Block and Roll: A Metric-based Evaluation of Reputation Block Lists

Siôn Lloyd[1], Carlos Hernandez-Gañán[1] and Samaneh Tajalizadehkhoob[1]

[1] *ICANN*

Abstract

Reputation Block Lists (RBLs) serve as a common defense mechanism against harmful and unwanted internet content. These lists contain the IP addresses, domain names, or full URLs of known spam sources, phishing, malicious sites or other unwanted content. By using RBLs, internet service providers, email providers, and other organizations can effectively safeguard their users from online threats. They are also used for more academic research and as training sets for machine learning models. To help evaluate and understand the effectiveness of RBLs, this paper covers a set of metrics that can be used to evaluate and characterize them. These metrics include RBL focus, mechanics, metadata, volume, overlap, timeliness, and churn. We categorise the metrics into four groups: a general description; metrics that can be directly measured; metrics that can be indirectly measured and metrics that can only be discovered second-hand. When it comes to RBLs there is no “one size fits all”. We argue that understanding the strengths and weaknesses of any one RBL, or set of multiple RBLs, is key to getting a good fit for a particular use-case. To maximize the benefit of RBLs, we suggest combining two or more to get a fuller picture than can be provided by any single RBL.

Keywords

DNS abuse, blocklists, phishing, malware

1. Introduction

Domain name and IP address reputation lists have been used for many years as a way to identify and block potentially harmful or unwanted traffic on the internet. The earliest known reputation list was created by Paul Vixie in the 1990s, and was called the “Real-time Blackhole List” or RBL [1]. This list contained the IP addresses of known spam sources and was used by mail servers to block incoming email from those sources. Over time, similar lists were created for other types of online activities, such as domain or URL reputation lists for identifying malicious or phishing websites, and IP address reputation lists for identifying sources of malware or other online threats. Today, these lists are widely used by internet service providers, email providers, and other organisations to help protect their users from online threats. They continue to evolve and improve as new threats emerge and new technologies are developed to combat them.

We refer to these sources as “Reputation Block Lists” or RBLs, others may call them by slightly different names like “threat intelligence”, “security feeds”, “abuse feed” or similar. They can contain different identifier types: domain names, IP addresses or full URLs, and in many cases a mixture of two or more identifier types. They can also specialise in particular threat types, like spam, phishing, malware, *etc.*; or they may contain a mixture of multiple threat types. They can differ in collection methodology, licensing, distribution method, intended use and almost every other conceivable way.

There are many examples of RBLs being used in many different scenarios, some more obvious than others, for example services like google safe browsing[1] can be thought of like an RBL protecting a browser user from known phishing sites. The academic community also utilises RBLs to understand the current and historical reputation of domain names in various types of analysis, to measure security threat concentrations within the internet intermediaries such as TLD, registry, registrars or hosting providers and finally to assess mitigation strategies of internet intermediaries [2, 3, 4, 5, 6, 7, 8, 9].

In many cases the use of this data is not necessarily aligned with how the producers intend it to be used, and so its suitability may not be clear. In other cases conclusions drawn from the analysis based on this data does not necessary reflect the specifications and limitations of the data. Moreover, for all use-cases it is hard to know if the RBL being used is the best fit, if there is a better option or if a combination of two or more RBLs would add enough benefit to justify any extra cost. Note that

APWG.EU Tech 2023, June 21–22, 2023, Dublin, IE

sion.lloyd@icann.org (S. Lloyd); carlos.ganan@icann.org (C. Hernandez-Gañán); samaneh.tajali@icann.org (S. Tajalizadehkhoob)

CEUR Workshop Proceedings (CEUR-WS.org)

[1] https://developers.google.com/safe-browsing

the cost can be in terms of time and complexity as well as financial, so even free open-source feeds have some cost associated with them.

Misalignment with the intended use can have a significant impact on a project. For example, an RBL which contains low confidence or not vetted entries could result in an appreciable number of incorrect entries, known as false positives. Such a data feed might be perfectly acceptable if used to protect a small network where the mitigation of incorrect entries has a low associated cost. However, the same RBL may not be suitable for an application where a false positive results in a time and resource consuming investigation.

