Anti-Spam Research Group	J. Levine
Internet-Draft	Taughannock Networks
Intended status: Standards Track	October 10, 2008
Expires: April 13, 2009	

DNS Blacklists and Whitelists
draft-irtf-asrg-dnsbl-07

Status of this Memo

Abstract

The rise of spam and other anti-social behavior on the Internet has led to the creation of shared blacklists and whitelists of IP addresses or domains. The DNS has become a de-facto standard method of distributing these blacklists and whitelists. This memo documents the structure and usage of DNS based blacklists and whitelists, and the protocol used to query them.

IRTF Notice

This document is a product of the Anti-Spam Research Group (ASRG). Comments and discussion may be directed to the ASRG mailing list, asrg@irtf.org.
This document represents the consensus of the Anti-Spam Research Group of the Internet Research Task Force.

Table of Contents

1.   Introduction

In 1997, Dave Rand and Paul Vixie, well known Internet software engineers, started keeping a list of IP addresses that had sent them spam or engaged in other behavior that they found objectionable. Word of the list quickly spread, and they started distributing it as a BGP feed for people who wanted to block all traffic from listed IP addresses at their routers. The list became known as the Real-time Blackhole List (RBL).

Many network managers wanted to use the RBL to block unwanted e-mail, but weren't prepared to use a BGP feed. Rand and Vixie created a DNS-based distribution scheme that quickly became more popular than the original BGP distribution. Other people created other DNS-based blacklists either to compete with the RBL or to complement it by listing different categories of IP addresses. Although some people refer to all DNS-based blacklists as ``RBLs'', the term properly is used for the MAPS RBL, the descendant of the original list. (In the United States, the term RBL is a registered service mark of Trend Micro[MAPSRBL] (, "MAPS RBL+," .).)

The conventional term is now DNS Blacklist or Blocklist, or DNSBL. Some people also publish DNS-based whitelists or DNSWLs. Network managers typically use DNSBLs to block traffic and DNSWLs to preferentially

accept traffic. The structure of a DNSBL and DNSWL are the same, so in the subsequent discussion we use the abbreviation DNSxL to mean either. This document defines the structure of DNSBLs and DNSWLs. It describes the structure, operation, and use of DNSBLs and DNSWLs but does not describe or recommend policies for adding or removing addresses to and from DNSBLs and DNSWLs, nor does it recommend policies for using them. We anticipate that management policies will be addressed in a companion document.

> **Requirements Notation:** The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] (Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," March 1997.).

---

2. Structure of an IP address DNSBL or DNSWL

A DNSxL is a zone in the DNS[RFC1034] (Mockapetris, P., "Domain names - concepts and facilities," November 1987.)[RFC1035] (Mockapetris, P., "Domain names - implementation and specification," November 1987.). The zone containing resource records identifies hosts present in a blacklist or whitelist. Hosts were originally encoded into DNSxL zones using a transformation of their IP addresses, but now host names are sometimes encoded as well. Most DNSxLs still use IP addresses.

2.1. IP address DNSxL

An IPv4 address DNSxL has a structure adapted from that of the rDNS. (The rDNS, reverse DNS, is the IN-ADDR.ARPA[RFC1034] (Mockapetris, P., "Domain names - concepts and facilities," November 1987.) and IP6.ARPA[RFC3596] (Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6," October 2003.) domains used to map IP addresses to domain names.) Each IPv4 address listed in the DNSxL has a corresponding DNS entry created by reversing the order of the octets of the text representation of the IP address, and appending the domain name of the DNSxL.
If, for example, the DNSxL is called bad.example.com, and the IPv4 address to be listed is 192.0.2.99, the name of the DNS entry would be 99.2.0.192.bad.example.com. Each entry in the DNSxL MUST have an A record. DNSBLs SHOULD have a TXT record that describes the reason for the entry. DNSWLs MAY have a TXT record that describes the reason for the entry. The A record conventionally has the value 127.0.0.2, but MAY have other values as described below in Combined IP address DNSxLs. The

TXT record describes the reason that the IP is listed in the DNSxL, and
is often used as the text of an SMTP error response when an SMTP client
attempts to send mail to a server using the list as a DNSBL, or as
explanatory text when the DNSBL is used in a scoring spam filter. The
DNS records for this entry might be:

```
99.2.0.192.bad.example.com    A      127.0.0.2
99.2.0.192.bad.example.com    TXT
         "Dynamic address, see http://bad.example.com?192.0.2.99"
```

