Should network operators hop on the data plane?

Max Resing
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
m.resing-1@student.utwente.nl

ABSTRACT

Services on the internet are continuously targeted by scanners that try to automatically break into a system. To increase their success rate, scanners target specific IP address ranges in which they expect vulnerable hosts. Those scanners can be observed with the help of network telescopes and honeypots. Monitoring the malicious activity reveals the originating IP addresses of a scanning host.
Some service providers have developed systems that automate the task of monitoring the Internet and identifying the origins of malicious scanning activity. Those findings are automatically evaluated to publish blocklists in so called data feeds periodically. Those systems are mostly cloud-based which raises the question if their feeds also find those scanners which do not target networks of cloud infrastructure.
In this paper, we assess a specific data feed provider by setting up honeypots not only in cloud-based environments but also in residential areas and campus networks. The resulting data set provides valuable insights in scanning activity aiming at different kinds of networks.
A geographical and temporal analysis delivers indicators that different scanners target different protocols. Further, the analysis shows that certain scanners target specific networks exclusively. Particularly scanners of residential areas are hard to discover with cloud-only sensing infrastructure. Ultimately, the research supports network operators to estimate the limitations of the data feeds.

1. INTRODUCTION

Over the past decades, the internet has become omnipresent in our daily lives. Today's society heavily depends on the numerous services which are connected to the internet and which provide useful tools to the billions of daily users. These services play a crucial role in our daily life since they affect the way we communicate, work, travel, entertain and even teach ourselves.

The increasing number of users unavoidable leads to an increased amount of cyber criminality. One such activity is IP and port scanning. In this work, scanning activity is defined as an automated program or script which scans an IP range for online hosts. If a malicious scanner identifies an online host, it tries to break in with predefined credentials.

A popular tool to defend against those scanners is IP blocklisting. Blocklists are essentially a collection of IPs that were identified as malicious. A whole industry has emerged from the idea of blocklists [17]. Different service providers try to compete with each other by providing the best lists of scanner IPs. Certain tools, such as network telescopes and honeypots, are used to identify scanner addresses. Scanners apply different strategies to find as many hosts online as possible [24]. Thus, a key parameter to find as many scanner IPs as possible is diversity. Diversity should increase the chance that the infrastructure in use is scanned by as many scanners as possible. One such service has achieved diversity by operating honeypots in cloud infrastructure distributed all over the globe [16].

In this paper, we hypothesise that global diversity is insufficient to identify all scanners. Internet service providers (ISP) apply different configurations to networks depending on the purpose of the network [23]. For instance, residential networks tend to have a dynamic IP assignment, whereas industrial areas obtain static IPs. In return, scanners might improve their rate of success by targeting specific IP ranges or specific types of networks. Some scanners could be optimized on default credentials of consumer software which might be operated mostly in residential area networks.

To study if this is indeed the case, we focus on three kinds of networks: Campus networks, networks in residential areas and cloud infrastructure. We operate multiple honeypots in each type of network. The honeypots record login attempts and thus, provide valuable insights on scanning activity tailored to specific types of network.

We perform a geographical and temporal data analysis to disclose patterns in scanning activity. Ultimately, we compare our data to the data feeds of *dataplane.org* to assess the overlap of scanner IPs. This analysis also reveals whether just a global distribution of honeypots is sufficient to identify a large portion of scanners, or if the diversity of network types also matters. This work helps network operators to decide whether the dataplane.org feeds can improve the defense mechanisms of their own networks. Are these block list recommendations a useful asset for every network, administrated by network operators?

2. BACKGROUND

To understand this paper, it is essential to understand scanning activity on the Internet. A scanner is an agent which targets an IP address space and tries to determine whether specific services are operating on certain ports. After identifying such a service, a malicious scanner attempts to break in by brute-forcing common username and password combinations [15].

35th Twente Student Conference on IT July 2nd, 2021, Enschede, The Netherlands.

Kikuchi and Terada have shown that a large number of unauthorized login request originates from automated attacks [14]. Those scanners can either operate fully autonomously around the clock or start scanning after being activated manually for a certain period of time.

The scanning activity of a single host is easily mitigated with tools like Fail2Ban [13]. To avoid it, some scanners might operate in a distributed manner. In such a case, the scanner is installed on numerous compromised hosts instructed by a command and control server [30, 31]. Such a network of infected devices is called a botnet. Other ways attackers try to avoid ending up on blocklists is by using virtual private networks [27] or the Tor network.

Network operators and researchers continuously monitor activity on the Internet to provide the best defense against these automated attacks [7, 18]. Different technology has been proven to be useful to monitor activity on the internet. Two common tools are network telescopes [28] and honeypots [3]. A network telescope is an IP subnet without any hosts operating in them. Thus, incoming traffic can be seen as anomalous and likely originates from scanners. In contrast, a honeypot operates as an actual host in frequented networks. It pretends to serve certain services to trick an attacker into believing that it is an easy target.

Data obtained with the help of these tools is analysed to extract information about the origins of attacks. Some providers publish their findings in online data feeds. Intrusion detection and prevention systems use this kind of information [16] and fetch the data from those feeds.

2.1 Related Work

Passive network data analysis has been an element of research for a while. In 2004, the value of network telescopes was assessed [21]. The paper shows how network telescopes can help to identify global events by analysing the unsolicited network traffic, which arrives at the network telescope. In 2012, Woodhead used network telescopes to analyse backscattering and packet noise on the internet to analyze malicious events on the internet [28]. The research focuses mostly on providing statistics about malicious traffic. A more detailed analysis of traffic on darknets has been performed by Fachkha et al. in the same year [7]. This paper distinguishes itself from Woodheads work by focussing more on technical analysis. It elaborates on the type of threatened systems and the software which was targeted.