2. Objectives

Given the problem introduced above, in this document we propose a method to evaluate and characterise an RBL; not just in isolation but also in how multiple RBLs complement one another. We'll look at the general description of the RBL; things we can measure directly; things that we can make approximations of and things that we can only discover second-hand. We'll also discuss the implications and limitations of these measurements.

This work has been informed from earlier examples [10, 11, 12, 13, 14, 15, 16, 17]; but we have kept or modified parts of their suggested method to suit our requirements. As such; our approach is grounded in the projects that we have been involved with, other parties with other experiences may well have other metrics which they regard as important.

To move towards evaluating an RBL, or group of RBLs, we propose metrics that help measuring multiple aspects of a list. We then demonstrate the methods by which the metrics can be measured. Recall though that this work is based on the sources that we are already familiar with; it is likely that other RBLs have features which will require modifications to this method.

We will not discuss here the steps required to read RBL data as this will vary between RBLs. We do show, in Appendix A, the database schema that we use to harmonise data into a single, consistent, format. All of the RBL data we read is written into this structure, although it has had to evolve as new RBLs with new fields have been added.

Finally, there ae some things that we are explicitly not trying to measure. We are not looking to put a score on an RBL or say that one is demonstrably better than another; we want to increase our understanding of RBL data used regularly by us and our community, so that we can either use them with confidence, or understand why they are not suitable for a particular project. We also do not consider cost or licensing terms here; although these could be significant factors in any decision on whether to use an RBL. Lastly, we are not aiming to evaluate the absolute effectiveness of our RBLs as some of the existing work have already looked at that aspect [10, 11, 12, 13, 14].

The metrics we use are listed below and described in more detail in section 3.

2.1. General RBL Description

These are characteristics that we will know before we start to ingest data into our system. It may be a feature that initially brought the RBL to our interest, maybe to fill an identified gap in our other RBLs. We also include details that we need to know during the integration of an RBL into our system, like how it is distributed and what data it contains.

- RBL focus - What entry types does it contain (spam, phish, *etc.*)
- RBL mechanics - how is the RBL disseminated, what format is the data in, *etc.*
- Metadata - does the RBL provide more context on list entries, like malware family, phished brand, *etc.*

2.2. What We Can Measure Directly

These are the metrics we can measure directly based on the information provided in the RBLs.

- Volume - how many entries are present
- Overlap - how many entries in one RBL are in common with other RBLs
- Timeliness - how quickly do entries appear (compared to other RBLs)
- Churn - how dynamic are the entries

2.3. What We Can Measure Indirectly

These are metrics that can be measured indirectly from the data. So where we maybe have to sample the data in order to get an approximation of the answer, or where we have surrogate measurements in lieu of the thing that we actually want to investigate.

- Liveliness - how many entries are "active"
- Purity - how many are potential false positives
- Accuracy - what proportion of stated threat types match reality

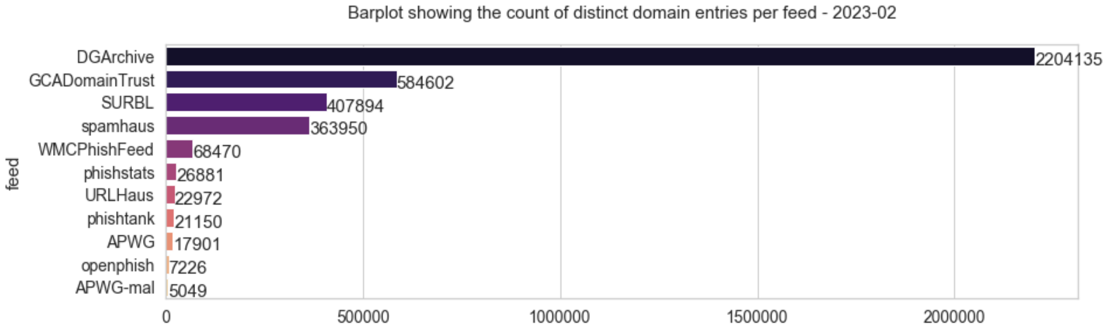


Figure 1: Number of unique domains seen over a fixed period of time

2.4. What We Cannot Measure

These are characteristics which cannot be derived from the data itself, but are discovered based on second-hand information. For example we look at the documentation for the RBL, consult FAQs or talk to the RBL providers to get this information.

