Blocklist Babel: On the Transparency and Dynamics of Open Source Blocklisting

Álvaro Feal, Pelayo Vallina, Julien Gamba, Sergio Pastrana, Antonio Nappa, Oliver Hohlfeld, Narseo Vallina-Rodriguez, and Juan Tapiador

***Abstract*—Blocklists constitute a widely-used Internet security mechanism to filter undesired network traffic based on IP/domain reputation and behavior. Many blocklists are distributed in open source form by threat intelligence providers who aggregate and process input from their own sensors, but also from third-party feeds or providers. Despite their wide adoption, many open-source blocklist providers lack clear documentation about their structure, curation process, contents, dynamics, and inter-relationships with other providers. In this paper, we perform a transparency and content analysis of 2,093 free and open source blocklists with the aim of exploring those questions. To that end, we perform a longitudinal 6-month crawling campaign yielding more than 13.5M unique records. This allows us to shed light on their nature, dynamics, inter-provider relationships, and transparency. Specifically, we discuss how the lack of consensus on distribution formats, blocklist labeling taxonomy, content focus, and temporal dynamics creates a complex ecosystem that complicates their combined crawling, aggregation and use. We also provide observations regarding their generally low overlap as well as acute differences in terms of liveness (i.e., how frequently records get indexed and removed from the list) and the lack of documentation about their data collection processes, nature and intended purpose. We conclude the paper with recommendations in terms of transparency, accountability, and standardization.**

***Index Terms*—Security Management, Functional Areas; Network Monitoring and measurements, Methods; Internet Connectivity and Internet Access Services, Service Management.**

I. Introduction

Network management practices often rely on data feeds assigning labels to network entities for access control, *e.g.,* feeds provided by threat intelligence platforms to identify and block content regarded as malicious, untrustworthy, or simply bad reputed. Such feeds are commonly regarded as *blocklists* (or *blacklists*) and can contain a variety of actionable records, including IP addresses, hostnames, URLs, TLS certificate fingerprints, or file hashes. The effectiveness of blocklisting to raise the bar against unwanted behavior, traffic, and content has driven a steady growth in the threat intelligence sharing ecosystem, with a variety of products and platforms streamlining the aggregation, enforcement, and distribution of actionable data. Even if the market penetration of blocklisting is hard to know, Spamhaus–one of the oldest threat intelligence providers–states that as of October 6, 2020 their blocklists are protecting an estimated 3,126,410,000 user mailboxes [1]. Similarly, MISP (Malware Intelligence Sharing Platform [2]), a popular open source threat intelligence platform, reports that it is used by more than 6,000 organizations worldwide. Beyond blocklists, other feeds can be used by applications to block or restrict access. One example are feeds of Tor exit node IPs, *e.g.,* used by Wikipedia to block users of the Tor anonymity network from editing Wikipedia due to abuses in the past [3].

Blocklists can be proprietary (*i.e.,* commercial) or open source. The former are often available through rate-limited, license-based, or pay-per-use mechanisms and are maintained by for-profit companies specialized in threat intelligence. On the other hand, we define *open source* blocklists as lists that are openly and freely available for anyone to collect and use. The ecosystem of open source blocklists is heterogeneous, with different types of providers, from for-profit companies that make publicly available—often partially—some of their data feeds, to individual users or organizations that publish their own collected and curated data. One key aspect of this diversity is that different providers follow different methods to collect, curate, maintain, and label their blocklists. Yet openness does not necessarily imply transparency. Often times, their internal data collection and sanitization processes are not sufficiently documented, thus impeding their optimal usage and, ultimately, their widespread adoption. We elaborate on these issues below:

- **Data sources:** While some providers operate their own sensors and honeypots to detect malicious activities, others might aggregate records from other providers' blocklists. Yet, many blocklist providers do not document their data sources. This impedes blocklist consumers from assessing the limitations, representativity, applicability, and scope of a given feed. Understanding the nature of the records contained in a blocklist and the potential overlap between different providers is instrumental for an optimal exploitation, particularly for end-users combining multiple feeds into a single, consolidated database. When open source blocklist providers fail to document their data sources, users might end up aggregating lists compiled for different purposes or that are indeed indexing the same entries, possibly under different tags or labels.
- **Data curation:** Blocklist providers should make available a clear description of the methodology followed to guarantee that the data they provide is sound and does not include false positives or outdated records. For instance, a given IP that was part of a botnet in the past can be benign later down the road. Similarly, when new malicious indicators appear, they

Á. Feal, P. Vallina, J. Gamba and N. Vallina-Rodriguez are with IMDEA Networks Institute.

Á. Feal, P. Vallina, J. Gamba, S. Pastrana, A. Nappa and J. Tapiador are with Universidad Carlos III de Madrid.

A. Nappa is with U.C. Berkeley

O. Hohlfeld is with Brandenburg University of Technology.

N. Vallina-Rodriguez is with ICSI.

should be included in well-maintained blocklists as soon as possible. For those blocklists feeding from each other, record propagation across lists might negatively contribute to the dissemination of false positives, making it harder to remove records once they are detected as such.[1] As a result, the update frequency of lists, including the addition or removal of records, is an important characteristic to measure their liveness and data curation processes.

- **Record and blocklist labeling:** Open source blocklists are widely heterogeneous in terms of the type of records contained in their feeds and the potentially harmful behavior that they try to protect against (*i.e.,* tracking, phishing, or malware). Yet, many providers publish their feeds without sufficiently describing their actual purpose or application. While some platforms rely on public taxonomies to label records in their list, many blocklists simply provide a custom label, if any. Even for those that try to label the nature of their data and their data sources, there are substantial methodological differences (and subjective perceptions) across providers when defining their records. For instance, labeling differences often come down to a granularity issue, with some lists using generic labels such as "malicious," while others offer a specialized classification system for malicious types. These differences in terms of labeling methodology and strategy are problematic because (*i*) they result in instances where a given entry appears in blocklists that have different purposes according to their providers; and (*ii*) complicate their comparison and aggregation into more comprehensive threat intelligence feeds.

The aforementioned reasons call for an empirical analysis and measurement of the soundness, freshness, dissemination, and nature of the records present in open source blocklists, as well as the transparency and documentation practices of their respective providers. This effort is essential not only to inform network operators relying on these security resources, but also to design more effective defenses that account for the complementary strengths and limitations of individual blocklists when used in isolation. While previous research efforts have analyzed the dynamics and intra-dependencies (*i.e.,* their overlap) of specific types of blocklists (*e.g.,* spam, phishing, or proprietary blocklists) [5], [6], [7], the dynamics, accuracy, and limitations of the open source blocklist ecosystem remain unknown. In this paper, we tackle similar problems but we look at the grand scheme of things, studying a dataset formed by a large number of open source blocklists that are not limited to a single type of malicious activities. We want to understand what are the problems that arise when a company or user aggregates open source blocklists from different sources and threat type. This strategy also allows us to study the dynamics and inter-dependencies between lists from different categories. We investigate how the lack of transparency from blocklist providers regarding their data collection, data sanitization and categorization processes can lead to problems to users that try to improve their protection by aggregating lists from different providers. This is an under-researched area that deserves a closer examination given the widespread use of open source blocklists in operational environments.

To address the challenges discussed above, in this paper we run an extensive campaign of identification and crawling of 2,093 open source blocklists from 69 providers (Section III). Our dataset includes feeds from prominent providers (*e.g.,* MISP or Spamhaus) and blocklists used for different cases and applications: from identifying users from anonymity networks to blocking malicious content. Our large-scale and diverse dataset allows us to gain a unique view to investigate fundamental questions related to the nature, dynamics, pitfalls, transparency and purpose of the open-source blocklist ecosystem. Our paper contributions are as follows:

- We inspect the ecosystem of open source blocklists providers, analyzing their characteristics and how transparent they are (Section IV). We also study the differences across blocklists in terms of the type of records that they include (Section V).
- We analyze the dynamics of the blocklists in our dataset by studying their changes (*i.e.,* record additions and removals) and update rates over time (Section VI). We observe that 30% of the lists never change and that 7% change at least daily. In the case of those blocklists that are updated, 12% only add entries, 32% only sporadically remove them (possibly due to sanitization efforts), and 56% both add and remove records.
- We study the relationships between blocklists in terms of content overlap (Section VII) and how records propagate among multiple blocklists over time (Section VIII). These complementary views allow us to observe a high overlap between some providers. We find groups of providers consistently adding and removing the same entries, which can be an artifact of providers feeding from the same sources or from one another.
- We show that many providers do not label their blocklists (40%), nor document their purpose. Moreover, in those cases where labels are provided, blocklisted entries are not labeled consistently across providers (14.8%). We also show that external data-sources can increase the confidence on provider labels, but they cannot fully solve the labeling problem. This result highlights the many challenges to be faced when trying to aggregate blocklists that supposedly protect against the same type of malicious or undesired behavior (Section IX).

Our analysis shows that the lack of ground truth about the maliciousness of a given resource and provider's opacity make it impossible to study issues such as the reason why a given entry is blocklisted, whether the inclusion of a given entry was correct, or which providers are "better" and should thus be trusted by users. Overall, our findings and observations provide constructive discussion on topics related to transparency, existing challenges, and best practices in operational settings for blocklists providers, end-users, and researchers. Finally, in order to foster research and allow for reproducibility, we make our datasets and parsers available to

[1] Anecdotally, a thread in AlienVault's forum [4] describes how false positives were found in one of their blocklist. When reported, the provider argued that this list was used only for historical reputation and risk analysis. Yet the entries propagated to other blocklists feeding from AlienVault that were unaware of this purpose, contributing to the spread of erroneous records across blocklists and thus to incorrect blocking.

the research community [8].

II. Related work

In the last years, several studies have measured and analyzed the effectiveness of blocklists to identify and prevent unwanted behaviors such as user tracking [9], [10], [11] or spam detection [12], [13], [14], [15], as well as to study the overlap between blocklist providers [16], [17], [18], [19], [20], [21], [22].

Sheng *et al.* showed that phishing blocklists are slow at detecting campaigns, and that they have large variations in terms of coverage, effectiveness, and speed of reporting [16]. Kuhrer *et al.* created a system that parses 49 malware blocklists, analyzing the presence of domains across them and their overlap [17]. They conclude that, while one can obtain a high number of domains from parsing these blocklists, several months of analysis are necessary to get a whole understanding of their dynamics. Phuong *et al.* analyzed the differences between 14 public and private lists, showing that some blocklists are almost identical and that different versions of the Google Safe Browser were developed independently [19].