Although network telescopes are more useful to detect DDoS activity, Richter describes a way to utilize them to monitor scanning activity [24]. He presented his findings in 2019 and shows that scanners target different subnets. The Heo and Shin paper of 2018 has a similar field of research [10]. The paper explains how network telescopes can be used to monitor scanning activity. In comparison to the Richter paper, their research has used a much smaller network telescope to trace scanners.

The second technique to identify scanners are honeypots. In their 2011 paper, Marchese et al. described how they identified the interests of attackers by the use of honeypots [18]. Just recently, Thom et al. have published a paper in which they analyze honeypot data of all over the globe [27]. Their data analysis shows that successful login attempts are shared among the bots within global botnets. Furthermore, they analyze the actions which a bot performs, once it gains access to a host.

In 2018, Kristoff published a paper in which he describes the architecture of his data feed service [16]. A similar system design is presented by Bloedorn et al. in 2001 [2]. Unlike Kristoff's paper, Bloedorn provides a framework for data mining in general. Nevertheless, the systems architecture reveals many similarities. Both papers have inspired the architecture used for the work in this paper.

Part of this work is also the assessment of the dataplane.org data feeds. Similar research has been conducted before. Recently, Feal et al. have focused on the transparency of open source blocklisting [8]. Furthermore, Li et al. also consider commercial providers of blocklists. Both papers conclude that it is a challenging task to acquire ground truth to assess data feeds.

3. METHODOLOGY

This paper aims to assess the quality of the dataplane.org feeds. To do so, we construct a database and collect data of potential scanning activity on the internet over a period of two weeks. Afterwards, we evaluate the obtained data set. In general, we try to uncover if the type of network influences the scanning activity.

We try to answer the hypothesis of Section 1 by analysing the requesting IP addresses. We expect different geographical properties and temporal characteristics when analysing the data concerning the type of targeted network. Lastly, we compare the scanner IPs gathered by our honeypots with the data feeds of dataplane.org.

This section describes the information gathered by our sensor infrastructure the methods with which we assess the data. Further, it describes the setup of a typical honeypot. The purpose of a honeypot is to gather data. Further, we provide a high-level description of the entire sensing infrastructure.

3.1 Activity of Scanners

A login attempt on one of our honeypots, also called a sensor, logs not only the originating IP address but also the timestamp. Additionally, when the data is added to the database, it is flagged with the originating type of network as well as a unique honeypot identifier. These attributes offer plenty of options to analyse the scanning activity. The paper will focus on the geographical distribution of activity as well as on temporal activity.

The geographical analysis will reveal which countries are most active when considering the scanning activity of different protocols. Further, we will try to argue why certain countries lead the list of scanning activity. The analysis focuses not only on the number of requests but also puts it in relation to the number of distinct IP addresses on a country level. The analysis will be completed with a short excursion to traffic routed over the Tor network. Tor is a popular tool to hide the identity of its users.

The timestamp of the login attempts provides various ways to analyse the number of login attempts. The hypothesis is that scanning activity differs, depending on parameters like the hour of the day or the day of the week. Even the difference between working days and weekdays could reveal some irregularities. Also, public holidays may influence the scanning activity.

After inspecting the activity mentioned earlier, we focus on the combination of temporal and geographical analysis. We assess scanning activity with respect to the scanners timeyone. The focus of this analysis is to inspect if the local time of the scanners has an influence on the activity. As an example, a certain region can have a general base-load of scanning activity. This activity can be caused by automated around-the-clock scanners. During the day-

time, when the local public is awake, we might encounter an increased number of IPs and requests. Such an increase could potentially be explained by manual scanning on top of the around-the-clock scanners.

Additionally, we benchmark the dataplane.org feeds to allow network operators to decide how useful those feeds will be when integrated into their own defensive network mechanics. To answer this, we mostly focus on the originating IP addresses of scanning activity. First, we determine the share of overlapping IPs. We evaluate the number of addresses recognized by our sensors. Then, we compare how many of them are also listed in the data feeds.

Next, we investigate the time passed when an IP address shows up in the *dataplane* data feeds after our own honeypots have recognized it. According to their website, the dataplane.org feeds are updated each hour. This allows us to accurately determine the time delta between the moment of our detection and the moment a scanner appears in the feeds.

3.2 Setup of a Honeypot

A central component of this research is to gather data about scanning activity in the three different types of networks: cloud, campus and residential. We operate honeypots in each of this kind of network to log unsuccessful login attempts of scanners. In this work, a honeypot is required to run on a Debian-based Linux system, such as Debian, Ubuntu or Raspberry Pi OS. Furthermore, a honeypot needs to be capable of recording login attempts for SSH and Telnet. The use of well-established software mitigates the risk of failure due to an immature product or implementation. Hence, we decided to use OpenSSH as our SSH service of choice. As a Telnet software, we chose the popular telnetd service which is part of the GNU inetutils package.

To make the honeypots attractive for scanners, the services are configured such that they listen on the default ports of the corresponding protocols: Port 22 for SSH and port 23 for Telnet.

The software was configured such that it denies unauthorized access. Failed login attempts are written to the system log files. Each night, the Linux logrotation tool rotated the log files. Afterwards, a script automatically extracted those records which are relevant for this research and transmitted them to a central database server in text format.