Some DNSxLs use the same TXT record for all entries, while others
provide a different TXT record for each entry or range of entries that
describes the reason that entry or range is listed. The reason often
includes the URL of a web page where more information is available.
Client software MUST check the A record and MAY check the TXT record.
If a range of addresses is listed in the DNSxL, the DNSxL MUST contain
an A record (or a pair of A and TXT records) for every address in the
DNSxL. Conversely, if an IP address is not listed in the DNSxL, there
MUST NOT be any records for the address.

2.2. IP address DNSWL

Since SMTP has no standard way for a server to advise a client why a
request was accepted, TXT records in DNSWLs are not very useful. Some
DNSWLs contain TXT records anyway to document the reasons that entries
are present.
It is possible and occasionally useful for a DNSxL to be used as a
DNSBL in one context and a DNSWL in another. For example, a DNSxL that
lists the IP addresses assigned to dynamically assigned addresses on a
particular network might be used as a DNSWL on that network's outgoing
mail server or intranet web server, and used as a DNSBL for mail
servers on other networks.

2.3. Combined IP address DNSxL

In many cases, an organization maintains a DNSxL that contains multiple
entry types, with the entries of each type constituting a sublist. For
example, an organization that publishes a DNSBL listing sources of
unwanted e-mail might wish to indicate why various addresses are
included in the list, with one sublist for addresses listed due to
sender policy, a second list for addresses of open relays, a third list
for hosts compromised by malware, and so forth. (At this point all of
the DNSxLs with sublists of which we are aware are intended for use as
DNSBLs, but the sublist techniques are equally usable for DNSWLs.)

There are three common methods of representing a DNSxL with multiple sublists: subdomains, multiple A records, and bit encoded entries. DNSxLs with sublists SHOULD use both subdomains and one of the other methods.

Sublist subdomains are merely subdomains of the main DNSxL domain. If for example, bad.example.com had two sublists relay and malware, entries for 192.0.2.99 would be 99.2.0.192.relay.bad.example.com or 99.2.0.192.malware.bad.example.com. If a DNSxL contains both entries for a main domain and for sublists, sublist names MUST be at least two characters and contain non-digits, so there is no problem of name collisions with entries in the main domain, where the IP addresses consist of digits or single hex characters.

To minimize the number of DNS lookups, multiple sublists can also be encoded as bit masks or multiple A records. With bit masks, the A record entry for each IP is the logical OR of the bit masks for all of the lists on which the IP appears. For example, the bit masks for the two sublists might be 127.0.0.2 and 127.0.0.4, in which case an entry for an IP on both lists would be 127.0.0.6:

```
    99.2.0.192.bad.example.com    A      127.0.0.6
```

With multiple A records, each sublist has a different assigned value such as 127.0.1.1, 127.0.1.2, and so forth, with an A record for each sublist on which the IP appears:

```
    99.2.0.192.bad.example.com    A      127.0.1.1
    99.2.0.192.bad.example.com    A      127.0.1.2
```

There is no widely used convention for mapping sublist names to bits or values, beyond the convention that all A values SHOULD be in the 127.0.0.0/8 range to prevent unwanted network traffic if the value is accidentally used as an IP address.

DNSxLs that return multiple A records sometimes return multiple TXT records as well, although the lack of any way to match the TXT records to the A records limits the usefulness of those TXT records. Other combined DNSxLs return a single TXT record.

2.4. IPv6 DNSxLs

The structure of DNSxLs based on IPv6 addresses is adapted from that of the IP6.ARPA domain defined in [RFC3596] (Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6," October 2003.). Each entry MUST be a 32-component hex nibble-reversed IPv6 addresses suffixed by the the DNSxL domain. For example, to represent the address:

```
2001:db8:1:2:3:4:567:89ab
```

in the DNSxL ugly.example.com, the entry might be:

```
b.a.9.8.7.6.5.0.4.0.0.0.3.0.0.0.2.0.0.0.1.0.0.0.8.b.d.0.1.0.0.2.
                  ugly.example.com. A 127.0.0.2
                                       TXT "Spam received."
```

Combined IPv6 sublist DNSxLs are represented the same way as IPv4
DNSxLs, replacing the four octets of IPv4 address with the 32 nibbles
of IPv6 address.
A single DNSxL could in principle contain both IPv4 and IPv6 addresses,
since the different lengths prevent any ambiguity. If a DNSxL is
represented using traditional zone files and wildcards, there is no way
to specify the length of the name that a wildcard matches, so wildcard
names would indeed be ambiguous for DNSxLs served in that fashion.