Previous works have also defined and reported metrics to analyze the coverage, correctness or uniqueness of blocklists and threat intelligence services. Pitsillidis *et al.* studied the aptness of spam feeds for different purposes [6]. They showed that there are major differences in the way that data is collected and that users should therefore pick the feed that better fits their purpose. Metcalf *et al.* compared 86 different blocklists and reported on their scale and overlap [7]. They showed that IPs and domains are unique to a blocklists over 80% of the time and that there is little overlap across blocklists. Ramanathan *et al.*developed BLAG, which attempts to improve the accuracy of blocklists by aggregating different lists into a master blocklist [21].

Li *et al.* analyzed 55 threat intelligence sources—including proprietary ones—highlighting their limitations and shortcomings [5]. They designed metrics to measure characteristics of these lists and showed that there is substantial variation in terms on content across lists; that larger feeds do no necessarily contain better data; and that most IPs appear only on a single list. Finally, Ramanathan *et al.* showed that over 50% of the 151 publicly available IPv4 blocklists in their study contained reused IP address, and that this could affect as many as 78 legitimate IPs for up to 44 days [23].

Table I highlights the differences and similarities between our study and previous work. Our work provides a novel and complementary approach to the study of blocklists, focusing primarily on the open blocklist ecosystem, their nature, and their transparency. Contrary to some of the related work [16], [17], [6], we study the synergies between blocklists with different purposes. In addition, our large dataset of open source blocklists gives us an unique and more holistic view of the ecosystem of blocklists and their relationships [19], [5]. To the best of our knowledge, our work is also the first one to show how the lack of documentation and operational differences among providers can negatively impact end-users.

Ref.	# lists	Type	Time Span	Update strategy	Labeling issues	Overlap	Record propagation	Removal propagation	Transparency
[16]	8	Phishing	48 hours	✗	✗	✓	✗	✗	✗
[17]	49	Malware	–	✗	✗	✓	✗	✗	✗
[19]	14	Public/Private	–	✗	✗	✓	✗	✗	✗
[6]	10	Spam	3 months	✓	✗	✓	✗	✗	✗
[7]	86	–	1 + 1.5 years	✗	✗	✓	✓	✗	✗
[5]	55	Public/Private	1.5 years	✓	✗	✓	✗	✗	✗
[21]	157	Public	11 months	✓	✗	✗	✗	✗	✗
Ours	2,093	Public	6 months	✓	✗	✓	✓	✓	✓

Table I: Comparison to related work on blocklists

III. Data Collection

In February 2019, we started a snowball sampling process to discover and harvest as many blocklists and providers[2] as possible. First, we leveraged available online sources such as forums, blogs, and white papers to identify the most influential threat intelligence vendors in the blocklisting market [24], [25], [26]. We complemented this list with targeted web searches. Finally, for the purpose of this study, we filtered only those providers offering open feeds and datasets.

Our final dataset contains 2,093 feeds from 69 providers, including Dshield, Talos Intelligence, MalwareBytes, abuse.ch, Spamhaus, and the Malware Information Sharing Platform (MISP) [27]. Given our focus on open-source intelligence, we exclude commercial blocklists with strict ToS, record-oriented APIs, and rate-limited APIs (*e.g.,* Facebook's Threat Intelligence). We note that some of the blocklists in our collection are proprietary but were made publicly available without restriction. Also, some of the blocklists that were originally identified and added in our pipeline became obsolete during the period of study, e.g. due to owner no longer maintaining it or since they are only active for a particular set of events like a ransomware campaign. We analyze the liveness of the blocklists in Section VI.

We intentionally looked for blocklists of different origins and purposes: from blocklists covering malware campaigns (*e.g.,* Maltrail [28]), to feeds listing IPs associated to anonymity networks (*e.g.,* Blutmagie for Tor). While the latter might not be advertised as blocklists, they can be used for secondary purposes such as blocking users from these networks (*e.g.,* as used by Wikipedia to prevent Tor users from editing [3]). Therefore, we not only include network entities that should be blocked because their content is deemed as dangerous, untrustworthy, or inappropriate, but also listings of elements that can be actioned in more complex filtering policies and applications (*e.g.,* whitelisting domains or reputation scores). For simplicity, we will use the term "*blocklist*" throughout the paper to collectively refer to all the feeds in our dataset. It can be argued that the inclusion of these lists could be problematic as most people would not necessarily use them for blocking purposes. However, the high overlap between them and actual blocklists justifies this decision (further details in § VII).

[2]We define a provider as any entity (organization or individual) that compiles records into a blocklist and makes it publicly available, regardless of the type of blocklists or entity providing it.

The diversity across different lists and providers introduces side effects as the data format used by each blocklist lacks homogeneity. These formats range from simple text files listing the items (one line per entry) to structured formats such as JSON files, and even custom ones (*e.g.,* entries grouped together and separated by a custom delimiter). The lack of a standardized format required us to develop customized processors to parse each selected blocklist.

A. Blocklist crawling approach

As a result of our blocklist discovery process, our dataset grew over time. The observations of this paper covers a 6-month period (August 7, 2019 to February 13, 2020) for which we have a stable set of blocklists. We crawl and timestamp each blocklist every 8 hours and then parse them, discarding comments and malformed entries.

During this 6-month period, we collected more than 13.5M unique records from 2,093 blocklists. We parse and classify each blocklist record by its type (*e.g.,* domains, IP blocks, URIs). Then, for each new hostname and URL, we augment our dataset with active measurements executed sequentially as new entries get indexed from two European countries. Specifically, we obtain `A`, `AAAA`, `MX`, `NS`, and `CNAME` records. Finally, we leverage Fortiguard's service to label domains and IPs according to a commercial service [29]. This process will allow us to compare labels across services in Section IX using a reliable reference.

Ethical considerations. This study is not considered as human-subject research. The analysis is based on publicly available data. We notified both our national CERT and our academic network provider prior to any data collection, particularly for the active measurements.

B. Challenges and limitations

We next report the limitations that we have identified during our data collection and curation processes.

Dataset. Some blocklists providers such as Maltrail and MISP create blocklists tailored to specific attack campaigns. The unique perspective on what is a "blocklist" for these providers has also implications in the analysis, since their dynamism should be measured not only in terms of records per blocklist, but also in terms of newly created blocklists. We indexed the blocklists offered by these providers only once. Therefore, the interpretation of our results for these providers should be considered with this limitation in mind.

Crawling periodicity. Our 8-hour tick crawling strategy imposes some restrictions on our measurement and the interpretation of some of the results, particularly those related to the propagation of records across blocklists. Nevertheless, this affects a relatively low number of blocklists (as we report in Section VI, only 7% of the blocklists change at least daily), but it prevents us from analyzing blocklist behavior beyond the 8-hour granularity.

Record and format heterogeneity. We found a wild ecosystem of public blocklists that lack a commonly accepted format to distribute content. Consequently, we are forced to manually analyze each one of the 2,093 blocklist and develop ad-hoc crawlers and parsers to access and store their records. Even if we did our best to guarantee correctness by regularly checking for new formats, it is still possible that unnoticed changes on a blocklist (or on specific record formats) result in our parser missing some of the entries. To account for this, we keep a raw copy of the blocklist on every crawl, to re-parse a given instance if needed.

Problems in the crawling infrastructure. There were three days in which our crawling infrastructure went down. Specifically, we lack data for September 14th, October 17th, and November 7th, 2019. In some rare cases, some blocklists were unreachable during a given crawl.

IV. Blocklist Providers

As a result from our decision to collect and study open source blocklists from different providers and different purposes, we are able to collect a diverse dataset in terms of type of blocklists and providers. Table II provides high-level statistics for the most relevant providers. It is worth noting that the volume and type of records vary greatly across providers. Six different blocklists offered by Bambenek account for more than 5M records, and over 99% of them are not indexed by any other blocklist provider. The analysis of the records offered by each provider also talks about their diversity in terms of data collection. While MISP and Maltrail account for the majority of the individual blocklists, they account for less than 2% of the total records. This reflects on their tendency to release a large number of highly-specialized blocklists. In terms of the type of records distributed, we observe a clear distinction between comprehensive providers (*e.g.,* Abuse.ch and MISP) that offer blocklists covering many types of items and others that only index specific types of record and application.

A. Providers transparency

Table II shows that open source blocklists providers present differences in terms of the type of data that they include, how they collect this data, and how they present it to the users. These providers need to make this information available in their documentation so that users can make a correct use of open source blocklists. To verify how prevalent this practice is, we visit the homepage of all 69 providers and analyze whether they inform users about their collection methods, the taxonomy used to classify their records, or their sanitization strategy to prevent the inclusion of false positives.

In the course of this study, some of the providers' webpages have become unavailable (9%). We find that, overall, it is common (83%) to find contact details which allow users to reach the provider for different purposes (*e.g.,* to report a false positive). However, the lack of transparency becomes more evident when analyzing more specific details:

- **Data sources and data collection:** 31% of the providers do not provide any relevant information about their data sources and data collection processes. For those that disclose this information, we find that they often do a combination of the following strategies: (*i)* aggregation of other blocklists (21%); (*ii*) use of their own detection techniques (33%);

Table II: Summary statistics of the 10 largest blocklist providers (by the number of records) in our dataset.

Provider	Blocklists	Records							
		Aggregated	Daily Median	Unique	IP (Prefixes/Blocks)	Hostnames	URIs	Tor	Other
Bambenek	6	5,325,326	64,820	99.73%	0.12%	99.51%	-	-	0.37%
MalwareWorld	4	2,310,696	147,674	10.76%	79.76%	19.38%	0.01%	0.01%	0.85%
Lashback	1	1,859,313	268,207	73.07%	100.00%	-	-	-	-
NiX Spam	1	1,085,245	6,946	52.53%	100.00%	-	-	-	-
Firehol Project	16	991,322	3,622	55.06%	100.00%	-	-	-	-
MalwareBytes	15	909,425	129,977	94.01%	-	99.99%	-	0.01%	0.01%
Blacklist.de	1	495,649	28,797	0.24%	100.00%	-	-	-	-
AlienVault	1	422,605	74,793	0.02%	100.00%	-	-	-	-
Cins	1	420,420	15,000	1.32%	100.00%	-	-	-	-
Malshare	1	405,445	2,369	100.00%	-	-	-	-	100.00%

and (*iii*) user submissions (11%), typically by means of the technology supplied by the provider.