3.3 Setup of the Database Server

The data gathered from the different honeypots is stored centrally in a PostgreSQL database. The database server reads the text files which were previously transmitted by the honeypots, extracts important information and stores it in different tables. As already described in Section 3.1, the data feeds are updated on an hourly basis. To determine the overlap of scanner IP addresses we need to chronologically store the information of the data feeds. Therefore, a cron job downloads the two data feeds for SSH and Telnet each hour and imports them into the database.

Last but not least, some data evaluation requires geographic IP information. Hence, we query geoinformation of all IP addresses recognized by our honeypots from the online services *ipinfo.io* [12] and *ip2location.com* [1].

3.4 Actual Infrastructure

We operated 11 honeypots in total. An overview of honeypots and their locations is provided in Table 1.

ID	Type	Location
1	Campus	University of Twente, Enschede, NL
2	Campus	University of Twente, Enschede, NL
3	Campus	University of Twente, Enschede, NL
4	Residential	Hengelo, Overijssel, NL
5	Residential	Borne, Overijssel, NL
6	Residential	Arnhem, Gelderland, NL
7	Residential	Naples, Campania, IT
8	Cloud	Digital Ocean, Amsterdam, NL
9	Cloud	Digital Ocean, Frankfurt, GER
10	Cloud	Hetzner, Nuremberg, GER
11	Cloud	myLoc, Dusseldorf, GER

Table 1: Overview of Honeypots and their locations

Network	Protocol	# IP Addresses	# Requests
All	**SSH**	5,983	648,952
	Telnet	1,199	5,981,546
Cloud	**SSH**	3,667	311,239
	Telnet	416	2,742,344
Campus	**SSH**	2,395	172,419
	Telnet	371	1,434,345
Residential	**SSH**	875	165,294
	Telnet	431	1,804,857

Table 2: Comparison of the number of IP addresses and the number of requests with respect to the protocol recorded by the honeypots.

The honeypots operate in three kinds of networks: Residential area networks, campus networks and cloud infrastructure. Although there are more types of networks that can be investigated, for instance industrial networks or governmental infrastructure, we decided to include these types of networks in this work. The reason is that we have easy access to operate a honeypot in them. In each different type of network, the honeypots were located in different locations to ensure diversity. The choice of locations is a trade-off between availability, ease of access and time constraints.

4. RESULTS

This section first elaborates on the data set and its limitations. Then it explains the differences between SSH and Telnet scanning before providing insights into the geographical distribution of scanning activity. Furthermore, we briefly elaborate on the role of Tor routed scanning. The following analysis focuses on temporal characteristics of scanning activity and evaluates the behaviour of scanners with respect to their local timezones. We conclude this section with the assessment of the dataplane.org feeds.

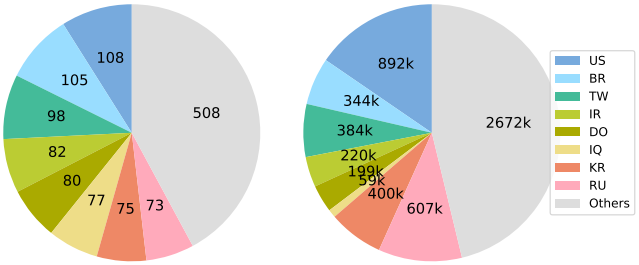
4.1 Dataset

The data set used in this work is obtained by our own infrastructure. The honeypots were effectively logging for two weeks from Monday, May 17 to Sunday, May 30. During these two weeks, we encountered thousands of different IP addresses of scanners as well as millions of requests. More details are provided in Table 2.

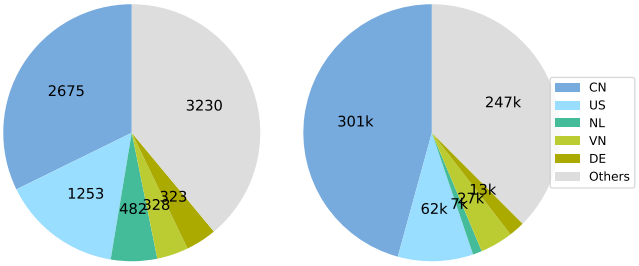
As the table shows, the number of scanner IPs for Telnet is significantly smaller than the number of SSH scanners. In contradiction, the number of requests is larger by a factor of approximately 10 for Telnet.

4.2 Limitations

One of the limitations of this work is the choice of honeypots. Time constraints forced us to choose honeypots



(a) Telnet scanners per country



(b) SSH scanners

Figure 1: Countries with largest shares of scanning activity according to our honeypots. The number of IP addresses on the right, the number of requests on the left.

based on availability than on diversity. Thus, the majority of residential honeypots were located in the Netherlands. Likewise, the only campus network considered in this work was the network of the University of Twente.

Additionally, the two-week period is rather short. Although two weeks is sufficient to provide reliable insights in a diurnal analysis, e.g. the hours of the day, it is rather short to provide conclusive evidence on weekly patterns.

Lastly, the protocols SSH and Telnet were chosen on its widespread popularity. More protocols can provide more insights but also requires more time to evaluate the data set.

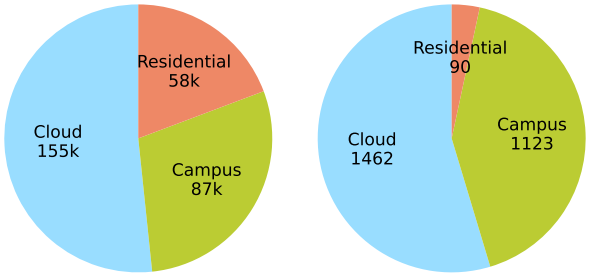
4.3 SSH and Telnet Scanning

For decades, SSH is the preferred choice for remote access. During our research, even the German Federal Office for Information Security (BSI) contacted us since they discovered an open Telnet service on one of our cloud servers. Clearly, on servers, the use of Telnet is discouraged in favor of the more secure SSH protocol.