- Catchment - are there geographic blindspots, collection method gaps (*e.g.* no mobile data), *etc.*
- Entry retesting - how frequently are entries retested to check if they should still be present on the RBL
- Reliability - is the data always available or are there issues transferring

3. Method

Looking in more detail at the metrics outlined above, in the remainder of the paper we demonstrate what we measure and, where appropriate, how we might use visualisations. Sticking to our four categories.

3.1. General RBL Description

RBL Focus The first thing to consider is the threat types that the RBL contains. Does it focus on a single threat type or contain multiple types? How does this relate to any other RBLs in our set, does it fill a known gap?

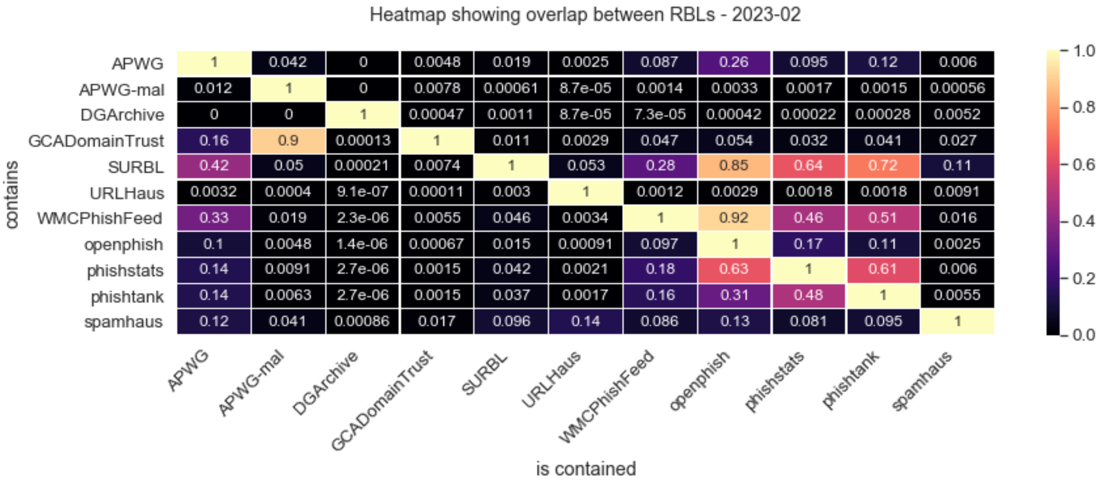
RBL Mechanics A prosaic but significant issue is how we can read the RBL and merge it into our larger dataset. We need to understand what delivery mechanism is used, is there an API, do we get the data formatted in CSV, JSON, *etc.* . Also, when we read the RBL does it provide the whole current list or a stream of new entries (a list of "point in time" observations). In the case of the latter the decision on how long an entry remains active is decided by us rather than the provider.

Metadata It can be useful to have context around a particular entry, and some RBLs provide more information, like a timestamp the entry was added, the malware family seen, the brand being phished and so on. Another useful data point is whether the entry is believed to be a malicious registration or a compromised but otherwise legitimate registration. All of this forms the metadata of an observation.

3.2. What We Can Measure Directly

Volume Possibly the easiest measurement to make is how many entries are present. Although here some care needs to be taken that the same thing is being measured in each case. For example, some RBLs contain just domains while others may contain URLs, but of course multiple URLs may well map to a single domain. We look at unique entries over a period of time, preferably a month or more, to give as good a representation as possible. This is particularly significant for those RBLs which provide a stream of new entries and so don't have the concept of a "current list". If we look at unique domains we see something like Figure 1. We can also produce similar figures but showing unique hosts, URLs, domains broken down by different threat types, *etc.* .

Higher volumes are, in general, desirable; this is not, however, the whole story. For example the DGArchive [18] data is based on enumeration of domain generation algorithms, and so the majority of those entries may never be registered. It is therefore arguable that we are not comparing like with like to other RBLs; we look to addresses issues