- **Data curation:** Only 20% of the providers declare that they have data sanitization process to detect and eliminate false positives, and include contact forms or email addresses for reporting errors.
- **Labeling strategy:** For 33% of the providers we cannot find any documentation related to how they classify the entries in their dataset (*i.e.,* a taxonomy). For those that provide such information, we find two clear approaches: (*i)* lists for which every entry has its own description (13%); or (*ii)* lists where every entry belongs to the same type of threat (43%).

To prevent blocklist misuse, developers should be clear about the expected usage of a given list. For instance, some lists are meant as a classification service rather than as blocklists (*e.g.,* listing IPs that belong to a given country or mobile operator [30]). We have identified a group of feeds commonly known as *aggressive lists* that are a superset of a vendor's standard feed, including entries detected by their algorithms or reported by users, but not proven to be malicious [31], [32]. Lack of documentation can lead to customers relying on these lists and unintentionally over-blocking.

Takeaway. The open source blocklists ecosystem is highly complex, with providers that follow very different data collection and sanitization strategies and often fail to document these processes. Overall, blocklist providers should take action to enhance the transparency of the blocklist compilation process, with clear and unambiguous guidelines that help operators and researchers decide which blocklists better fit their needs, and also limit undesirable record propagation.

In the rest of this paper we will strive to shed light into this ecosystem by analyzing the differences among providers in terms of: *(i)* the type of record that they contain (Section V); *(ii)* what are their update cycles and whether providers are proactive in adding new threats to their lists while removing false positives (Section VI); *(iii)* how the overlap (Section VII) and propagation of records across providers (Section VIII) show which providers are potentially feeding from each other or the same data sources; and *(iv)* how the lack of a common labeling framework can cause problems for end-users (Section IX), preventing sound usage and aggregation of lists from different providers.

Table III: Distribution of the number of records per provider and blocklists per record type.

	Unique records	**% providers**	**% blocklists**
Hostnames	6,861,861	33%	65%
IP (Prefixes/Blocks)	5,619,843	81%	48%
URIs	573,814	23%	64%
Onion	496	16%	4%
Other	518,081	35%	55%

V. Record-type Analysis

During the development of our blocklist parsers (Section III), we find five primary types of records which are not necessarily documented by each provider: hostnames, IP addresses (including individual IPv4 and IPv6 addresses as well as blocks), URIs and Tor hidden services. We group under the "Other" all the records that do not match any of the previous categories. Table II shows the percentage of records from each of these categories present in the blocklists from the most relevant providers. We refer the reader to Table III for detailed statistics for each category.

Hostnames. The majority of the records indexed by our blocklists are hostnames (51% of the total), over 75% of them uniquely indexed by Bambenek, the largest provider in our dataset. According to Fortiguard's domain classification, the majority of these domains are involved in malicious activities. Over 30% of the domains are classified as "Spam URLs" followed by "Malicious Websites" (5%) and "Phishing" (4%).

IP addresses and blocks. This category accounts for 41% of the total records and encompasses individual IPv4 and IPv6 addresses, IP blocks, and services represented by pairs of IP addresses and TCP or UDP ports (*i.e.,* `<IP>:<PORT>`). [3] We find 6,203 IP addresses with 890 different port numbers.[4] We use the DNS resolution of the domains found in our corpus of blocklists and find that 64 domains (median) resolve daily to IP addresses that are also blocklisted by other providers.

[3]Notably, the aggregate of all IPv4 records (single IPs and prefixes) account for 30% of the whole IPv4 address space.

[4]While the most frequent port numbers are 80, 443 and 8080, it is also worth noting the presence of port numbers 1604, 1177, and 12345. These TCP ports are associated with known malicious Remote Access Control services like DarkComent [33], njRAT [34], and NetBus [35].

The size of the IP blocks indexed by blocklists is diverse. The most common size of IPv4 prefixes is `/32` (which is in fact a single IP address) followed by `/24`. The largest IPv4 block that we identified is `103.160.0.0/11`, which was found in the FireHOL list [36] and remained present in all crawlings of this list that we performed. Some blocklists also contain arbitrary IP ranges (*e.g.,* `37.49.226.211 - 37.49.226.213`) that do not follow the standard CIDR notation, hence adding more complexity to the parsing process. In terms of autonomous systems (AS) owning these IP addresses or blocks, we find that the median number of ASes that have at least one blocklisted IP is 26,683 per day for IPv4, and 246 for IPv6. We found 130 ASes owning IP-level records constantly blocklisted during our 6-month analysis, including eyeball ISPs like Comcast (US) or AT&T (US), and cloud and hosting providers like SeeWeb (IT) or OVH (FR).

URIs. We identify 573,814 entries with an URI in 64% of the blocklists. We observe that 20% of the URIs report `<IP>:<PORT>` pairs rather than a hostname (78% of them are already indexed as IP addresses or hostnames).

Onion. We identified 496 different Onion services, which represent the smallest group based on the type of entry. In this group, we can observe 6 different suffixes, `.onion` (48%), `.onion.cab` (8%), `.onion.city` (2%), `.onion.direct` (2%), `.onion.to` (33%), and `.onion.top` (7%). These last five suffixes are typically used to get access to Onion sites without using Tor Browser [37]

Other. Finally, we classify as "Other" category those entries that do not match any of the previous categories. This category accounts for the 35% of the total entries, 90% of which are hashed entries of any type.

Takeaway. Our record-level analysis confirms that open source blocklists are very different in terms of the type of content that they provide. Therefore, it is important that providers clearly document what type of records they include. This will prevent users from undesired side effects such as over-blocking (*e.g.,* the combined size of all lists in our dataset leads to blocking 30% of the whole IPv4 address space).

VI. Growth and Liveness

It is important to understand whether providers update their blocklists regularly, and what are the update mechanisms. Addition of records suggests an interest towards completeness, while removals demonstrate willingness to avoid bloating and to remove false positives. During our observation window, we find that over 30% of blocklists remain unchanged but we see large differences in blocklists' dynamics based on their provider. Over 80% of the blocklist from providers like Maltrail and Abuse.ch were not updated during our observation window. The most extreme case are 7 providers publishing 8 blocklists who never updated their products during the observation window, including blocklists from national research and education networks like RedIris. The percentage of blocklists that remain unchanged also varies with the stability of the type of record: 34% of the blocklists indexing onion services never changed.

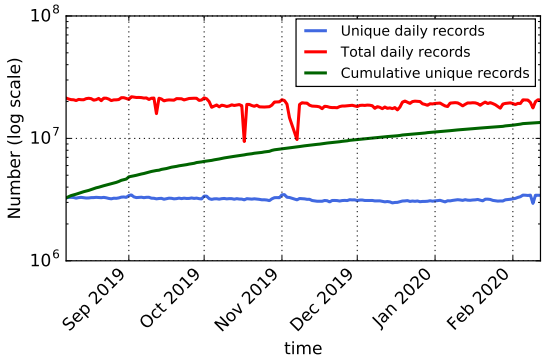


Figure 1: Unique, daily and cumulative entries per day

Figure 1 shows that the number of total records indexed per day remains quite stable for our 6-month study.[5] Yet, different blocklist update cycles cause slight variations in the number of total records indexed across all blocklists every day. The daily variance is larger for certain types of records: blocklists indexing IPs, hostnames, and URIs show a higher dynamism.

A. Update cycles

The frequency of changes varies across individual blocklists and providers. We focus on those blocklists that are indeed updated during our crawl period, and find that 7% change at least once per day (median value). This group contains 48 different providers, including Bambenek, Lashback, NiX Spam, and Firehol (See Table II). We observe that 11 blocklists from 6 providers, including PGL Yoyo, Turris, Cedia and Maltrail update their content weekly. The number and the nature of these changes varies across blocklists and providers:

1) 12% of the blocklists that change only add new entries, 70% of which are related to Maltrail. Nevertheless, these blocklists do not grow significantly as a result: only 14k new entries appeared in these blocklists during the entire measurement period when aggregated.
2) 32% of the blocklists only remove records (98% of them are published by MISP, which suggests efforts to minimize false positives). Overall, the total number of deleted records (48k) is not very significant.
3) We observe that 56% of the blocklists both remove and add entries in every update.

To better quantify the balance between the number of removals and additions, for every crawl, we compute the ratio of added and removed records as compared to the previous one. We focus the analysis on providers to avoid possible fluctuations between blocklists from the same provider (*e.g.,* due to entries moving from one list to another). We can see that the distribution of added and removed entries is very similar for most providers. However, there are some extreme cases. For instance: (1) Malshare hash blocklists regenerates almost completely its contents at every update, suggesting that they should be treated as newly observed entries; and (2) Malcode renews its contents with a ratio of 66% additions and 45% removals. We also observe differences in the update cycles of

[5]The three drops are due to outages in our crawling infrastructure (§ III-B).

blocklists that only index a single type of record. The trend for those that only index hostnames is to add more entries than they remove, while in the case of blocklists that only index IP addresses the behavior is the opposite. This is likely due to the fact that IP addresses are more volatile, and addresses that get re-assigned can lose their negative behaviors [23].

B. Record-level liveness

As discussed before, records might get removed at different rates due to different approaches to the sanitization process. To better understand this, we analyze for how long a given record survives in a blocklist. We study two months of data, from December 1st, 2019 to January 31st, 2020, to eliminate noise potentially introduced due to service outages in our crawling infrastructure (§ III-B). Our observations should remain valid for larger time windows. We observe that 30% of the entries, collected in the first week, never get removed. In order to reduce potential biases due to a record being removed late (we analyze latency in Section VIII), for each blocklist we track the entries indexed in the first week, between December 1st to December 7th, and we check if they remain indexed in the 2 months period in the same blocklist. We observe that 30% of the entries observed in the first week are never removed. Figure 2 shows the liveness of entries (*i.e.,* percentage of crawls in which an entry is present at the same blocklist) by record type. As previously, we compiled the entries observed per blocklists in the first week of December 2019, and then we computed the number of crawls each entry remain indexed on the blocklist from the 8th of December to the 31st of January. We can see that the number of entries that are never removed varies depending the type of entry, from the 43%, and 42% of the hostnames and Onion services, to the 11% of the URIs, However, we observe that 43% of the IP remains less than 50% of the crawls in the two months. In the case of the hostnames, the percentage of entries that remain less than 50% of the crawls drops to 14% of the entries. Further, 40% of the removed IPs got re-indexed during this period of time by the same blocklist. This suggests that, contrarily to other type of records, IP indexation is more volatile: entries tend to be indexed and removed more frequently. In Section VIII we further analyze the dynamism of records across blocklists and the update latency for the different providers.

