on at least two different days, issuing a single request every time. This is confirmed by Fig. 3 where we see that almost 30% of the IPs have been seen on at least two different days.

That number is surprisingly high. Indeed, at this stage, we have to remind the reader that these IPs are proxy IPs and that the actual client machine sending a request is hidden behind. The proxy service offers a pool of addresses to be given to these clients. Let us call P the size of that pool. Figure 3 shows how many times a given address has been picked over a period of 56 days. The fact that there are 2,801 that have been used twice over that period is inconsistent with the assumption that the addresses would be randomly picked out of a very large pool of millions of IPs. Indeed, to calculate the probability that a given IP got picked twice over this period comes down to resolving the classical birthday paradox which can be generalized as follows:

> *Given n random integers drawn from a discrete uniform distribution with range [1,d], what is the probability p(n; d) that at least two numbers are the same? (d = 365 gives the usual birthday problem.)* [22]

In our case, n is equal to 56, the number of days where IPs from the pool are assigned to clients and d is equal to the size of the pool P. We want to assess the probability that the same IP would be drawn twice over that period of 56 days. We can rephrase the birthday problem for our needs as follows:

> *Given 56 random integers drawn from a discrete uniform distribution with range [1,P], what is the probability* $p(56; P)$ *that at least two numbers are the same?*

The formula $1 - (\frac{P-1}{P})^{\frac{56(55-1)}{2}}$ gives an approximate result:

- If $P = 10000000$ then $p(56, 10M) \approx 0.000154$
- If $P = 1000000$ then $p(56, 1M) \approx 0.001538$
- If $P = 100000$ then $p(56, 100K) \approx 0.015282$

Clearly, considering that we have seen more than 30% of the IPs drawn at least twice, either P is significantly lower than the number announced by the proxy services, or the assignment of IPs is not randomly done, or both.

Regarding the total amount of IPs, we saw only 13,897 different ones. Every day the number of distinct IPs, on average 371 (shown in yellow in Fig. 4), was similar to the number of requests, on average 410. Thus, it is clear that most IPs send a single request and reappear some time later. In the same figure, the green columns represent the cumulative number of unique IPs observed in our honeypot since the beginning of the experiment. The figure shows that the daily increment decreases over time, suggesting that it will eventually reach a maximum.

To better characterize and understand the threats ecosystem we are facing, we try to find a mathematical model that approximates as closely as possible the assignment of IPs made by the proxy provider. We use that model to derive the most likely size of P. This is done in the next Section.

6 Modeling Results

6.1 Introduction

We propose two distinct modeling approaches to assess the most likely size of the pool of IPs P put at the disposal of the stealthy APBs we have observed. Both models deliver a value which is below 70K, i.e. three orders of magnitude less than the 70M IPs supposedly provided by [12].

In the first approach (subsection 6.2), we look at the IPs assigned every day by the proxy to the bots. We model this as a drawing process made in a pool of size P and we try to find the best probability distribution function that would

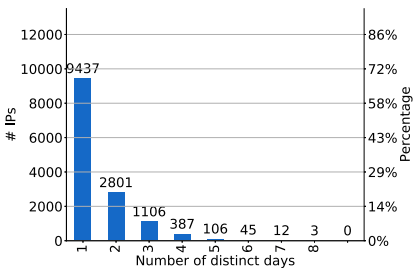

Fig. 3. Left (resp. right) Y axis: absolute (resp. relative) amount of IP addresses which were seen in X distinct days during the whole experiment.

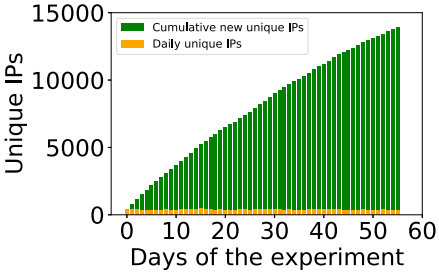

Fig. 4. Cumulative curve of the new unique IPs in comparison with the daily unique IPs.

produce similar results to the ones we have witnessed. From there, we derive the value of P.

In the second approach (Subsect. 6.3), we look for a fitting curve to approximate the one shown in Fig. 4 and, by extrapolating it, see what maximum value it would reach, and when.