In practice, SSH servers are secured with additional software such as Fail2Ban to defend against brute force attacks [13]. Telnet became more popular with the rise of the Internet of Things (IoT). Those are usually low-powered devices with very few resources available. Thus, they rather use the more lightweight Telnet protocol for remote management [29, 20]. The infamous Mirai botnet has put this obvious problem back on the agenda [19, 10]. Furthermore, IoT devices do not operate intrusion prevention software such as Fail2Ban, which means they do not slow down brute force attacks. That would explain the difference mentioned at the beginning.

4.4 Geographical Activity of Scanners

This subsection explains the findings concerning the geographical distribution of IP addresses and requests. First, we focus on Telnet scanners before discussing SSH scanning.



(a) Number of requests (b) Number of IPs

Figure 2: Overview of total numbers related to Chinese SSH scanning activity by the type of network.

As Figure 1a shows, Telnet scanning is distributed equally over the globe. Although there are smaller differences between the leading countries, there is no country that clearly dominates Telnet scanning activity. The number of IP addresses as well as the number of requests is rather uniformly distributed. In general, we observe that countries with a developed IT infrastructure also cause many login requests related to Telnet scanning.

Compared to Telnet, SSH scanning differs a lot. As shown in Figure 1b, two major countries dominate the scanning activity of SSH - the US and China. Both countries operate a huge number of scanners (according to the number of IP addresses). A similar pattern is shown when considering the number of requests. Mostly China, but also the US is leading in this field. An interesting observation is, that the Netherlands has quite a large number of scanner IP addresses, but a particularly small number of requests. This observation is discussed in Section 4.5.

Another interesting observation is, that Chinese scanners mainly target cloud and campus infrastructure, as Figure 2 clearly shows. Fewer requests are sent (Figure 2a) but more importantly, the number of scanners targeting residential areas is negligible in comparison to campus and cloud networks (Figure 2b).

The prevalence of Chinese scanning activity in campus and cloud networks might be related to interest in intellectual property. According to Segal, China is one of the main intruders concerning the theft of intellectual property from governmental and corporate computers [25].

As described previously, Telnet scanning is distributed much more diverse among different countries than SSH scanning. What exactly causes such a different scanning behaviour?

On the one hand, there is SSH. The SSH protocol is widely used to administrate regular servers. Most of the operating systems based on Unix use SSH as the default tool for remote administration. Therefore, SSH scanning intends to break into the server and steal data. On the other hand, there is Telnet. Telnet is the predecessor of SSH. It is insecure and hence, no longer used for server access. However, the Telnet protocol is more lightweight and a Telnet service requires fewer resources to operate. This makes it attractive as a remote management tool for IoT devices. In 2017, Margolis et al. have analysed the IoT centric Mirai botnet [19]. Comparing the leading countries of Telnet scanning to existing research such as the Margolis et al. paper, we find almost the same list of countries. Consequently, we can say that a major part of Telnet scanning is likely connected to botnets.

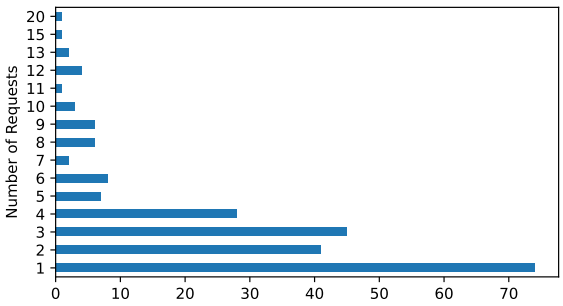


Figure 3: Overview of the number of IP addresses of Tor exit nodes counted based on the number of requests each of them sent

Figure 4: Comparison of login attempts routed over the Tor Network. Number of IPs on the left, number of requests on the right.

4.5 The Importance of Tor Traffic

When talking about malicious internet traffic one should also consider a popular medium of cybercrime - the Tor network, or simply Tor [6]. Tor is used by cybercriminals to hide their traces. The Onion Router, or Tor for short, is a tool, which implemented the idea of onion routing [9]. It allows sending network traffic encrypted and anonymously. If a user sends a request to a server over Tor, the encrypted request is routed over multiple Tor nodes before the exit node sends it to the server.

The list of IPs of exit nodes is published online by the Tor project [26]. We used this list to filter the gathered login attempts on this list of IPs. Summarizing the requests and their corresponding IP addresses reveals that login attempts originate from plenty of different Tor exit nodes. However, all of those Tor routed requests are all targeting SSH services. Not a single Telnet request was routed over Tor.

According to our observations, Telnet scanning is presumably not performed over Tor. It is a hint that Telnet scanners do not aim to stay anonymous while scanning. One of the possible explanations could be a command and control architecture of botnets, where the compromised bots do not require to stay anonymous.

Moreover, there are two indicators why we assume that Tor routed SSH logins do not originate from automated scanning. Firstly, as Figure 3 shows, the number of requests sent by each Tor exit node is comparably small. 85% of the IPs have not performed more than 5 requests. The statistic is headed by a single IP, a US-based exit node, with 20 requests. Figure 4 provides an overview of the vast differences. Approximately 500 out of 8000 IP addresses are routed over Tor (left pie chart), but just a marginal number of 752 login attempts were made from Tor nodes (right pie chart). We speculate that such small numbers of requests are likely performed manually. Secondly, the usage of Tor. Tor routing is much slower than regular routing, which makes it inconvenient for large-scale scanning operations.