Takeaway. We have shown that providers have different updating cycles, from multiple times per day (*e.g.,* Bambenek) to weekly (*e.g.,* Turris) to not being updated at all (*e.g.,* RedIris). Also, not all of the blocklists keep historical data. For instance, Malshare creates a new file every time that new entries get indexed. Providers must clarify their update cycles (and lifetime) so that users can crawl these lists accordingly.

VII. Overlap

Threat Intelligence benefits from information sharing across providers [38]. In fact, as a result of the open nature of our target blocklists, it is reasonable to expect a certain amount of overlap between them. In this section, we quantify the overlap between pairs of blocklists providers at the record-level, both in an aggregated fashion (*i.e.,* across our 6-month observation window) and daily. To do so, we use two complementary metrics: the overlap coefficient [39] and Jaccard's similarity index [40]. The overlap coefficient has the advantage of being equal to 1 if one of the compared sets is a subset of the other, so it will reveal the presence of potential list aggregators. Jaccard's similarity, instead, will allow us to measure how large these subsets are.

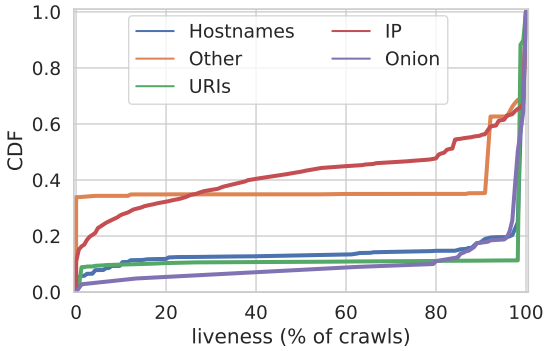


Figure 2: Record liveness from December 1st 2019 to January 31st 2020 by record type.

A. Analysis of providers

The analysis of overlap reveals strong similarities between some pairs of providers. While we find that 60% of the possible pairs of providers do not overlap at all, there are cases in which there is a significant overlap between blocklists from different providers. Appendix A shows heatmaps of the overlap coefficient and Jaccard's index per provider. In particular, the 4 lists issued by MalwareWorld have a mean overlap coefficient of at least 50% with 49% of the lists issued by 32 other providers. In this case, MalwareWorld indicates on its website that it aggregates from multiple lists covered in our study [41]. However, not all providers are so transparent about their practices. Two providers that largely overlap are CEDIA [42] and Lehigh university [43], with an overlap coefficient of 1 and a Jaccard index of 99.7% (*i.e.,* they are almost identical) but none of them acknowledges the aggregation of data from other sources. According to Table II, Bambenek—the largest provider in our list—only shares a small fraction of records with other providers. Yet, the overlap analysis reveals that they share records with the Internet Storm Center.

In the majority of cases, even if the overlap coefficient between two providers is high, the Jaccard index stays low: below 20% in 99.7% of the cases. For instance, the mean Jaccard index value between lists provided by MalwareWorld and other providers is only 1.6%.

These observations suggest the existence of two types of open blocklist providers: **aggregators** and **aggregated**. The former are lists with high number of entries that have a high overlap with the latter, but the Jaccard index is low. However, the fact that a blocklist contains a high number of records does not necessarily mean that it is an aggregator. For instance, the mean overlap coefficient between lists provided by Lashback (the third largest provider in our dataset) and other providers is only 3% (the mean Jaccard index in this case is 0.1%).

In terms of transparency, by manually checking the websites of the 8 pairs of providers that have an overlap coefficient of at least 20%, we observe that only two (MalwareWorld and Firehol Project) providers acknowledge that they aggregate information from other blocklists and list their sources explicitly.

These results suggest different levels of macroscopic interdependencies between providers: while some considerably influence each other or feed from the same providers, others are more independent. We further analyze these dependencies temporally and across providers in Section VIII.

B. Temporal evolution of the overlap

The record overlap between pairs of providers can change over time. For example, two providers might be sharing entries at some point but might later dissociate due to a change of license, the deployment of alternative data sources, or the subscription to commercial feeds. To analyze whether temporal changes occur, we compute a matrix of the overlap coefficient between every pair of blocklists grouped by provider in our dataset for every crawl. Then, we compute the difference between the matrices from two consecutive crawls and obtain its Frobenius norm [44]. We finally normalize the norm to get a result between 0 and 1, the maximum value being the norm of a matrix where all the values are ones.

The Jaccard index evolves very little over time: when grouping the blocklists by providers, the norm of the difference matrix between consecutive crawls never exceeds 4%. The overlap coefficient, instead, shows more variation but stays relatively low. When grouping the blocklists by provider, the normalized norm of the difference matrices is lower than 5% for 78% of the cases. This indicates that, regardless of the evolution of the blocklists, the overlap between them varies very little over time. In other words, when a pair of lists overlap, they tend to evolve in the same way: new entries added to one list tend to appear in others, and vice versa. For these cases, the presence of the same records in multiple blocklists cannot be used as an indication of consensus as they might come from the same source. This can cause unexpected issues down the road. For instance, if a domain gets blocklisted by mistake, this error can be propagated to multiple lists and the owner of the domain will have to contact dozens of blocklists providers to get unlisted (in case that they provide any type of contact).

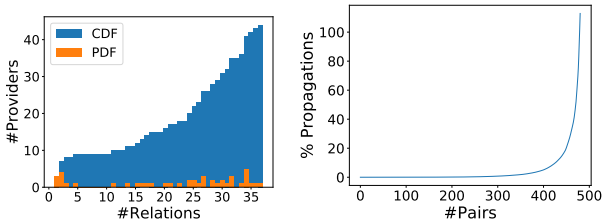
Takeaway. We have shown that several providers present a high overlap, and thus might be feeding from the same sources or from each other. Nevertheless, these overlapping providers do not clearly document that they take entries from other lists. This lack of transparency prevents researchers and end-users from making informed decisions about which blocklists better fit their needs, and might result in them using blocklists with high overlap.

VIII. Record Propagation and Influence

So far, we have observed that the total number of unique records in our dataset remains constant, but there are frequent additions and removals in specific blocklists (Section VI). We have also seen that, regardless of the fact that most providers do not inform users of their data sources (Section IV-A), we observe significant overlaps between pair of providers, and that these overlaps tend to persist in time (Section VII). In this section, we aim to understand whether this overlap is an artifact of blocklists' providers feeding from the same sources (or from each other). To do so, we take a closer look at the propagation of records across blocklists and how providers influence each other at a record level and longitudinally. We also analyze the latency of the different providers when updating their blocklists, following a similar approach to previous work [5], while also measuring the latency of record removals.

Type	#Providers	#Relations	#Propagations
IP	44	478 (96.37%)	1,050,135 (72.24%)
Hostname	10	13 (2.62%)	398,317 (27.40%)
URI	6	5 (1.01%)	5,186 (0.36%)

Table IV: Summary of the propagation of records found in the 2-month period of study



(a) Distribution of total relations per provider

(b) Distribution of the number of propagations per pair of providers

Figure 3: Analysis of relations between providers in terms of IP record propagation

Our analysis is based on computing the trajectory of each record across different blocklists. To do so, we first identify the date when a record gets indexed for the first time at each blocklist. Then, for each record r we create a timeline sequence $TS(r) = [S_{t_1}, S_{t_2}, \ldots, S_{t_N}]$, where each S_{t_i} is the set of blocklists that index r for the first time in time t_i. We consider relative timestamps, *i.e.,* $t_0 = 0$ and each subsequent t_i denotes the time units elapsed since t_0. A propagation occurs if a record is observed in two different blocklists at times t_i and t_j with $i < j$. Since we collect data every 8 hours (see Section III), a time unit corresponds to this interval.

The analysis is based on the observations in a time span of two full stable months (December 2019 and January 2020).[6] As shown in Section VII-B, the evolution of the overlap is constant across the 6-month period of our dataset, and thus it is reasonable to expect that the findings from two months can be generalized. We found a total of 1,453,638 propagations of records, all of them of type IP, URI or hostname (onions and hashes do not propagate). Table IV shows the number of providers that overlap (*i.e.,* they index the same record at some point), the number of relations (*i.e.,* pairs of providers

[6]As discussed in Section VI-B, this avoids the noise caused by the crawling infrastructure outages.

with at least one propagation between each other), and the total number of propagations for each type of record.

Hostnames dominate our dataset (see Table III) but the overlap between pairs of providers tends to be low. Manual inspection reveals that most of the propagations of domains (98.6%) are due to MalwareWorld feeding from Internet Storm Center. Similarly, 92.4% of the URLs that propagate are phishing URLs shared between PhishTank and OpenPhish, two of the providers with larger overlap.

Figure 3a shows the distribution of total relations and providers of IP-level records. More than 35 providers out of 40 (79%) have more than 10 relationships with other providers. Only 3 of them relate to just one another. This validates the results from previous sections: the sharing of IP records is common within the blocklists in our dataset, whereas URLs and hostnames are not commonly shared, except for a few providers. Therefore, in the remainder of this section we focus the analysis on IP records.

A. Latency of record changes

Previous works have studied the latency of providers when *adding* new records [5], [7]. However, these works lack a latency analysis of providers when *removing* records. Indeed, the blocklist update process should consider both data insertion and removal. This is paramount to reduce bloating when a record is no longer needed, and also to correct and prevent the uncontrolled dissemination of false positives. Accordingly, we investigate the latency of providers when both indexing and de-indexing records from their blocklists.

1) Latency of record additions: We analyze the latency of providers adding records that have been observed previously in a different blocklist. To do so, we divide the entire timeline of all records into regular relative slots, where the first slot represents the first crawl where the record is observed. Then, we count the number of times that a blocklist indexes a record in each slot. Figure 4 shows the ratio of records that are indexed at each time slot for each of the providers with more than 1k propagations. To ease visualization, we group slots using different intervals (see x-axis labels), which allows us to analyze with low granularity the temporal dynamics in the few hours after its first indexing.

One key observation is that once a new IP is indexed by any of the providers, most of the others update their blocklists fast. This suggests that the update process for these providers is automated and new records are unlikely to go through any additional checkings. However, some providers are slower when adding entries. For example, in the case of Turris, 41% of its records are indexed in the first slot, whereas 25% are in slot T8 (*i.e.,* a delay of 6 days since their first observation). Turris claims to gather data from their own routers and releases records weekly [45], due to the IPs being first analyzed and classified according to observed behavior. This case evidences the trade-off of such an strategy: while some data is unique (and possibly contains fewer false positives than non-curated items), it takes longer to reach users.