	APWG	APWG-mal	DGArchive	GCADomainTrust	SURBL	URLHaus	WMCPhishFeed	openphish	phishstats	phishtank	spamhaus
APWG	1	0.042	0	0.0048	0.019	0.0025	0.087	0.26	0.095	0.12	0.006
APWG-mal	0.012	1	0	0.0078	0.00061	8.7e-05	0.0014	0.0033	0.0017	0.0015	0.00056
DGArchive	0	0	1	0.00047	0.0011	8.7e-05	7.3e-05	0.00042	0.00022	0.00028	0.0052
GCADomainTrust	0.16	0.9	0.00013	1	0.011	0.0029	0.047	0.054	0.032	0.041	0.027
SURBL	0.42	0.05	0.00021	0.0074	1	0.053	0.28	0.85	0.64	0.72	0.11
URLHaus	0.0032	0.0004	9.1e-07	0.00011	0.003	1	0.0012	0.0029	0.0018	0.0018	0.0091
WMCPhishFeed	0.33	0.019	2.3e-06	0.0055	0.046	0.0034	1	0.92	0.46	0.51	0.016
openphish	0.1	0.0048	1.4e-06	0.00067	0.015	0.00091	0.097	1	0.17	0.11	0.0025
phishstats	0.14	0.0091	2.7e-06	0.0015	0.042	0.0021	0.18	0.63	1	0.61	0.006
phishtank	0.14	0.0063	2.7e-06	0.0015	0.037	0.0017	0.16	0.31	0.48	1	0.0055
spamhaus	0.12	0.041	0.00086	0.017	0.096	0.14	0.086	0.13	0.081	0.095	1

Figure 2: Overlap of unique domain entries seen between RBLs over a fixed period of time

like these later on. It is also true that some threats are more serious or active than others and so some entries offer more ”value”.

Overlap If we are looking to add a new RBL into our existing data, it is interesting to know how many entries are in common with our current data. Again we need to aggregate over a period of time and be careful to compare apples with apples. It may also be instructive to see different threat types separately. One simple measure is the overlap of unique domains shown in Figure 2.

This shows how much of one RBL is contained within another (and vice versa). For example, if we look at SURBL and openphish we can see that SURBL contains 0.85 (85%) of openphish. However, openphish contains just 0.015 (1.5%) of SURBL; while the absolute number of domains in common is the same, the difference is the underlying size of the RBL.

The view shows us some other interesting features; while the majority of overlaps are small, less than 5% or so, there are some which are much higher. This is where open sources are being read and incorporated into other RBLs, presumably after being validated to the required standard for that RBL. This could be significant if entries on multiple RBLs are being taken as multiple independent observations, when they may in fact stem from a single original source.

Timeliness The view above is interesting, and shows some cross-pollination between RBLs, so the next question is where two or more RBLs have the same entry, which gets it earlier and by how much? To this end we look at the time delta between an entry appearing on our “base feed” that we are considering and any other RBL, this gives us visualisations like that shown in Figure 3.

Here entries with a negative time show the base feed leading other RBL entries, whereas a positive time indicates it lagging behind. So ideally we want to see more weight to the left of the graph indicating that the RBL being considered is consistently getting entries earlier than others.

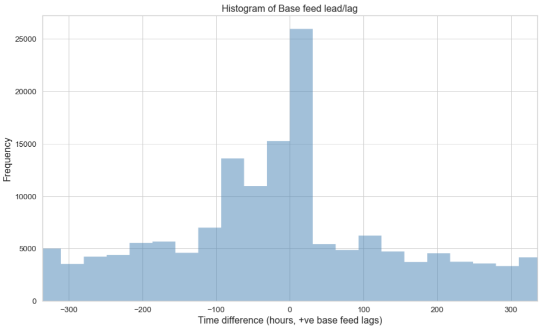


Figure 3: Where overlap is seen we can show if our considered RBL saw the domain earlier or later than the others

Churn For the RBLs that provide their whole current set of entries on each read, it is also useful to know how dynamic the list is. If an RBL’s volume stays the same as the previous iteration, is it because the list is static, or is it because as many entries

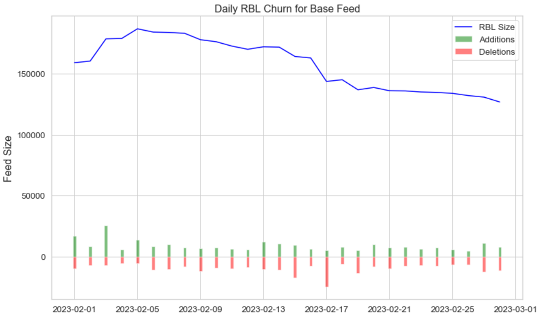


Figure 4: Volume over time for a single RBL along with the number of additions and deletions