6.2 IP Assignment as a Drawing Process

General Principle. Figure 3 tells us how many IPs have been assigned to a bot only once, or twice, or three times ... over the duration of the experiment. We model this assignment process by a daily probabilistic drawing process without replacement. We arbitrarily define a pool size $\mathcal{P}$. On a given day, we draw from our pool, without replacement, a number of values equal to the amount of distinct IPs seen that day. We do this every day, keeping track of which value got drawn several times during this exercise. We use these accumulated results to produce a histogram similar to Fig. 3.

We use the Wasserstein[2] distance to assess the similarity between this histogram and the one from Fig. 3, making the reasonable assumption that the values of the produced histogram are derived from the real one by small and non-uniform perturbations.

We have no reason to believe that the drawing is done every day instead of every 2 days or 3 days or more. We thus repeat the process with other window sizes s (2 to 10), but we proceed with a drawing with replacement. We impose an additional constraint though. A given value cannot be drawn more than s times, i.e., once per day. Once a value has been drawn s times, it is not replaced in the pool anymore.

[2] This distance is known as the earth mover's distance, since it can be seen as the minimum amount of "work" required to transform one histogram into another, where "work" is measured as the amount of distribution weight that must be moved, multiplied by the distance it has to be moved [11].

We use different probability distribution functions to ensure that they do not produce drastically different sizes. Various other functions could have been used. Our goal is not to find the best one but to show that several "good enough" ones deliver the same ballpark figure for P.

Algorithm Used. We studied the distribution of the IPs for subgroups of days of size s ranging from 1 to 10. To group the days, we have used juxtaposed windows (as opposed to sliding windows) to ensure that our final histogram contained the same amount of values as the one in Fig. 3. We chose juxtaposed windows to not count twice the IPs of a singular day and reproduce thus a coherent replica of the observed data.

We have run simulations with different population sizes $\mathcal{P}$. We have incremented $\mathcal{P}$ by 10,000, starting with the initial value of 10,000 up to 100,000. Moreover, we have tested values from 100,000 to 200,000 with an increment of 20,000. For a given time window, we have produced as many histograms as distinct population sizes. Each histogram is obtained thanks to 100 simulations. Each simulation produces its own histogram and we compute the Wasserstein distance between this histogram and the empirical one. An average Wasserstein distance value is then obtained from these 100 simulations. The lowest value of this average distance corresponds to the size P which best represents the observed data. For each window size, for each population size, for each simulation, we have plotted the distances obtained using a boxplot representation. This algorithm has been applied using three distinct probability distribution functions, as explained here below.

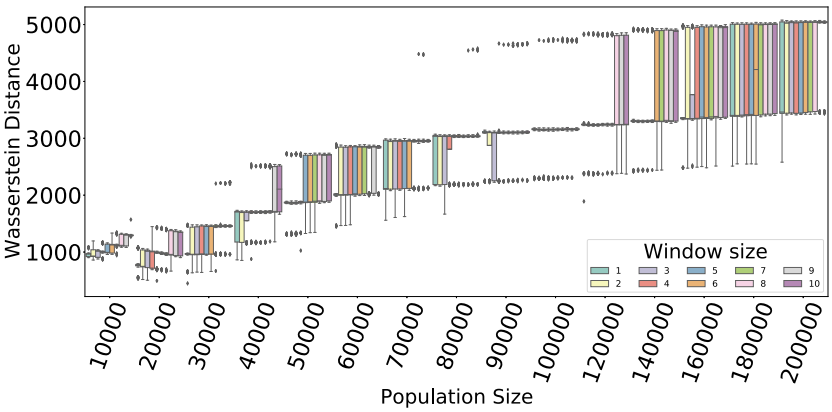

Fig. 5. Uniform distribution: for each population size on the x axis, a group of boxplots displays the Wasserstein Distances (y axis) obtained in the 100 experiment for that population size. Each color represents the window size used for the simulation.

Uniform Distribution. The simplest model is the one where all IPs, every day, have the same probability of being assigned to a bot. To model this, we use a uniform distribution as the probability distribution function in our drawing process. Figure 5 shows for each window size (colored legend) and for each population size (X-axis), the boxplots of Wasserstein Distances (Y-axis) obtained in all the experiments. We clearly see that for $\mathcal{P}$ bigger than 30K, the bigger its value, the more different is the obtained histogram with respect to Fig. 3. The best distances are obtained for the low value of $\mathcal{P}$ of 20K, for all time window sizes. The Wasserstein distance is quite high though, around 1,000 and we have looked for other distributions with the hope of obtaining smaller distances.