4.6 Temporal Activity of Scanners

On average, each honeypot has received around 5,000 SSH and 50,000 Telnet requests. Cloud-based honeypots registered around 6,000-8,000 requests per day, campus-based honeypots around 4,000-6,000. Residential honeypots logged just around 3,000-4,000 requests daily. The picture for Telnet looks different, as shown in Figure 5. Firstly, the amount of requests per day varies significantly more. Secondly, there is no clear order in which we could arrange

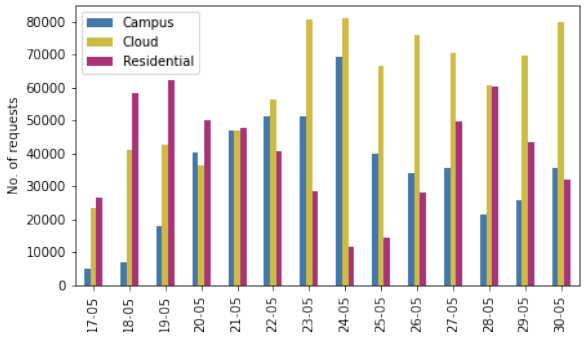


Figure 5: Number of requests logged on each day of the logging period.

the network types.

It appears, that the first week, scanners target mostly residential areas, although, during the weekends the number of requests declines for residential areas. In the second week, most requests appear on cloud platforms.

Further, we compared the number of login attempts per honeypot within each type of network. The more popular cloud providers, such as *Hetzner* [11] and *DigitalOcean* [5] are targeted more frequently than the less known cloud provider *myLoc* [22]. This holds for both protocols, Telnet and SSH.

To explain this phenomenon, internationally operating cloud providers usually grew over the years. This means, they have a long corporate history, which automatically leads to more exposed public data, such as IP ranges. The data is available due to research and data leaks. Thus, malicious scanners can be implemented such that they target specifically those cloud providers in order to increase their chance of success.

On campus networks, two out of three honeypots were targeted similarly heavily. However, one honeypot did neither register many SSH requests, nor many Telnet requests. To give an example: Two of the campus-based honeypots received around 5000-7000 SSH login attempts each day. The third one just around 100-1000. Similar extremes are observed for Telnet data. We cannot find a clear reason to explain this outlier.

Last but not least, the residential honeypots. The IP addresses of two out of four honeypots were already hosting an SSH service before we used them to operate honeypots. Those were the honeypots in Arnhem, the Netherlands and the one in Italy. Exactly these two honeypots also

logged twice as many requests per day as the other two honeypots.

A possible explanation is that the IP addresses of these hosts potentially end up in scanning databases of (distributed) scanners. Specifically these two hosts might receive an additional number of requests compared to new honeypots.

Unlike the situation with SSH, no honeypot had the default Telnet port exposed to the Internet before we set it up. Therefore, also no regularity can be observed. Two honeypots receive 60,000-140,000 requests per day, one honeypot received between 10,000 and 40,000 Telnet requests daily. The last honeypot is an exception. This one receives a marginal number of 500-1000 requests per day. Since all honeypots are located in a different location, one possible explanation is that the operating ISPs have different mechanics to block malicious traffic.

Previously, we described that the day of the week might have an impact on scanning activity. We expect a change in the number of requests during the weekend, assuming that the weekly routine also has an influence on cyber criminality. However, when we look at the sum of requests per day of the week, we cannot discover a trend. SSH peaks on Mondays and Saturdays in cloud environments. Those peaks are caused by the peaks of the *Hetzner* honeypot.

Furthermore, Telnet peaks on Sundays in cloud environments. For both protocols, the peaks are not particularly distinctive from the other days. From only these two weeks of data gathering, we cannot conclude that weekends are more active than regular working days. Nevertheless, there are slight tendencies. Similar research in a larger setting that lasts considerably longer can help to find more definite answers.

4.7 Temporal Activity per Timezone

The remaining temporal analysis focuses on each hour of a day. First, we inspect the total number of requests hour by hour. Afterwards, we dive deeper into the timezones of the requesting IP addresses.

All the honeypots operate in the same time zone - UTC+02:00. This means we can inspect the logging attempts in the local Dutch time. Considering SSH by the hour, the sum of requests mimics a wave pattern. The maximum turning points appear approximately at the same time of the day. The first one appears in the early morning, the second one around midday and the last one before midnight. The pattern is visible in Figure 6. The slight peaks are highlighted with a light background. The same pattern appears if we inspect the traffic of each network separately. Therefore, we suspect a connection.

In Section 3.1 we proposed that each timezone has a baseload of scanning activity and an increased active time of the day caused by the sleeping pattern of locals. The observation of Figure 6 is an indicator of this hypothesis. Evaluating the activity by the hour mapped on the local time of the scanners leads to more explicit results.

Before doing this, we first focus on the Telnet plots. In terms of Telnet, the wave pattern is more regular. Still, the increasing and declining patterns are present but they span over more hours of the day.