Record addition can be caused by re-offending entities, *i.e.,*an IP being used for different purposes at different times.

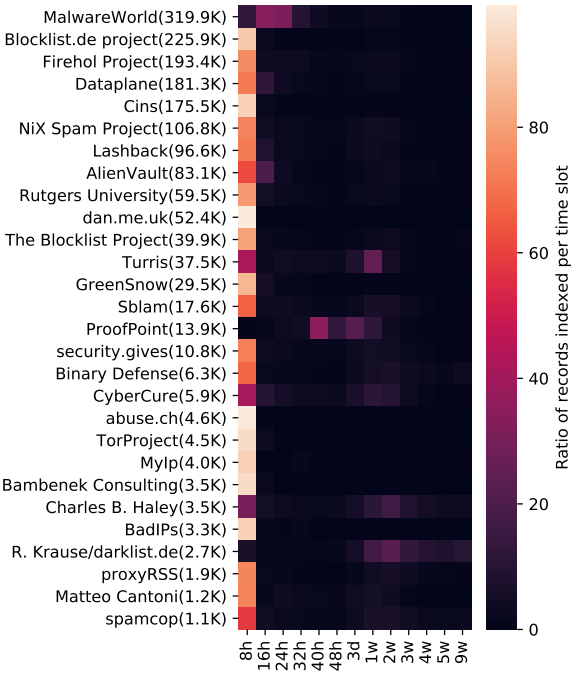


Figure 4: Addition latency of each blocklist provider for IP records. Numbers in parentheses show the total of IP records shared.

When a record which is already present in a list and is later indexed in a different list, this can mean that the IP is being re-used for a different purpose, or that the provider of the second list took longer to detect the same misbehavior. While difficult, we attempt to differentiate between these two cases by looking for IPs that appear in a new list at least 1.5 months after their first appearance in a different lists. This allow us to find out only 0.5% IPs with the potential to be considered as re-offending. By manual analysis, we find cases where the IP is a Tor Exit node. IPs from Exit nodes are shared by different users, which can lead to them being indexed at different times (we will further investigate how this can lead to the IP receiving multiple lables in § IX). In other cases, we confirm that the IP is actually being used for different purposes at different times (concretely, for brute-forcing at first, and for spam 2 months afterwards). In these cases, these IPs might correspond to bulletproof hosting servers, *i.e.,*hosts that are not easy to track and take down (and those remain operative even if they are blocked), and which are usually traded in underground markets [46].

2) Latency of record removal: We assume that if a provider A indexes a record first, and then a provider B indexes the same record, the removal of the record in A should also be followed by B. Yet, providers might implement different sanitizing and filtering processes, and they might opt to keep records in their blocklists according to their own policies (*e.g.,* to have an historical perspective).

First, we count additions and removals per blocklist and then we track the propagation of these events across blocklists and at different times. For each record, we refer to the blocklist P that indexes it first as the *preceding* blocklist, and blocklists

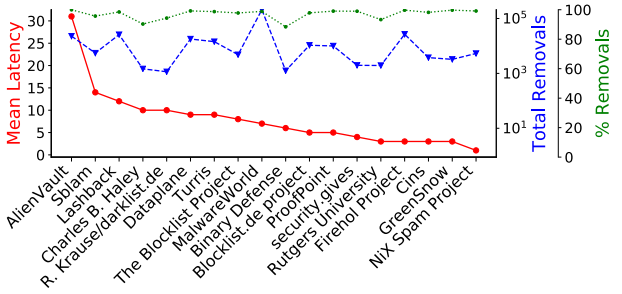


Figure 5: Removal latency (mean for periods of 8 hours), total removals and percentage per provider

$S_1, S_2, \ldots, S_N$ indexing it at a later point as *successors*. We analyze whether removals in P are followed by removals in the successors and, if so, compute the latency.

Figure 5 shows the total number of records removed, the percentage with respect to the number of records added, and the average latency for each provider. For clarity, we only include providers with more than 1k records in common with preceding blocklists. Overall, all the providers remove records consistently (all providers have a ratio of removals higher than 88%), which indicates that either they mirror their preceding blocklists or they implement similar policies.

This high ratio of removals suggests that providers actually aim at cleaning-up their lists. However, we observe substantial differences in the policy across providers. For example, AlienVault and Firehol are two providers that respectively removed around 22k and 26k records that were previously removed in their preceding blocklists. This means that these two providers regularly update their blocklists. However, AlienVault takes, on average, 31 crawls (around 10 days) to remove a record, whereas Firehol takes just 3 (around 1 day). According to AlienVault, the provided list of IPs should be used as a reputation list rather than an actual blocklist [47]. However, other entities feed from AlienVault for blocking purposes. Thus, IPs that do not longer qualify as malicious and that are not removed will propagate to these entities and might cause unwanted disruptions [4]. Other providers update their blocklists rather frequently and fast. For instance, the Nix Spam Project removes records frequently (more than 5.3k) and quickly (at every crawl on average, *i.e.,* 8 hours). The Nix Spam Project provides a quick form to request removals, which suggests that it proactively aims at removing false positives from their lists.

B. Correlation among blocklists

In Section VII we showed that the relationships and the influence between blocklists are consistent over time. We next investigate providers that are correlated over time, *i.e.,* that evolve similarly. The distribution of propagations across pairs of providers (Figure 3b) indicates that a minority of pairs account for most of the propagations, with a substantial amount of providers having very low overlap. Specifically, 23 out of 478 pairs (around 5%) account for more than 80% of the propagations. This suggests that some providers accumulate most of the propagations, *i.e.,* they either feed from each other

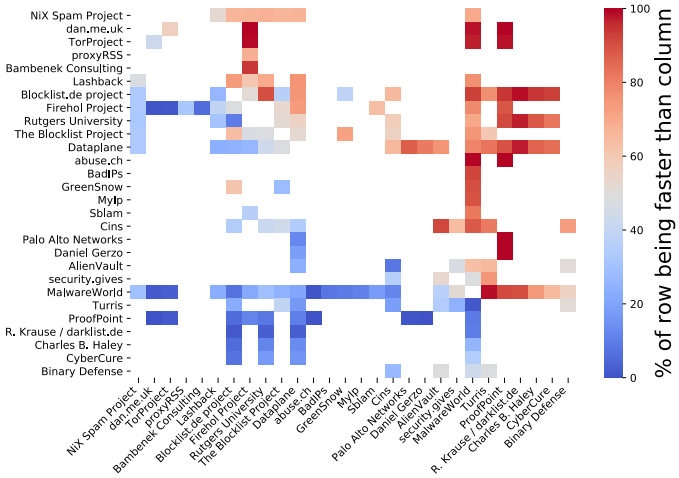


Figure 6: Speed comparison of providers. Each P_{ij} cell is the percentage of i adding a record before j

or use the same sources. In order to compare each pair of providers (i, j), we calculate three metrics. First, λ_{ij} accounts for the number of times that the two providers indexed the same record simultaneously. Second, β_{ij} counts the number of times that i was faster than j, meaning that i indexed the same record before j did. Finally, α_{ij} counts the number of times that i was slower than j.

We observe a cluster of various providers with more than 1k records indexed simultaneously, suggesting that these feed from the same sources. This includes, among others, MalwareWorld, AlienVault, Firehol, and ProofPoint. Figure 6 provides a comparison between providers with more than 1k common records in terms of speed. Each cell P_{ij} represents the percentage of times that the provider i indexed a record before provider j (*i.e.,* β_{ij} divided by $\beta_{ij} + \alpha_{ij}$). The analysis reveals that two providers of Tor exit nodes (*dan.me.uk* and *TorProject*) are always faster than their peers. Moreover, 83% of their common IPs were indexed simultaneously, and in most cases (93%) these records were unknown for the other providers. This suggests that these two providers give the most updated information regarding Tor exit nodes in our dataset. In fact, *Dan.me.uk*, claims on its documentation that it fetches information from *TorProject* every 30 minutes, explaining this observation [48].

Also, MalwareWorld is in general slower than its feeds [49], but it also has various records indexed simultaneously with them. A potential reason is the fact that we use time units of 8 hours due to our crawling methodology which disallow us to observe differences in record addition smaller than this time span. Interestingly, MalwareWorld is sometimes faster than its feeds. For example, in 68% of the times it was faster than Turris and in 30% they were simultaneous. Indeed, only 251 of their common IPs (0.9%) were first indexed by Turris. As discussed before, this confirms that the sanitized IPs released by Turris were included in other feeds from MalwareWorld.

Takeaway. Systematic propagation of records from one list to another can be an artifact of both lists feeding from the same sources with different update times, but it can also mean that

the second list is aggregating records from the first one. We also note that these might be due to re-offending entities, but the percentage of these is negligible for our period of study. Even if we have not measured accuracy of the content indexed by each blocklist, we hypothesize that aggregators feeding from many other providers in an automated and rapid way might get a broader coverage, but can also include outdated records and false positives. While IPs propagate fast across blocklists, some providers take longer to include them than others. We argue that these providers likely curate their records in an attempt to reduce false positives. This behavior also prevents users from taking the presence of the same records in multiple blocklists as an indication of consensus, as some blocklists potentially copy from each other without verifying or validating the soundness of their sources.

IX. The problem of labeling

As discussed in Section IV-A, one of the problems with blocklists is the lack of transparency regarding the nature of their contents. In this section, we look at several ways in which users can infer why a given entry was blocklisted and the challenges and problems associated to the labeling of records.

A. Provider-defined labels

We find 358 different vendor-provided labels for 73% of the blocklists. This confirms that (1) over 25% of the providers do not clearly document the nature of their blocklists; and (2) the lack of an established classification framework for blocklists has resulted in providers defining whatever custom labels they deem appropriate to tag their records (*e.g.,* "nefarious-activity-abuse"). Some providers opt to have bigger lists that contain entries related to a general problem (*e.g.,* "Malware"), while others have more lists broken down into specific types of attacks (*e.g.,*"Mirai" or "VoIP attacks"). This results in a big constellation of labels that can make it hard for users to make sense of what exactly the content of each list is.