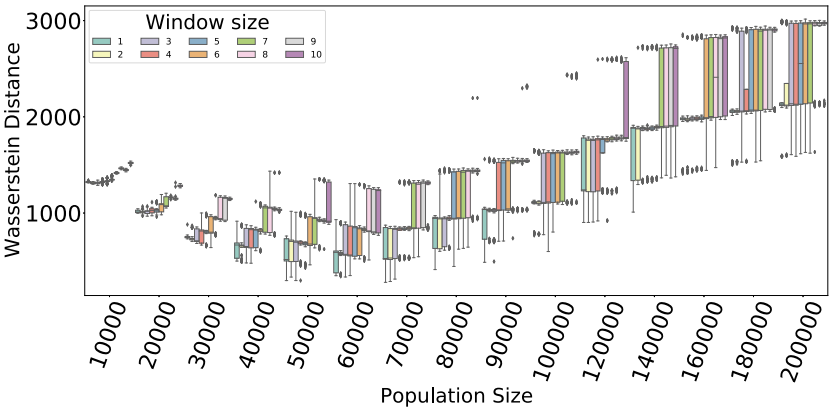

Fig. 6. Gaussian distribution: for each population size on the x axis, a group of boxplots displays the Wasserstein Distances (y axis) obtained in the 100 experiment for that population size. Each color represents the window size used for the simulation.

Gaussian Distribution (aka normal). It is reasonable to imagine the existence of a bias in the IP assignment process that would lead some IPs to be more frequently used whereas others would be rarely picked. This could be due, for instance, to the simple fact that some residential IPs might be more frequently available (online) than others. Another reason could be that proxies, to ensure a better quality of service, assign preferably IPs "close" to their customers. Our goal is not to identify the causes of these biases but, simply, to assume that they could exist and, thus, model this possibility. To do so, we have run our algorithm with a Gaussian distribution. For the sake of concision, the results presented here correspond to the parameters mu=0.5 and $sigma$=0.1. Other choices lead to the same lessons learned and this combination offers the best distances. We offer in Fig. 6 a similar representation as in Fig. 5. This model seems to be a better approximation since the best Wasserstein distance is now half of the one obtained for the uniform distribution. As expected, the size $\mathcal{P}$ does grow since a number of IPs are now very rarely chosen. Its value, around 60K, is still three orders of magnitude below the claimed 70M.

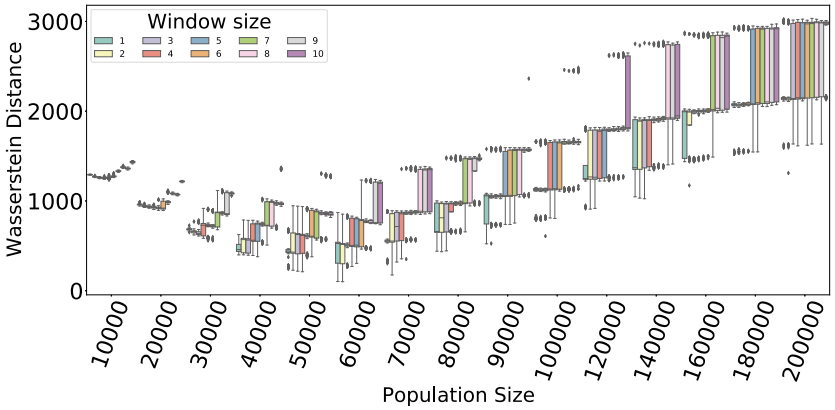

Fig. 7. Beta distribution: for each population size on the x axis, a group of boxplots displays the Wasserstein Distances (y axis) obtained in the 100 experiment for that population size. Each color represents the window size used for the simulation.

Beta Distribution. Last but not least, we present also the results obtained with the Beta distribution, with $alpha = 1$ and $beta = 5$; which enables us to represent a different form of bias in the choice but the results are very consistent with an optimal size P of 60K and a Wasserstein distance below 500. The results are represented in Fig. 7.

6.3 Fitting Curve

As explained before, a distinct approach to get an informed estimate of the size P consists in starting from the values observed in Fig. 4, in finding a fitting function and in extrapolating its values.