To go into more detail with the previous statement, we realigned the timestamps according to the timezone of the originating IP address and highlighted the active hours of the day. Figure 7 shows three example timezones that

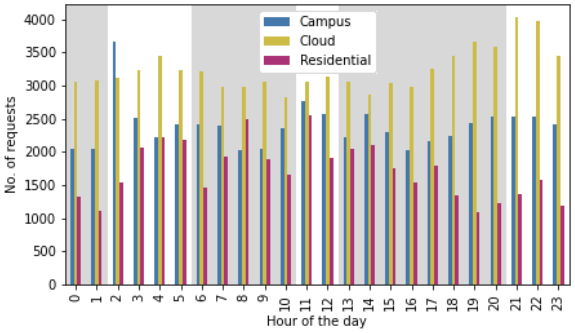


Figure 6: Sum of SSH requests classified by hour of the day. Peaks are highlighted (white background).

demonstrate the assumption of having more frequent and less frequent hours of the day. The timezone UTC-07:00 is one of the US timezones. Moscow, Belarus and the Middle East are located in UTC+03:00 and China, as well as South Asia, belong to UTC+08:00.

A similar (and more definite) observation can be made with Telnet requests. Figure 8 is an excerpt of timezones in South America (UTC-04:00) as well as Europe and Africa.

All in all, the hypothesis of having a base-load of scanning activity in each timezone that temporarily increases seems to be indeed true. A similar observation was made by Dagon et al. which explain the differences with hosts being switched off regularly [4]. Nevertheless, the peak times we observed are not necessarily during the regular daytime working hours. This makes it hard to say definitively whether these patterns are connected to human intervention or to switched on hosts (rather than to fully automated scanning). However, the data of timezone UTC+00:00 in Figure 8 seems to be a scanner that is just active during certain hours of the day.

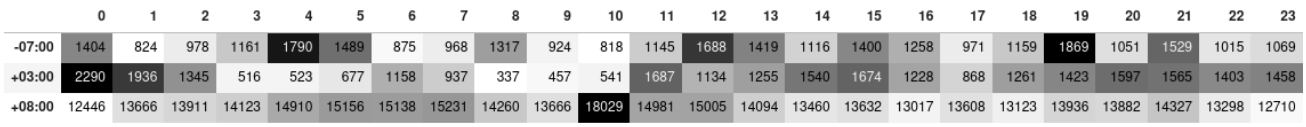
4.8 Coverage of Dataplane Feeds

This subsection determines the completeness of the dataplane.org feeds. First, to put the results into relation, it helps to have a broad idea of their infrastructure. According to Kristoffs paper of 2018, the dataplane infrastructure operates around 100 sensors on 6 continents with at least one IP in about 1/3 of all IPv4 /8 networks [16]. Kristoff claims he opted for cloud-based sensors only.

In comparison, our sensor infrastructure incorporates 11 honeypots of which just four are cloud-based. Clearly, our sensors will not find as many scanner IP addresses as the dataplane scanners can. The expectation is that a major part of those scanners, which target cloud infrastructure will also appear in the data feeds.

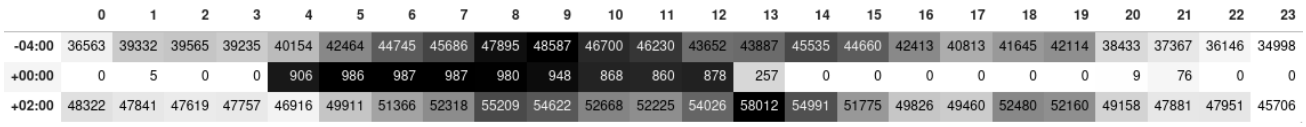
Similar to the previous sections, the data evaluation distinguishes between SSH and Telnet data. During the two-week period, our sensors registered 5174 distinct IP addresses, which targeted SSH honeypots. Out of these 5174 IPs are 4345 IP addresses (84%) also covered by the SSH data feeds online.

Table 3 provides an overview of the number of IP addresses logged by our sensors. The table shows two lines per network type. The first row shows the number of distinct IP addresses logged in this type of network. The second row only shows those IPs which exclusively appeared in this network. The term “exclusive” can be described best with an example: The campus-based honeypots registered 2395



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
-07:00	1404	824	978	1161	1790	1489	875	968	1317	924	818	1145	1688	1419	1116	1400	1258	971	1159	1869	1051	1529	1015	1069
+03:00	2290	1936	1345	516	523	677	1158	937	337	457	541	1687	1134	1255	1540	1674	1228	868	1261	1423	1597	1565	1403	1458
+08:00	12446	13666	13911	14123	14910	15156	15138	15231	14260	13666	18029	14981	15005	14094	13460	13632	13017	13608	13123	13936	13882	14327	13298	12710

Figure 7: Example timezones with number of SSH requests counted aligned by the active hours of the day, local time of the scanners



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
-04:00	36563	39332	39565	39235	40154	42464	44745	45686	47895	48587	46700	46230	43652	43887	45535	44660	42413	40813	41645	42114	38433	37367	36146	34998
+00:00	0	5	0	0	906	986	987	987	980	948	868	860	878	257	0	0	0	0	0	0	9	76	0	0
+02:00	48322	47841	47619	47757	46916	49911	51366	52318	55209	54622	52668	52225	54026	58012	54991	51775	49826	49460	52480	52160	49158	47881	47951	45706

Figure 8: Example timezones with number of Telnet requests. Similar to Figure 7.

distinct IP addresses which tried to break in. Out of these 2395 addresses, only 1689 do not show up in the logs of honeypots in cloud or residential networks.

As the table shows quite clearly, the majority of IPs observed on our honeypots also appear in the dataplane.org feeds. Around 90% of the scanner IPs also show up in the dataplane.org feeds. The scanners of residential areas show a different picture. Only 28% of the IPs observed in our residential honeypots also show up in the *dataplane.org* feeds. Not only is the coverage smaller, but also the number of IP addresses.