These inconsistencies—which bear some resemblance with the classic malware labeling problem [50]—might be caused by different vendor sensitivities or by the need to address a particular market. For instance, while an email-protection vendor labels an IP block as spam [51], a phishing-oriented source might refer to it as phishing [52]. As a result, blocklist consumers might not be aware of the subtle differences or similarities between such labels and their actual purpose, soundness, and accuracy. Differences in labels can affect the ability from users to judge whether a given blocklist or record is relevant to the specific blocking needs of a system. This can also result in unwanted side-effects, as users might end up blocking entries that are not necessarily malicious or dangerous for their application. Furthermore, labeling inconsistencies make it hard for users to compare and merge together lists from different providers, as understanding which labels are equivalent to each other might not be obvious. This issue can be further aggravated when the blocklist aggregation process is done automatically.

B. Measuring labeling disagreements

This subsection measures how different approaches to data labeling, combined with the propagation of records from one list to another, can result in the same entry being labeled differently by different providers. To do so, we need to reduce the number of labels to a manageable figure, as it would be hard to draw accurate conclusions over 358 different labels. We follow a clustering approach and made a manual effort to normalize all observed labels into the following 8 meta-labels:

Malware (39% of blocklists). Any form of engagement in malware distribution, without distinction of the type of malicious software involved.

Attack (15%). Sites engaged in (possibly various forms of) malicious activities, but not exclusively restricted to malware.

Miscellanea (3%). Blocklists that serve multiple purposes and therefore do not clearly fall in a single category.

Anonymity (1%). Sites or services providing some form of source anonymity for communications.

Phishing (0.9%). Sites engaged in phishing activities regardless of the specific technique.

Spam (0.6%). Distribution of unwanted messages regardless of the purpose or the channel (email, social networks, *etc.*).

Anti-tracking (0.2%). Blocklists used to block advertising or tracking services. Also includes ad-blocking.

Bad reputation (0.2%). Sites with a bad reputation score, though no specific reason is known.

For the remainder 40% of blocklists, the provider does not supply an original label or enough information to classify the blocklist into any of the above categories. We tag them as **Not defined**. Each one of these normalized labels refers to a different underlying phenomenon explaining the nature of the blocklist, such as an abuse type (*e.g.,* Malware or Spam) or a relevant signal for filtering (*e.g.,* Anonymity). Nevertheless, this normalization process relies on the manual inspection and codification of scarce, incongruent, and often confusing terms.

Label changes. We investigate how often a given record appears later on in a blocklist with a different meta-label than the original label observed. We find that 14.8% of all entries appear in blocklists with at least two different labels. For these entries, the median number of different labels that they have is 2, with some records being indexed by blocklists accounting together for as many as 6 different labels.

Figure 7 shows how often a record label changes between two specific meta-labels. We note that we exclude pairs of labels that represent less than 0.05% for visibility reasons. Most of these changes are caused by differences in classification strategies between different providers. For instance, we find that the most common cases of label changes are entries that are labeled as "Miscelanea" changing to more specific labels such as "Attack", "Malware" or "Anonymity". This can be a direct result of providers using different strategies to label threats, and also of some providers that maintain lists that aggregate entries from different types of threats, while others are much more specialized. We also find that it is common for entries labeled as "Attack" to later appear on blocklists labeled as "Malware", "Anonymity" or "Bad Reputation".

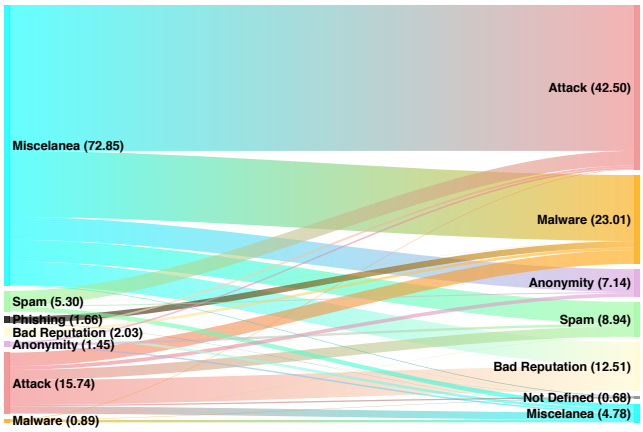


Figure 7: Label changes of entries across blocklists of different categories

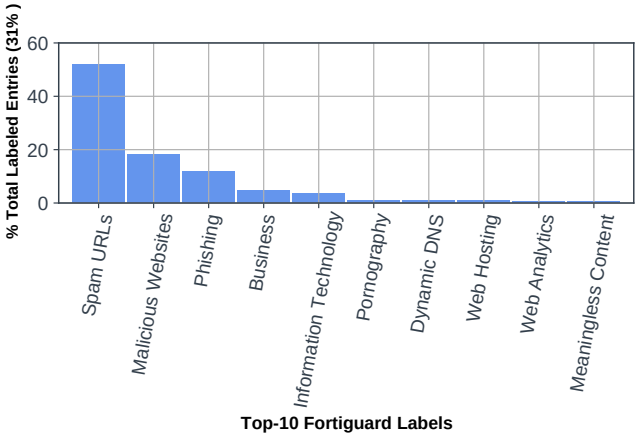


Figure 8: Top-10 most common labels in hostnames, URIs IPs according to Fortiguard. The percentage is relative to the number of entries labeled by Fortiguard.

While we can not automatically and systematically study why a given record changes labels, we manually look at the cases with a higher number of labels. Namely, there are 9 IPs—all belonging to TOR exit nodes—that have 6 different labels. IPs from exit nodes can be used by many different users that might engage in different type of activities (illicit or not). Therefore, we believe that these label changes are a direct result of the different potentially malicious uses that TOR users are making of this exit node (*e.g.,*a user can use this exit node to launch an attack or to distribute malware, while another one uses it to distribute spam).

Label changes like those observed in our measurement can affect negatively the effectiveness of blocklisting as a security strategy, for instance, by aggravating the problem of incorrect blocking. In addition, it forces end-users to put in place a reconciliation mechanism to unify labels when different decisions are to be made over different categories of entries.

C. Label comparison with external data sources

One way in which end-users can try to overcome the limitations of labels based on provider information is to add extra information from external sources. We extended our analysis to analyze every hostname, URI and IP indexed in our dataset with Fortiguard [29], a commercial domain classification and threat intelligence service specialized in content filtering. Following the recommendations of Vallina *et al.* [53], we use Fortiguard for three main reasons: 1) its high coverage (*i.e.,*the number of entries for which it provides a meaningful label); 2) its label constellation, which includes specific labels for malicious services; and 3) its ability to label IP addresses. While Fortiguard is able to provide a meaningful label for some IP addresses, IP-level blocking is not really what the service is built for. To the best of our knowledge, there is no free and unlimited service that provides information about the type of service behind an IP. This translates on Fortiguard's coverage on IPs (1%) being much lower than for hostnames (46%) and URIs (99%), and in consequence reducing the general coverage to 31% of the total entries.

Figure 8 shows the 10 most popular labels for all entries present in blocklists (31% of the total ones). We observe that Fortiguard categorizes as malicious 83% of the labeled entries, highlighting the presence of *Spam URLs* (52%), *Malicious Websites* (18%), *Phishing* (12%), and *Dynamic DNS* (1%).

Labeled Blocklists. One of the main problems with dedicated blocklists, defined as those created to block a specific type of content, is to quantify if the indexed entries can be defined with the label assigned to the blocklist (*e.g.,*spam domains in a blocklist for spam). To this end, we aggregated all the entries indexed on blocklists equally labeled using the meta-labels defined in Section IX-B. For some categories, we observe that adding Fortiguard data can increase the confidence of the original label, as most domains fall in a category equivalent to this label. For instance, we observe that 55% of the entries indexed on phishing blocklists are labeled as phishing. Similarly, in Malware blocklists, 32% and 14% of the labeled entries (77% of the total entries) are labeled as *Malicious Websites* and *Phishing* respectively. In some cases, this external data contradicts the original label, as in the case of the blocklists labeled as spam, for which less than 15% of the indexed entries are classified as malicious. However, in this particular case, Fortiguard only provides a meaningful label for 1% of the entries. Finally, in the case of blocklists with the *Not defined* super category (i.e., those that we were not able to classify), we observe that 75% of the labeled entries are classified as potential malicious services, with 49% of them labeled as *Malicious Websites*.

This experiment highlights the fundamental issues that exist in the open blocklist ecosystem. It also shows how relying on external data sources to the cyberdefense pipeline can help blocklist consumers in two ways: (1) if the provider label agrees with the external label, this increases the confidence on the record's correct classification (and mitigate undesired side-effects); and (2) records in blocklists where the provider does not include enough information to label the entries can benefit from the use of an external data source such as Fortiguard to make up for the lack of a classification.

Takeaway. We have shown that most blocklist providers follow very different strategies for labeling their data, finding a total of 358 labels in our dataset. Furthermore, even

when applying an artificial grouping to try and reduce the dimensionality of the problem, the lack of information given by a large number of providers (40%) alongside the high number of entries that present more than one label during our study (14.8%) make it extremely difficult to rely on provider info for labeling entries. Finally, we showed that adding external data sources (such as Fortiguard) can help increasing confidence on the original labels, or providing a classification when the provider does not. Nevertheless, the low coverage of these external services (31%), particularly for IP-level records, prevents this from being a complete solution to the problem.

X. Discussion

Our analysis reveals that, despite their popularity, blocklists present poorly understood dynamics. Their update rates depend on the provider and the type of records that they index, and the lack of transparency and homogeneity in the ecosystem opens critical operational and research challenges. In this section, we discuss the most relevant implications of our study and aims to provide recommendations to providers, researchers and practitioners.

Recommendations for providers. We believe that there are several steps that providers can take to improve the ecosystem of open source blocklists. Mainly, all providers should make an effort to be more transparent. Blocklists should provide users with a clear explanation of the type of records that they contain, how they are gathered, and whether they take care of removing false positives. Then, users could choose what is the list that better suits their needs, reducing the risks of over and under-blocking. We also believe that providers should come up with a common way to label entries. One way would be to define a common taxonomy, with clear rules for including a given record into a specific category. Efforts by the antivirus community such as XARF and MARF [54], [55] are a good example on how these common taxonomies can be helpful for the ecosystem. Another possibility is to move towards more standardized taxonomies such as MISP. MISP galaxy [56] provides a common structure that can be used to report malicious entries, helping companies and users process blocklists without the need to create ad-hoc parsers for each provider. Reaching a consensus on labeling practices and blocklists format would increase the overall applicability of these lists and ease the process of blocklist comparison, integration and maintenance.