To do so, after having looked at the data at our disposal, we have observed that, roughly speaking, the amount of new IPs (i.e., never seen so far) observed on a daily basis was decreasing linearly over time. We were thus hoping to be able to find a good fitting function [18], thanks to an exponentially decaying one. We found out by means of simulations that the best fit was achieved with the following function:

$$a * (1 - e^{-(x-b)/c}) \tag{1}$$

The parameters that provide the best fit are:

$$a = 2.77313369e + 04$$
$$b = -4.77879543e - 01$$
$$c = 8.04885708e + 01$$

The fitted curve is represented in Fig. 8. To assess their similarities, we calculate the Pearson correlation factor [21] and obtain the value 1.000 which indicates

a total positive linear correlation, confirming the adequacy of our fitting function which is visible by the quasi superposition of both curves in Fig. 8

We can now use that fitting function to extrapolate the total amount of distinct IPs we would have seen, had we been able to run the experiment for 3 years. Figure 9 shows how the curve reaches a plateau after a bit more than a year. Thus, according to this distinct approach, the bots we have observed only have a couple of tens of thousands of IPs at their disposal, a value which is consistent with the ones found with the first approach.

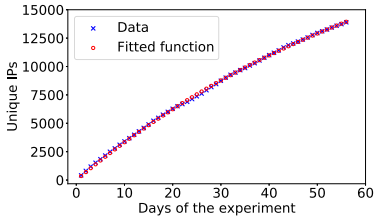

Fig. 8. Projection of the real data on the fitting curve values

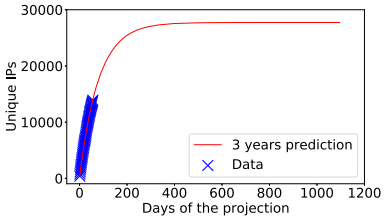

Fig. 9. 3 years prediction of the number of different IPs that would have been seen in the honeypot

7 Complementary Results

In this Section, we present additional pieces of evidence to those already provided in [2], which confirm that the IPs we have analysed are, indeed, quite likely provided by proxy services.

These IPs are supposed to be residential IPs; i.e., they belong to legit users who could, possibly, be interested in buying tickets. To verify this, we have looked for the presence of these IPs in the logs of 17 other airlines. We found out that during the experiment, five bookings have been realised by 5 of our IPs. In Table 1 we indicate when the booking was done vs. when the same IP was seen in our honeypot logs. As expected, the dates differ greatly. Moreover, none of these requests had the bot signature associated with them. They look perfectly legit. This confirms two things i) some of these IPs are likely used by legit users, ii) the risk of blocking legit customers when blocking identified proxy IPs remains extremely small.

On the other hand, the simplest way to implement a proxy is to open some ports and have a proxy server listening behind it. This should thus be detectable by the various actors who scan the Internet continuously, looking for threats and, or, vulnerabilities. We have used two such systems to see if they had identified our IPs as behaving like proxies. First, we have used IPInfo.io [7] which provides a boolean value for each IP in the categories “VPN”, “Proxy”, “Hosting”. According to the provider of that service, VPNs are usually encrypted traffic endpoints so typically, if there is a VPN operating on an IP address, there will

Table 1. Timestamp of the bookings and the honeypot requests made by the same IPs.

Booking time	Request time
2020-01-17	2020-02-01
	2020-02-05
	2020-02-14
2020-02-26	2020-01-10
	2020-01-23
2020-02-29	2020-02-01
2020-02-06	2020-02-23
2020-02-07	2020-01-24
	2020-02-02
	2020-02-19

Table 2. Distribution of the fraud score of IPQualityScore

Score (S)	% of IPs	# of IPs
$S < 75$	28%	3958
$S \in [75, 85]$	46%	6371
$S \geqslant 85$	26%	3568

be either encrypted traffic or ports open which will obviously show a VPN is being used. Proxies are usually just a "HTTP forwarding" service and redirect traffic to somewhere else (internal domains, other servers, etc.) [8]."Hosting" category specifies if the IP belongs to hosting providers.