---

3. Domain name DNSxLs

A few DNSxLs list domain names rather than IP addresses. They are
sometimes called RHSBLs, for right hand side blacklists. The names of
their entries MUST contain the listed domain name followed by the name
of the DNSxL. If the DNSxL were called doms.example.net, and the domain
invalid.edu were to be listed, the entry would be named
invalid.edu.doms.example.net:

```
invalid.edu.doms.example.net    A     127.0.0.2
invalid.edu.doms.example.net    TXT    "Host name used in phish"
```

Name-based DNSBLs are far less common than IP based DNSBLs. There is no
agreed convention for wildcards.
Name-based DNSWLs can be created in the same manner as DNSBLs, and have
been used as simple reputation systems with the values of octets in the
A record representing reputation scores and confidence values,
typically on a 0-100 or 0-255 scale.

---

4. DNSxL cache behavior

The per-record time-to-live and zone refresh intervals of DNSBLs and
DNSWLs vary greatly depending on the management policy of the list. The
TTL and refresh times SHOULD be chosen to reflect the expected rate of
change of the DNSxL. A list of IP addresses assigned to dynamically
allocated dialup and DHCP users could be expected to change slowly, so

the TTL might be several days and the zone refreshed once a day. On the other hand, a list of IP addresses that had been observed sending spam might change every few minutes, with comparably short TTL and refresh intervals.

---

5. Test and contact addresses

IPv4 based DNSxLs MUST contain an entry for 127.0.0.2 for testing purposes. IPv4 based DNSxLs MUST NOT contain an entry for 127.0.0.1. DNSBLs that return multiple values SHOULD have multiple test addresses so that, for example, a DNSBL that can return 127.0.0.5 would have a test record for 127.0.0.5 that returns an A record with the value 127.0.0.5, and a corresponding TXT record.
IPv6 based DNSxLs MUST contain an entry for ::FFFF:7F00:2 (::FFFF:127.0.0.2), and MUST NOT contain an entry for ::FFFF:7F00:1 (::FFFF:127.0.0.1), the IPv4-Mapped IPv6 Address (Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture," February 2006.) [RFC4291] equivalents of the IPv4 test addresses.
Domain name based DNSxLs MUST contain an entry for the [RFC2606] (Eastlake, D. and A. Panitz, "Reserved Top Level DNS Names," June 1999.) reserved domain name "TEST" and MUST NOT contain an entry for the reserved domain name "INVALID".
DNSxLs also MAY contain an A record at the apex of the DNSxL zone that points to a web server, so that anyone wishing to learn about the bad.example.net DNSBL can check http://bad.example.net.
The combination of a test address that MUST exist and an address that MUST NOT exist allows a client system to defend against DNSxLs which deliberately or by accident install a wildcard that returns an A record for all queries. DNSxL clients SHOULD periodically check appropriate test entries to ensure that the DNSxLs they are using are still operating.

---

6. Typical usage of DNSBLs and DNSWLs

DNSxLs can be served either from standard DNS servers, or from specialized servers like rbldns (Bernstein, D., "rbldns, in 'djbdns'," .) [RBLDNS] and rbldnsd (Tokarev, M., "rbldnsd: Small Daemon for DNSBLs," .) [RBLDNSD] that accept lists of IP addresses and CIDR ranges and synthesize the appropriate DNS records on the fly. Organizations that make heavy use of a DNSxL usually arrange for a private mirror of the DNSxL, either using the standard AXFR and IXFR or by fetching a file containing addresses and CIDR ranges for the specialized servers. If a /24 or larger range of addresses is listed, and the zone's server uses traditional zone files to represent the DNSxL, the DNSxL MAY use

wildcards to limit the size of the zone file. If for example, the entire range of 192.0.2.0/24 were listed, the DNSxL's zone could contain a single wildcard for *.2.0.192.bad.example.com.