Likely, some of the scanners of residential areas do not target cloud or campus infrastructure. We can just speculate about the motives. Maybe, these scanners have specialized in targeting consumer hardware, such as home routers and firewalls. Possibly, they even intend to stay under the radar of Internet blocklists.

The coverage of scanner IP addresses targeting Telnet services differs heavily from the SSH data. Firstly, as already shown in Figure 1, the number of scanning agents is much smaller. Furthermore, we see the general coverage is extremely small. Cloud coverage peaks at 15%, whereas scanners that appear exclusively in the logs of campus-based honeypots have coverage of as low as 7%.

Although the coverage is different between the three network types, we can argue that the differences are small enough to accept that the observation is similar for all types of examined network types. In the case of Telnet, the difference is at around 8%, which is negligible if we compare it to the differences of around 65% of SSH.

We see the overlap of Telnet scanner IPs with the dataplane.org feeds is significantly smaller than the overlap of SSH scanners. All networks have a considerably small overlap. Consequently, the large majority of Telnet scanners do not show up in the dataplane.org feeds. With just 11 honeypots, we discovered around 1000 IP addresses not in the feeds of Dataplane. This number is extremely large when considering that we encountered around 1200 Telnet scanner addresses in total. This shows that the dataplane.org infrastructure has large potential to improve their Telnet blocklist.

For those IPs that we observe both in our honeypots and on the dataplane.org feeds, we now examine the time between us observing them, and the IPs appearing on a dataplane.org blocklist.

Figure 9 illustrates the data according to the previous statement. To be precise, the figure only covers those IP addresses which were first discovered by one of our honeypots before showing up in the data feeds online. We only

	Total	Not in Feeds	In Feeds	%
Cloud	3667	339	3328	90.8
Cloud excl.	2895	339	2556	88.3
Campus	3695	72	2323	97.0
Camp. excl.	1689	71	1618	95.8
Residential	875	420	455	52.0
Resid. excl.	590	419	171	29.0

Table 3: Distinct scanner IPs targeting SSH services in different kinds of networks incl. the coverage by Dataplane

	Total	Not in Feeds	In Feeds	%
Cloud	416	352	64	15.4
Cloud excl.	402	342	60	14.9
Campus	371	343	28	7.5
Camp. excl.	360	335	25	6.9
Residential	431	384	47	10.9
Resid. excl.	418	376	42	10.0

Table 4: Distinct scanner IPs targeting Telnet services in different kinds of networks incl. the coverage by Dataplane

consider addresses that we observe in our honeypots *before* they appear on a dataplane.org blocklist (and thus exclude addresses that never show up on dataplane.org feeds, or were already present in these feeds). The SSH box plot shows clearly that the vast majority of the internet addresses appear in the dataplane feeds within the first 24 hours of us observing them in a honeypot. In contrast, it is not uncommon that Telnet scanner IPs appear days after our own sensors registered them. The data evaluation revealed that the boxplots look similar for each network type.

This entire subsection revealed that the dataplane feeds for SSH are generally reliable. Although scanners of residential networks will likely not appear in the feeds, this type of scanner is also less common than campus and cloud scanners. Next to SSH scanners, we also investigated the completeness of the Telnet data feeds. Apparently, the coverage of the Telnet scanners is much smaller. Section 4.4 has shown that Telnet scanning is globally more scattered. The bad performance of the Telnet scanner feeds can have multiple explanations. Either, the scanner IPs are hard to grasp, since botnets are extremely volatile and grow and shrink by the minute. Or the claimed diversity of dataplane sensors does not match the scanning behavior of Telnet scanners.

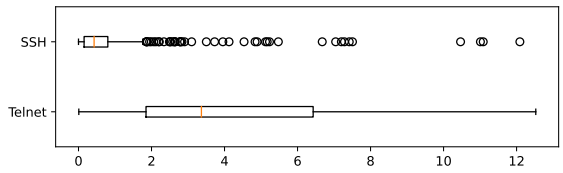


Figure 9: Days passed until the scanner IP showed up in Dataplane feeds

5. CONCLUSION

The goal of this work was to show that scanners exclusively target different networks. Thus, not having qualitative diversity in terms of network types leaves a blind spot on scanning activity. We obtained conclusive evidence by placing honeypots in different environments to argue that blocklist providers can not purely rely on cloud-based sensors.

Our sensor infrastructure operated honeypots in residential area networks, in cloud networks and on a University campus. We analysed scanner IPs concerning geographical and temporal properties in order to conclude that various scanners target different types of network.

We uncovered that the majority of SSH scanning originates from US-based and Chinese scanners, whereas Telnet scanning is distributed much more equally around the globe. Furthermore, we have shown that the diurnal rhythm of the local timezones of scanners affects its activity.

Lastly, we assessed scanners detected by our honeypots with the data feeds of dataplane.org. We provide evidence that the date feeds lack accuracy with respect to scanners that exclusively target residential networks. Additionally, we observed that data feeds of some protocols are less complete than others.

6. FUTURE WORK

Possible future work includes a deeper analysis of the data set to develop a better understanding of the scanning activity. Likewise, expanding the number of protocols and the time frame of logging can provide more insights and more convincing evidence. Moreover, as previous research has shown, the overlap in data feeds of providers differs. Thus, assessing more data feeds can reveal a better picture to argue about the completeness of these feeds.

7. REFERENCES

[1] Corporate Office Hexasoft Development Sdn. Bhd. IP2Location. https://www.ip2location.com/ [Online; accessed 18-June-2021].