Table 3 shows that a couple of IPs have been categorized as involved in suspicious activities but not as many as expected. However, the results obtained with IPQualityScore [9] are much more aligned with our expectations. As explained in their documentation, this service tells if an IP has been used in "automatic fraudulent behavior" in the "Bot status" category, while indicating a positive value of "Recent Abuse" if the IP has been involved in a "recently verified abuse across their network". The abuse behavior includes charging back, compromised devices, fake app installation. Moreover, the "VPN" category indicates server or data center IPs which allow tunneling. Finally, the "Proxy"[3] category, identifies a device being infected by malware, a user selling bandwidth from their connection or other types of proxy like SOCKS, Elite, Anonymous, Tor, etc. With this service, we can notice that the number of IPs involved in malicious activity is much higher in comparison to the first one. Furthermore, this service provides a general fraud score for the IP: this value ranges form 0 to 100, indicating a suspicious activity when higher than 75 and an high risk when greater than 85. Table 2 tells that around 72% of the IPs show a suspicious behavior, of which 28% are classified as high risk. This is quite consistent with the idea that malicious actors are hiding behind them, ruining the reputation of these IPS.

To dig deeper into the analysis of the malicious behaviors associated with these IPs, we looked for their presence in anti-spam DNS blocklists. Using the Python library Pydnsbl we checked multiple blocklists and we found out that 76% of the IPs were blocked at least in one of them at the time of our analysis

[3] A "VPN" is automatically a "Proxy" according to their definitions.

Table 3. IPInfo.io classification of the IPs

Type	Number of Ips	Percentage
VPN	180	0.013
Proxy	59	0.004
Hosting	1733	0.125

Table 4. IPQualityScore classification of the IPs (*From the total number of positive matches, 10213, we subtracted the number of positive values of VPN)

Type	Number of Ips	Percentage
VPN	9138	0.658
Proxy*	1075	0.077
Recent abuse	3878	0.279
Bot status	2780	0.200

(July 2020). Hence, we had the confirmation that these IPs were doing malicious activity also outside of our environment.

8 Discussion

The whole point of our experiment was to obtain, over a long period of time, a meaningful set of IPs that we could confidently say were behaving as they were members of the very same botnet. The very strong correlation in their activity patterns, detailed in [2], is as close as a ground truth one could hope for. The anti-bot detection solution identifies many more IPs as behaving like bots but our experience in looking at the logs gives us no assurance that IPs flagged with a given signature belong to the same botnet. Indeed, the goal of each signature is to fingerprint *"a bot"*, not *"the bot from botnet X, Y or Z"*. The analysis we have carried out in this paper required a *clean* dataset in order to be able to derive meaningful conclusions. We are very well aware though that, compared to all the bots that are out there, our dataset is relatively small and we do not pretend that our conclusions can, or should, be extended to all botnets that are in activity. Our results do only apply to the botnet we have studied. Having said so, all elements at our disposal, explained in the previous pages, indicate that this botnet is a perfect example of so called APBs, Advanced Persistent Bots, and is thus representative of the many others that are scraping websites. Therefore, we have good reasons to believe that our results could probably be generalized to many other botnets, without having, at the moment the data to support this claim.

If true, this would mean that large websites victims of web scraping bots would see the same IPs coming back regularly and that the grand total of IPs they would have to watch for would remain manageable (in the order of tens of thousands instead of tens of millions). An IP blocking strategy could thus be rejuvenated: seeding their sets of IPs with the ones clearly identified as behaving as bots, that strategy could enable them to catch the most evasive bots when they show up with a known bad IP. Redirecting these IPs to a fake web site instead of blocking them would also enable them to keep watching their behavior and,

possibly, redirect them to the real web site if their requests are not consistent with those of known bots (i.e., in the case of a false positive).

The results presented in this paper have helped us in convincing our partner, the major IT provider, to move forward into building such an environment and the work is under way. We felt it was important to share already now our preliminary results with the community not only in order to let other benefit from the gained insights but also, possibly, to obtain feedback on important elements we could have missed. We do hope our contributions will participate in diminishing the negative impact created by these bots on the global Internet ecosystem.

9 Conclusion

In this paper, we have studied in detail a specific web scraping botnet that is representative of the plague most airline websites are suffering from.

Thanks to two distinct mathematical models, we have shown that the total amount of IPs at the disposal of this botnet was most likely in the low tens of thousands. We have also given pieces of evidence that these IPs were provided by proxy services, thought to be able to provide tens of millions of IPs to their customers. If our finding applies, as we think it does, to other botnets then an IP-blocking strategy could be applied, contrary to the common belief. We encourage others to carry out similar experiments to confirm, or deny, our findings while we are in the process of testing our conjecture in a new large scale experiment.