Recommendations for blocklist users. While providers' opacity makes it hard, users should try to understand as much as possible from a given list before using it. In this paper we have shown how compiling records to block from several lists is not always helpful, as often this can lead to over-blocking and even under-blocking, since lists can be feeding off each other. Whenever possible, users should move towards those lists that are clear about how records are collected, as well as about their data sanitization and record removal processes. When users can not find a transparent list that fulfills their needs, we advise them to monitor the blocklist for a period of time before including it as part of their blocking infrastructure. This will allow them to better understand the type of entries indexed on every blocklists, as well as their dynamics, in order to implement blocking mechanisms accordingly to their needs.

XI. Conclusions and Future Work

In this paper, we empirically studied the transparency and dynamics of the ecosystem of open blocklists providers, gathering a dataset of 2,093 from 69 during a time period of approximately 6 months. We look at the synergies between blocklists, finding a high overlap between specific providers. We find that addition and removal of records is often propagated across those providers that have a high overlap. Interestingly, we find that other studies that include commercial blocklists reach different conclusions in terms of the overlap between providers [5], further proving that there are notable differences between both ecosystems. We also show that it is difficult to understand what is the content of these lists, as providers often fail to accurately label their data and document their processes. As a result, the same record can be labeled differently by distinct providers.

Gathering ground truth to measure the accuracy of blocklists is challenging [5]. While we attempt to use external data sources in our analysis as a form of ground truth, we nonetheless observed contradictions and mislabeling with respect to other blocklists. Establishing a tie breaker in this ecosystem requires an effort among the actors to unify and automate their classification procedures and criteria. This poses new challenges to the community (especially the open source one) in order to achieve new collaborative and open models. This lack of ground truth disallows us to understand whether IPs are re-offending (*i.e.,*being reused for a different purpose at a different times) or not (*i.e.,*being used for several ones at the same time). In general, classifying which IPs present potentially malicious behaviors is a complex (and unsolved) problem that falls out of the goals of this study.

In summary, future approaches to enhance and standardize threat intelligence feeds should emphasize transparency and unifying criteria to ease their compilation, vetting and maintenance, which will improve their effectiveness.

Acknowledgements

We would like to thank the anonymous reviewers and Dr. Qi Li for their valuable feedback. This research has been partially funded by the German Federal Ministry of Education and Research (AIDOS Nº16KIS0976); the Region of Madrid (EdgeData-CM NºP2018/TCS-4499); the Region of Madrid co-financed by European Structural Funds ESF and FEDER (CYNAMON-CM NºP2018/TCS-4566); the Spanish Government (DiscoEdge NºTIN2017-88749-R and ODIO NºPID2019-111429RB-C21/PID2019-111429RB-C22); and Google. The opinions on this article are those of the authors and not of the funding bodies.

References

[1] "About the Spamhaus Project," https://www.spamhaus.org/organization/, [Online; accessed 28-May-2020].

[2] C. Wagner, A. Dulaunoy, G. Wagener, and A. Iklody, "Misp: The design and implementation of a collaborative threat intelligence sharing platform," in *Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security*. ACM, 2016.

[3] "Wikipedia:advice to users using tor," https://en.wikipedia.org/wiki/Wikipedia:Advice_to_users_using_Tor, [Online; accessed 11-Oct-2020].

[4] AlienVault, "How to remove an ip from the reputation list?" https://success.alienvault.com/s/question/0D53q00009ghcN6CAI/how-to-remove-an-ip-from-the-reputation-list, 2019, [Online; accessed 23-May-2020].

[5] V. G. Li, M. Dunn, P. Pearce, D. McCoy, G. M. Voelker, S. Savage, and K. Levchenko, "Reading the tea leaves: A comparative analysis of threat intelligence," in *28th USENIX Security Symposium (USENIX Security 18)*, 2019.

[6] A. Pitsillidis, C. Kanich, G. M. Voelker, K. Levchenko, and S. Savage, "Taster's choice: a comparative analysis of spam feeds," in *Proceedings of the 2012 Internet Measurement Conference*. ACM, 2012.

[7] L. Metcalf and J. M. Spring, "Blacklist ecosystem analysis: Spanning jan 2012 to jun 2014," in *Proceedings of the 2Nd ACM Workshop on Information Sharing and Collaborative Security*. New York, NY, USA: ACM, 2015.

[8] "Dataset sample," https://drive.google.com/file/d/1ln_v3DOxb46BSowfXFXettOokA0OhCV/view, [Online; access 1-June-2020].

[9] A. Vastel, P. Snyder, and B. Livshits, "Who filters the filters: Understanding the growth, usefulness and efficiency of crowdsourced ad blocking," *arXiv preprint arXiv:1810.09160*, 2018.

[10] A. Razaghpanah, R. Nithyanand, N. Vallina-Rodriguez, S. Sundaresan, M. Allman, and C. K. P. Gill, "Apps, trackers, privacy, and regulators," in *25th Annual Network and Distributed System Security Symposium, NDSS*, vol. 2018, 2018.

[11] Z. Yu, S. Macbeth, K. Modi, and J. M. Pujol, "Tracking the trackers," in *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016.

[12] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri, "Know your neighbors: Web spam detection using the web topology," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007.

[13] N. Jindal and B. Liu, "Review spam detection," in *Proceedings of the 16th international conference on World Wide Web*, 2007.

[14] V. Krishnan and R. Raj, "Web spam detection with anti-trust rank." in *AIRWeb*, 2006.

[15] P.-A. Chirita, J. Diederich, and W. Nejdl, "Mailrank: using ranking for spam detection," in *Proceedings of the 14th ACM international conference on Information and knowledge management*, 2005.

[16] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and C. Zhang, "An empirical analysis of phishing blacklists," 2009.

[17] M. Kührer and T. Holz, "An empirical analysis of malware blacklists," *Praxis der Informationsverarbeitung und Kommunikation*, 2012.

[18] T. Yagi, J. Murayama, T. Hariu, S. Tsugawa, H. Ohsaki, and M. Murata, "Analysis of blacklist update frequency for countering malware attacks on websites," *IEICE Transactions on Communications*, 2014.

[19] T. Tran Phuong, M. Tokunbo, U. Jumpei, Y. Akira, M. Kosuke, and K. Ayumu, "Large-scale analysis of domain blacklists," in *The Eleventh International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2017)*. KDDI Research, Inc., 2015.

[20] S. Sinha, M. Bailey, and F. Jahanian, "Shades of grey: On the effectiveness of reputation-based "blacklists"," in *2008 3rd International Conference on Malicious and Unwanted Software (MALWARE)*. IEEE, 2008.

[21] S. Ramanathan, J. Mirkovic, and M. Yu, "Blag: Improving the accuracy of blacklists," in *25th Annual Network and Distributed System Security Symposium, NDSS*, vol. 2020.

[22] A. G. West, A. J. Aviv, J. Chang, and I. Lee, "Spam mitigation using spatio-temporal reputations from blacklist history," in *Proceedings of the 26th Annual Computer Security Applications Conference*, 2010.

[23] S. Ramanathan, A. Hossain, J. Mirkovic, M. Yu, and S. Afroz, "Quantifying the impact of blacklisting in the age of address reuse," in *Internet Measurement Conference 2020 (IMC)*. ACM.

[24] "Periodical report of the threat intelligence market," https://www.grandviewresearch.com/press-release/global-threat-intelligence-market, [Online; accessed 07-April-2019].

[25] "Threat intelligence market by solution," https://www.marketsandmarkets.com/pdfdownloadNew.asp?id=150715995&gclid=EAIaIQobChMIn_qrkb7I4AIVU4fVCh3bNg-sEAAYAyAAEgKtA_D_BwE, [Online; accessed 07-April-2019].

[26] "Threat intelligence market research report," https://www.marketresearchfuture.com/reports/threat-intelligence-market-4110, [Online; accessed 07-April-2019].

[27] "MISP — Open Source Threat Intelligence Platform," https://www.misp-project.org/, [Online; access 11-May-2020].

[28] "Maltrail," https://github.com/stamparm/maltrail/, [Online; accessed 26-April-2019].

[29] "FortiGuard Labs," http://fortiguard.com/, [Online; accessed 18-May-2020].

[30] "I-blocklist bluetack level 2," https://www.iblocklist.com/list.php?list=uwnukjqktoggdknzrhgh, [Online; accessed 13-Oct-2020].

[31] "Feodo tracker's aggressive listing," https://feodotracker.abuse.ch/blocklist/, [Online; accessed 09-April-2019].

[32] "Spamcop's aggressive listing," https://www.spamcop.net/bl.shtml, [Online; accessed 09-April-2019].

[33] B. Farinholt, M. Rezaeirad, D. McCoy, and K. Levchenko, "Dark matter: Uncovering the darkcomet rat ecosystem," in *Proceedings of The Web Conference 2020*, 2020.

[34] "Backdoor.NJRat," https://blog.malwarebytes.com/detections/backdoor-njrat/, [Online; accessed 11-May-2020].

[35] S. Kumar, S. Agarwala, and K. Schwan, "Netbus: A transparent mechanism for remote device access in virtualized systems," *Network*, 2008.

[36] "All cybercrime ip feeds by firehol," http://iplists.firehol.org/#about, [Online; accessed 07-April-2019].

[37] "Tor2web: Browse the Tor Onion Services," https://www.tor2web.org/, [Online; accessed 20-March-2020].

[38] R. Garrido-Pelaz, L. González-Manzano, and S. Pastrana, "Shall we collaborate? a model to analyse the benefits of information sharing," in *Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security*, 2016.

[39] M. Vijaymeena and K. Kavitha, "A survey on similarity measures in text mining," *Machine Learning and Applications: An International Journal*, 2016.

[40] P. Jaccard, "The distribution of the flora in the alpine zone," *New phytologist*, 1912.

[41] "MalwareWorld — Checker," https://www.malwareworld.com/, [Online; access 10-May-2020].

[42] "CEDIA," https://cedia.org.ec/, [Online; access 29-May-2020].

[43] "Malware Prevention through DNS Redirection," http://malwaredomains.lehigh.edu/, [Online; access 29-May-2020].

[44] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.

[45] "Project:Turris. GreyList," https://project.turris.cz/en/greylist.

[46] S. Alrwais, X. Liao, X. Mi, P. Wang, X. Wang, F. Qian, R. Beyah, and D. McCoy, "Under the shadow of sunshine: Understanding and detecting bulletproof hosting on legitimate service provider networks," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017.

[47] AlienVault, "Can i use the otx ip reputation list as a blocklist?" https://success.alienvault.com/s/article/Can-I-use-the-OTX-IP-Reputation-List-as-a-blocklist, 2019, [Online; accessed 23-May-2020].