[2] E. Bloedorn, A. D. Christiansen, W. Hill, C. Skorupka, L. M. Talbot, and J. Tivel. Data mining for network intrusion detection: How to get started. Technical report, Citeseer, 2001.

[3] A. Chuvakin. Honeypot Essentials. *Inf. Secur. J. A Glob. Perspect.*, 11(6):15–20, 2003.

[4] D. Dagon, C. C. Zou, and W. Lee. Modeling botnet propagation using time zones. In *NDSS*, volume 6, pages 2–13, 2006.

[5] DigitalOcean Inc. DigitalOcean Website. https://www.digitalocean.com/ [Online; accessed 24-June-2021].

[6] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. Technical report, Naval Research Lab Washington DC, 2004.

[7] C. Fachkha, E. Bou-Harb, A. Boukhtouta, S. Dinh, F. Iqbal, and M. Debbabi. Investigating the dark cyberspace: Profiling, threat-based analysis and correlation. In *2012 7th International Conference on Risks and Security of Internet and Systems (CRiSIS)*, pages 1–8. IEEE, 2012.

[8] Á. Feal, P. Vallina, J. Gamba, S. Pastrana, A. Nappa, O. Hohlfeld, N. Vallina-Rodriguez, and J. Tapiador. Blocklist babel: On the transparency and dynamics of open source blocklisting. *IEEE Transactions on Network and Service Management*, 2021.

[9] D. Goldschlag, M. Reed, and P. Syverson. Onion routing. *Communications of the ACM*, 42(2):39–41, 1999.

[10] H. Heo and S. Shin. Who is knocking on the telnet port: A large-scale empirical study of network scanning. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pages 625–636, 2018.

[11] Hetzner Online GmbH. Hetzner Website. https://www.hetzner.com/ [Online; accessed 24-06-2021].

[12] ipinfo.io. ipinfo.io Website. https://ipinfo.io [Online; accessed 18-June-2021].

[13] C. Jaquier, Y. Halchenko, D. Black, S. Hiscocks, and A. Busleiman. Fail2ban GitHub Repository, 2021. https://github.com/fail2ban/fail2ban [Online; accessed 27-April-2021].

[14] H. Kikuchi and M. Terada. How Many Malicious Scanners Are in the Internet? In *Information Security Applications*, pages 381–390. Springer, 2007.

[15] B. Knieriem, X. Zhang, P. Levine, F. Breitinger, and I. Baggili. An overview of the usage of default passwords. In *International conference on digital forensics and cyber crime*, pages 195–203. Springer, 2017.

[16] J. Kristoff. Building an Internet Security Feeds Service. *USENIX Open Access Policy*, page 31, 2018.

[17] V. G. Li, M. Dunn, P. Pearce, D. McCoy, G. M. Voelker, and S. Savage. Reading the tea leaves: A comparative analysis of threat intelligence. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 851–867, 2019.

[18] M. Marchese, R. Surlinelli, and S. Zappatore. Monitoring unauthorized internet accesses through a 'honeypot'system. *International Journal of Communication Systems*, 24(1):75–93, 2011.

[19] J. Margolis, T. T. Oh, S. Jadhav, Y. H. Kim, and J. N. Kim. An in-depth analysis of the Mirai botnet. In *2017 International Conference on Software Security and Assurance (ICSSA)*, pages 6–12. IEEE, 2017.

[20] L. Metongnon and R. Sadre. Beyond telnet: Prevalence of iot protocols in telescope and honeypot measurements. In *Proceedings of the 2018 Workshop on Traffic Measurements for Cybersecurity*, pages 21–26, 2018.

[21] D. Moore, C. Shannon, G. Voelker, S. Savage, et al. Network telescopes: Technical report. Technical report, Cooperative Association for Internet Data Analysis (CAIDA), 2004.

[22] myLoc Managed IT AG. myLoc Website. https://www.myloc.de/en.html [Online; accessed 24-June-2021].

[23] R. Padmanabhan, J. P. Rula, P. Richter, S. D.

Strowes, and A. Dainotti. DynamIPs: Analyzing address assignment practices in IPv4 and IPv6. In *Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies*, pages 55–70, 2020.

[24] P. Richter and A. Berger. Scanning the scanners: Sensing the Internet from a massively distributed network telescope. In *Proceedings of the Internet Measurement Conference*, pages 144–157, 2019.

[25] A. Segal. The code not taken: China, the United States, and the future of cyber espionage. *Bulletin of the Atomic Scientists*, 69(5):38–45, 2013.

[26] The Tor Project. Tor Exit Nodes. https://check.torproject.org/torbulkexitlist [Online; accessed 17-June-2021].

[27] J. Thom, Y. Shah, and S. Sengupta. Correlation of Cyber Threat Intelligence Data Across Global Honeypots. In *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0766–0772. IEEE, 2021.

[28] S. Woodhead. Monitoring bad traffic with darknets. *Network Security*, 2012(1):10–14, 2012.

[29] J. Wurm, K. Hoang, O. Arias, A.-R. Sadeghi, and Y. Jin. Security analysis on consumer and industrial iot devices. In *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 519–524. IEEE, 2016.

[30] H. R. Zeidanloo and A. A. Manaf. Botnet command and control mechanisms. In *2009 Second International Conference on Computer and Electrical Engineering*, volume 1, pages 564–568. IEEE, 2009.

[31] Z. Zhu, G. Lu, Y. Chen, Z. J. Fu, P. Roberts, and K. Han. Botnet research survey. In *2008 32nd Annual IEEE International Computer Software and Applications Conference*, pages 967–972. IEEE, 2008.