References

1. Amin Azad, B., Starov, O., Laperdrix, P., Nikiforakis, N.: Web runner 2049: evaluating third-party anti-bot services. In: 17th Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA 2020), Lisboa, Portugal (2020)
2. Chiapponi, E., Catakoglu, O., Thonnard, O., Dacier, M.: HoPLA: a honeypot platform to lure attackers. In: C&ESAR 2020, Computer & Electronics Security Applications Rendez-vous, Deceptive Security Conference, Part of European Cyber Week, Rennes, France (2020). http://www.eurecom.fr/publication/6366
3. Dietrich, S., Long, N., Dittrich, D.: Analyzing distributed denial of service tools: the shaft case. In: Proceedings of the 14th USENIX Conference on System Administration, pp. 329–339. New Orleans, Louisiana, USA (2000)
4. Haque, A., Singh, S.: Anti-scraping application development. In: 2015 International Conference on Advances in Computing, Communications andInformatics (ICACCI), pp. 869–874, Kochi, India (2015)
5. Imperva: Imperva bad bot report (2020). https://www.imperva.com/resources/resource-library/reports/2020-bad-bot-report/
6. Imperva: how bots affect airlines (2019). https://www.imperva.com/resources/reports/How-Bots-Affect-Airlines-.pdf
7. Comprehensive IP address data, IP geolocation API and database - IPinfo.io. https://ipinfo.io/
8. IpInfo.io: personal communication, August 2020
9. Fraud prevention—detect fraud—fraud protection—prevent fraud with IPQS. https://www.ipqualityscore.com/

10. Jung, J., Sit, E.: An empirical study of spam traffic and the use of DNS blacklists. In: Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement, IMC 2004, p. 370–375. Association for Computing Machinery, Taormina, Sicily, Italy (2004)
11. Levina, E., Bickel, P.: The earth mover's distance is the mallows distance: some insights from statistics. In: Proceedings Eighth IEEE International Conference on Computer Vision, ICCV 2001, vol. 2, pp. 251–256, Vancouver, Canada (2001)
12. World's leader in web data collection and proxy for businesses— luminati.io. https://luminati.io/
13. Mi, X., et al.: Resident evil: understanding residential IP proxy as a dark service. In: 2019 IEEE Symposium on Security and Privacy (SP), pp. 1185–1201. San Francisco (2019). https://doi.org/10.1109/SP.2019.00011, ISSN: 2375-1207
14. Motoyama, M., Levchenko, K., Kanich, C., Mccoy, D., Voelker, G., Savage, S.: Re: CAPTCHAs–understanding CAPTCHA-solving services in an economic context. In: Proceedings of the 19th USENIX Security Symposium, pp. 435–462. Washington, DC (2010)
15. Oxylabs: gather data at scale with an innovative proxy service—oxylabs. https://oxylabs.io/
16. Samarasinghe, N., Mannan, M.: Another look at TLS ecosystems in networked devices vs. web servers. Comput. Secur. **80**, 1–13 (2019). https://doi.org/10.1016/j.cose.2018.09.001
17. Samarasinghe, N., Mannan, M.: Towards a global perspective on web tracking. Comput. Secur. **87**, 101569 (2019). https://doi.org/10.1016/j.cose.2019.101569
18. Scipy.optimize curve fit function. https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html
19. Scraper api. https://www.scraperapi.com/blog/the-10-best-rotating-proxy-services-for-web-scraping/
20. The best residential proxy network with 40M+ IPs. https://smartproxy.com/
21. Stigler, S.M.: Francis Galton's account of the invention of correlation. Stati. Sci. **4**(2), 73–79 (1989). http://www.jstor.org/stable/2245329
22. Suzuki, K., Tonien, D., Kurosawa, K., Toyota, K.: Birthday paradox for multi-collisions. In: Rhee, M.S., Lee, B. (eds.) ICISC 2006. LNCS, vol. 4296, pp. 29–40. Springer, Heidelberg (2006). https://doi.org/10.1007/11927587_5
23. Vastel, A., Rudametkin, W., Rouvoy, R., Blanc, X.: FP-Crawlers: studying the resilience of browser fingerprinting to block crawlers. In: Starov, O., Kapravelos, A., Nikiforakis, N. (eds.) MADWeb 2020 - NDSS Workshop on Measurements, Attacks, and Defenses for the Web. San Diego, United States, February 2020. https://doi.org/10.14722/ndss.2020.23xxx
24. von Ahn, L., Blum, M., Hopper, N.J., Langford, J.: CAPTCHA: using hard AI problems for security. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 294–311. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_18