[48] "Tor node list," https://www.dan.me.uk/tornodes, [Online; accessed 13-Oct-2020].

[49] "MalwareWorld — List," https://malwareworld.com/textlists/blacklists.txt, [Online; access 10-May-2020].

[50] M. Sebastián, R. Rivera, P. Kotzias, and J. Caballero, "Avclass: A tool for massive malware labeling," in *International Symposium on Research in Attacks, Intrusions, and Defenses*. Springer, 2016.

[51] "Sblam spammer ips," https://sblam.com/, [Online; accessed 09-April-2019].

[52] "Openphish phishing filter," https://openphish.com/, [Online; accessed 09-April-2019].

[53] P. Vallina, V. Le Pochat, A. Feal, P. Marius, J. Gamba, T. Burke, H. Oliver, T. Juan, and N. Vallina-Rodriguez, "Mis-shapes, mistakes, misfits: An analysis of domain classification services," in *ACM Internet Measurement Conference 2020 (IMC)*. ACM.

[54] "eXtended Abuse Reporting Format," https://www.abusix.com/xarf.

[55] "Messaging Abuse Reporting Format," https://datatracker.ietf.org/wg/marf/documents/, [Online; accessed 1-Jun-2020].

[56] "Misp galaxy," https://github.com/MISP/misp-galaxy, [Online; accessed 16-Mar-2021].

APPENDIX A
OVERLAP

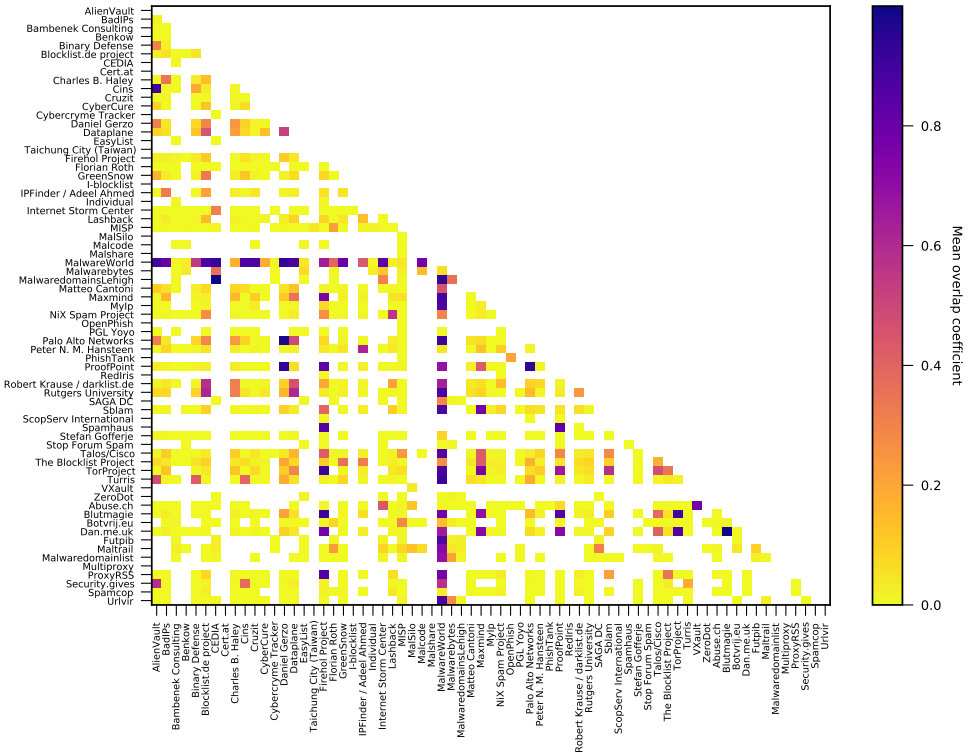


Figure 9: Mean overlap coefficient between lists when grouped by providers

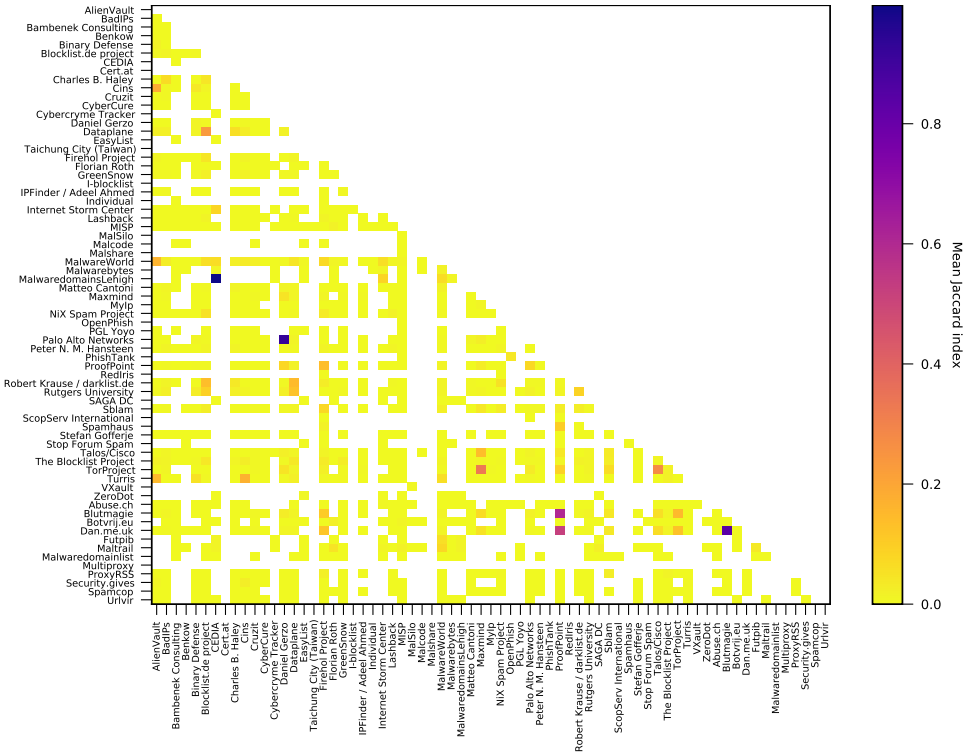


Figure 10: Mean Jaccard index between lists when grouped by providers

Álvaro Feal is a PhD student working at IMDEA Networks Institute under Prof. Narseo Vallina-Rodriguez's supervision. His research revolves around analyzing privacy threats in the mobile and web ecosystem using static and dynamic analysis techniques as well as network measurements. He has published his research in different venues such as the IEEE Symposium on Security and Privacy, USENIX Security, ACM IMC, PETS Symposium, IEEE ConPro, and CPDP.

Pelayo Vallina is enrolled in IMDEA Networks Institute as Ph.D. Student working at Global Computing Group. His research interests fall in the area of web privacy, regulatory compliance. and social networks. He has published in international peer-reviewed conferences such as ACM IMC (2019 and 2020) and WWW(2019). His works in the area of web privacy have received the attention of the media like El Pais.

Julien Gamba is a PhD student in the Internet Analytics Group at the IMDEA Networks Institute. His research revolves around user's security and privacy in Android devices. In his work, Julien uses both static and dynamic analysis, as well as other techniques specifically designed to understand the behavior of mobile applications. Recently, Julien was the first author of the first large-scale analysis of the privacy and security risks of pre-installed software on Android devices and their supply chain, which was awarded the Best Practical Paper Award at the 41st IEEE Symposium on Security and Privacy. This study was featured in major newspaper such as The Guardian (UK), the New York Times (USA), CDNet (USA) or El País (Spain). Julien was also awarded the ACM IMC Community Contribution Award in 2018 for his analysis of domain ranking services, and was awarded the NortonLifeLock Research Group Graduate Fellowship, the Google PhD Fellowship in Security and Privacy and Consumer Reports' Digital Lab fellowship.

Sergio Pastrana is Visiting Professor at Universidad Carlos III de Madrid, where he teaches various courses on cyberdefense and computer security. His research interests focus on different areas of security and privacy, including the measurement and analysis of the socio-technical factors and human aspects of cybercrime. He has published in top conferences such as WWW, IMC or RAID, and also in various international journals.

Antonio Nappa is an experienced researcher doing industrial and academic bleeding-edge research-to-product in different fields which range from computer security, privacy and cryptography to quantum computing, their definition and security model.

Oliver Hohlfeld is a professor of computer science and heads the Chair of Computer Networks at Brandenburg University of Technology (BTU). Before he was at RWTH Aachen University, TU Berlin, and Deutsche Telekom Laboratories. In his research, he uses a data-driven approach including user studies, empirical network measurement, and machine learning to understand and improve Internet performance and security.

Narseo Vallina-Rodriguez is an Assistant Research Professor at IMDEA Networks and a Research Scientist at the Networking and Security team at the International Computer Science Institute (ICSI) at the University of Berkeley, USA. Narseo is also a co-founder of AppCensus Inc. Narseo's research interests fall in the broad areas of network measurements, privacy, and mobile security. His research has been awarded with best paper awards at the 2020 IEEE Symposium on Security and Privacy (S&P), USENIX Security'19, ACM IMC'18, ACM HotMiddlebox'15, and ACM CoNEXT'14 as well as several industry grants and awards (*e.g.,*Google Faculty Research Awards, DataTransparencyLab Grant, and Qualcomm Innovation Fellowship) and he has been the PI of several NSF, NSA, H2020, and Spanish projects. His work in the mobile security and privacy domain has influenced policy changes and security improvements in the Android platform while his research on the privacy and security risks of pre-installed Android applications has received the AEPD Emilio Aced Award and the CNIL-INRIA Privacy Protection Award, both in 2020. He is also the recipient of the IETF/IRTF Applied Networking Research Award in 2016 and the Caspar Bowden Award in 2020.

Juan Tapiador is Professor of Computer Science at Universidad Carlos III de Madrid, Spain, where he leads the Computer Security Lab. Prior to joining His research interests include binary analysis, systems security, privacy, surveillance, and cybercrime. He has served in the technical committee of conferences such as USENIX Security, ACSAC, DIMVA, ESORICS and AsiaCCS. He has been the recipient of the UC3M Early Career Award for Excellence in Research (2013), the Best Practical Paper Award at the 41st IEEE Symposium on Security and Privacy (Oakland), the CNIL-Inria 2019 Privacy Protection Prize, and th e 2019 AEPD Emilio Aced Prize for Privacy Research. His work has been covered by international media, including The Times, Wired, Le Figaro, ZDNet, and The Register.