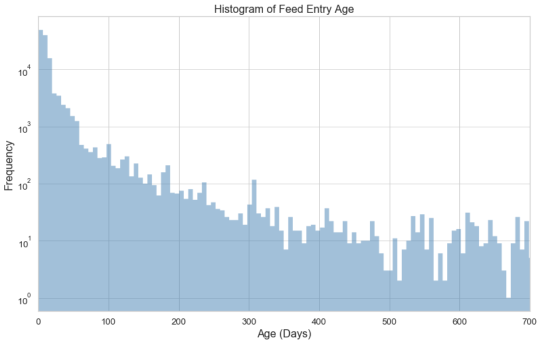


Figure 5: Histogram of entry ages for a single RBL, note the log scale

are being removed as are being added? To this end we can consider a single RBL over a period of time and plot its volume along with the number of new entries and removed entries as shown in Figure 4

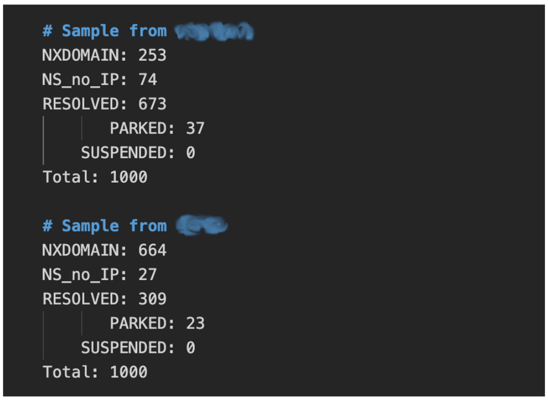
Note that removing stale entries which are no longer active threats can be as important as adding new entries, but is often not considered. To this end we can also look at the histogram of the ages of entries, see Figure 5, note the log y-scale. Figure 5 shows a healthy mix where the majority of entries have a short lifespan of days/weeks, with a small number being on the RBL for a year or more.

This analysis gives us more insight into how active the RBL is, how many new threats are being added and how many old threats are being removed. A higher churn reflects a more active RBL and so is seen as a positive feature. For those feeds which just provide "point in time" observations this analysis is not so relevant; although we can still look at the volumes of new threats being added.

3.3. What We Can Measure Indirectly

Liveliness Above we measured the volume of entries on an RBL. However, it is also interesting to know how many of those entries are "active". There may be entries which no longer resolve, or have been mitigated in other ways (for example, some registrars take control of the domain and "park" it).

We would struggle to capture this information for every entry on a sizeable RBL, and once we had finished we would need to start again to catch any new entries or changes in existing ones. One way to tackle this would be to pick a random sample of sufficient size to give us a measurement hopefully representative of the whole population.



```
# Sample from [redacted]
NXDOMAIN: 253
NS_no_IP: 74
RESOLVED: 673
        PARKED: 37
    SUSPENDED: 0
Total: 1000

# Sample from [redacted]
NXDOMAIN: 664
NS_no_IP: 27
RESOLVED: 309
        PARKED: 23
    SUSPENDED: 0
Total: 1000
```

Figure 6: Statuses of a sample of domains for two RBLs

If we see a large proportion of the entries not resolving then we need to think why this might be. While one reason may be that the RBL has stale information there may be other explanations. For example, maybe the RBL includes the output from one or more domain generation algorithms (DGAs), many of which are never actually registered.

Purity One of the more serious potential issues for RBLs is when they contain false positive reports, that is they contain entries which are not, and never have been, malicious. These entries are nearly impossible to discover *en masse*, they will only really become apparent during use. However, can we try to discover potential issues ahead of time? One thing which we look at is the overlap between the RBL and a source of "known good" entities. We are not aware of such a list, so use a surrogate source - a list of top domains, like the TRANCO top 1M. While these domains may still be malicious they are less likely to be. Also, for uses like blocking network traffic, any entry in the top 10,000 say would potentially be very disruptive.

We obviously want this score to be as low as possible, and where we suspect false positives we'd like to understand if there are explanations or mitigations we can use. To take DGAs as an example again, short DGA domains may coincidentally overlap with real words and legitimate registrations. To make this less of an issue it may be that only DGA domains with seven or more characters are retained.

Accuracy Where an RBL provides extra metadata, like threat types, do we believe that they are correct? Where we see entries in common between different RBLs, do they agree? This can be difficult to pin down as we do see the same entity reported for different threat types within the same RBL, so again we need to sample and check in order to get an idea of the scale of any issues.

We would like to be able to trust all the data that an RBL provides, not just the presence of entries, and the mis-classification of entries can have serious consequences in some cases. If an RBL has a low accuracy in terms of the metadata we may not be able to use it to generate statistics for example.

3.4. What We Can't Measure

Catchment RBLs have different collection mechanisms, even though some are aggregates of multiple primary sources. This will end up giving the RBL strengths and blindspots, which could be geographic or delivery related (*e.g.* no mobile data), no visibility of threats targeted at specific countries, *etc.*

Understanding of these can sometimes be found from FAQs, whitepapers, conversations with the providers or other second-hand methods. In many cases however the amount of information is, for operational reasons, limited.

We may need this information to identify RBLs that fill gaps in our current set, for other uses it may be that data for a particular locale is essential.

Entry Retesting We have seen that entries are removed from RBLs; but we cannot, from our measurements, definitively say why. Are statuses of entries being periodically reconfirmed, or are they just timed out? Some RBLs give this information but most do not, and deciding how long we trust entries for can be influenced by how this is being handled by the RBL.

Ideally all entries are frequently retested, but we appreciate that operationally this may not be possible.

Reliability A metric that can only be determined with continued monitoring and use, is whether the RBL data is always available, or are there sometimes issues transferring. This can influence our confidence in using an RBL in a production environment as if we have our own SLAs then the RBL should have something at least similar but preferably better.

For open-source RBLs with no contract (and therefore no SLA) only our experience with the RBL can give us this confidence.

4. Conclusions

In order to understand which RBL(s) are suitable for which projects, we need to understand the project requirements, the RBL characteristics and how multiple RBLs interact with each other.

We cannot claim that certain RBLs are better than others; but it can be that some RBLs are more suited to some projects.

However, from what we have seen of the RBLs we have access to, adding multiple sources increases the number of unique entities included and hence the comprehensiveness of the data used.

While in this work we outlined our evaluation processes, we emphasize the fact that these are not meant to be complete or prescriptive as they are predicated on our current use cases. It is quite likely that future projects, or new RBLs, will suggest new measures and modifications to existing ones.

References

[1] R. McMillan, What will stop spam?, http://sunsite.uakom.sk/sunworldonline/swol-12-1997/swol-12-vixie.html, 1997.

[2] ICANN, Domain abuse activity reporting, https://www.icann.org/octo-ssr/daar, 2017.

[3] Spamhaus, The World's Most Abused TLDs, https://www.spamhaus.org/statistics/tlds/, 2023.

[4] L. Interisle Consulting Group, Phishing landscape 2022, https://interisle.net/PhishingLandscape2022.pdf, 2022.

[5] C. David Barnett, The highest threat tlds - part 1, https://circleid.com/posts/20230112-the-highest-threat-tlds-part-1, 2013.

[6] J. Bayer, Y. Nosyk, O. Hureau, S. Fernandez, I. Paulovics, A. Duda, M. Korczyński, Study on domain name system (dns) abuse: Technical report, arXiv preprint arXiv:2212.08879 (2022).

[7] S. Maroofi, M. Korczyński, C. Hesselman, B. Ampeau, A. Duda, Comar: classification of compromised versus maliciously registered domains, in: 2020 IEEE European Symposium on Security and Privacy (EuroS&P), IEEE, 2020, pp. 607–623.

[8] M. Korczynski, M. Wullink, S. Tajalizadehkhoob, G. C. Moura, A. Noroozian, D. Bagley, C. Hesselman, Cybercrime after the sunrise: A statistical analysis of dns abuse in new gtlds, in: Proceedings of the 2018 on Asia Conference on Computer and Communications Security, 2018, pp. 609–623.

[9] S. Tajalizadehkhoob, T. Van Goethem, M. Korczyński, A. Noroozian, R. Böhme, T. Moore, W. Joosen, M. Van Eeten, Herding vulnerable cats: a statistical approach to disentangle joint responsibility for web security in shared hosting, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 553–567.

[10] S. Sinha, M. Bailey, F. Jahanian, Shades of grey: On the effectiveness of reputation-based "blacklists", in: 2008 3rd International Conference on Malicious and Unwanted Software (MALWARE), 2008, pp. 57–64. doi:10.1109/MALWARE.2008.4690858.

[11] J. Zhang, A. Chivukula, M. Bailey, M. Karir, M. Liu, Characterization of blacklists and tainted network traffic, in: M. Roughan, R. Chang (Eds.), Passive and Active Measurement, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 218–228.

[12] T. Vissers, P. Janssen, W. Joosen, L. Desmet, Assessing the effectiveness of domain blacklisting against malicious dns registrations, in: 2019 IEEE Security and Privacy Workshops (SPW), 2019, pp. 199–204. doi:10.1109/SPW.2019.00045.

[13] S. Ramanathan, J. Mirkovic, M. Yu, Blag: Improving the accuracy of blacklists, NDSS (2020). URL: https://par.nsf.gov/biblio/10205652. doi:10.14722/ndss.2020.24232.

[14] M. Kührer, C. Rossow, T. Holz, Paint it black: Evaluating the effectiveness of malware blacklists, in: Research in Attacks, Intrusions and Defenses: 17th International Symposium, RAID 2014, Gothenburg, Sweden, September 17-19, 2014. Proceedings 17, Springer, 2014, pp. 1–21.

[15] V. G. Li, M. Dunn, P. Pearce, D. McCoy, G. M. Voelker, S. Savage, Reading the tea leaves: A comparative analysis of threat intelligence, in: 28th USENIX Security Symposium (USENIX Security 19), USENIX Association, Santa Clara, CA, 2019, pp. 851–867. URL: https://www.usenix.org/conference/usenixsecurity19/presentation/li.

[16] A. Pitsillidis, C. Kanich, G. M. Voelker, K. Levchenko, S. Savage, Taster's choice: A comparative analysis of spam feeds, in: Proceedings of the 2012 Internet Measurement Conference, IMC '12, Association for Computing Machinery, New York, NY, USA, 2012, p. 427–440. URL: https://doi.org/10.1145/2398776.2398821. doi:10.1145/2398776.2398821.

[17] P. Vallina, V. Le Pochat, A. Feal, M. Paraschiv, J. Gamba, T. Burke, O. Hohlfeld, J. Tapiador, N. Vallina-Rodriguez, Mis-shapes, mistakes, misfits: An analysis of domain classification services, in: Proceedings of the ACM Internet Measurement Conference, IMC '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 598–618. URL: https://doi.org/10.1145/3419394.3423660. doi:10.1145/3419394.3423660.

[18] Fraunhofer, DGArchive website, https://dgarchive.caad.fkie.fraunhofer.de/welcome/, 2023.

A. Appendix 1: Database Schema

We write all of our RBL data to a single database table per month; most sources are read daily although some more frequently. Our current schema is shown in Table 1 although this has evolved with new RBLs and requirements.

Some processing is required for most entries to be written to these tables, for example domains are extracted from URLs, as are the TLD and suffix. This means we can get a more consistent view across all of our RBLs, coping with those which provide different fields in different formats, or use slightly different terminology.

Note that each time we read from a feed we add new entries rather than updating existing rows. This means that there will be duplicate entries when an entity is reported by an RBL for multiple days.

This is also true for RBLs which report on URLs, and so may have the same domain multiple times.

Table 1
RBL Data Schema

Column Name	Type	Notes
report_date	date	Some RBLs tell us, for others it's when we read that RBL.
domain	text	Stripped domain name
feed	text	Which source it came from
reason	text	Threat type - Spam, phishing, *etc.*
full_identifier	text	Some RBLs give URLs or include subdomains.
score	int	Some RBLs give a confidence score
suffix	text	Suffix according to the public suffix list
tld	text	Top-level domain
tld_type	text	country code (CC) or generic (gTLD) top-level domain
registrar	text	If known
reg_id	int	Registrar ID, if known
seen_since	timestamp	Initial report_date
url_shortener	boolean	Is it a known URL shortener (*e.g.* bit.ly); won't be reliable
sub_feed	text	Some RBLs aggregate other sources, if this is the case the original source will be here
notes	text	Any other info the RBL gave that might be useful. Will depend on the RBL
dga	boolean	Is the entry known to be from a domain generation algorithm
ip	boolean	Is the entry an IP address