DNSBL clients are most often mail servers or spam filters called from mail servers. There's no requirement that DNSBLs be used only for mail, and other services such as IRC use them to check client hosts that attempt to connect to a server.

A client MUST interpret any returned A record as meaning that an address or domain is listed in a DNSxL. Mail servers that test combined lists most often handle them the same as single lists and treat any A record as meaning that an IP is listed without distinguishing among the various reasons it might have been listed. DNSxL clients SHOULD be able to use bit masks and value range tests on returned A record values in order to select particular sublists of a combined list.

Mail servers typically check a list of DNSxLs on every incoming SMTP connection, with the names of the DNSxLs set in the server's configuration. A common usage pattern is for the server to check each list in turn until it finds one with a DNSBL entry, in which case it rejects the connection, or a DNSWL entry in which case it accepts the connection. If the address appears on no list at all (the usual case for legitimate mail), the mail server accepts the connection. In another approach, DNSxL entries are used as inputs to a weighting function that computes an overall score for each message.

The mail server uses its normal local DNS cache to limit traffic to the DNSxL servers and to speed up retests of IP addresses recently seen. Long-running mail servers MAY cache DNSxL data internally, but MUST respect the TTL values and discard expired records.

An alternate approach is to check DNSxLs in a spam filtering package after a message has been received. In that case, the IP(s) to test are usually extracted from "Received:" header fields or URIs in the body of the message. The DNSxL results can be used to make a binary accept/ reject decision, or in a scoring system.

Packages that test multiple header fields MUST be able to distinguish among values in lists with sublists since, for example, an entry indicating that an IP is assigned to dialup users might be treated as a strong indication that a message would be rejected if the IP sends mail directly to the recipient system, but not if the message were relayed through an ISP's mail server.

Name-based DNSBLs have been used both to check domain names of e-mail addresses and host names found in mail headers, and to check the domains found in URLs in message bodies.

---

7. Security Considerations

Any system manager that uses DNSxLs is entrusting part of his or her server management to the parties that run the lists. A DNSBL manager

that decided to list 0/0 (which has actually happened) could cause every server that uses the DNSBL to reject all mail. Conversely, if a DNSBL manager removes all of the entries (which has also happened), systems that depend on the DNSBL will find that their filtering doesn't work as they want it to.

Since DNSxL users usually make a query for every incoming e-mail message, the operator of a DNSxL can extract approximate mail volume statistics from the DNS server logs. This has been used in a few instances to estimate the amount of mail individual IPs or IP blocks send[SENDERBASE] (Ironport Systems, "Senderbase," .) [KSN] (Levine, J., "The South Korean Network Blocking List," .).

As with any other DNS based services, DNSBLs and DNSWLs are subject to various types of DNS attacks which are described in [RFC3833] (Atkins, D. and R. Austein, "Threat Analysis of the Domain Name System (DNS)," August 2004.).

---

8. IANA Considerations   [TOC]

This memo includes no request to IANA.

---

9. References   [TOC]

9.1. Normative References

[TOC]

[RFC1034]	Mockapetris, P., "Domain names - concepts and facilities," STD 13, RFC 1034, November 1987 (TXT).
[RFC1035]	Mockapetris, P., "Domain names - implementation and specification," STD 13, RFC 1035, November 1987 (TXT).
[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC2606]	Eastlake, D. and A. Panitz, "Reserved Top Level DNS Names," BCP 32, RFC 2606, June 1999 (TXT).
[RFC3596]	Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6," RFC 3596, October 2003 (TXT).
[RFC4291]	Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture," RFC 4291, February 2006 (TXT).

9.2. Informative References

TOC

[RFC3833]	Atkins, D. and R. Austein, "Threat Analysis of the Domain Name System (DNS)," RFC 3833, August 2004 (TXT).
[RBLDNS]	Bernstein, D., "rbldns, in 'djbdns'."
[MAPSRBL]	"MAPS RBL+."
[RBLDNSD]	Tokarev, M., "rbldnsd: Small Daemon for DNSBLs."
[SENDERBASE]	Ironport Systems, "Senderbase."
[KSN]	Levine, J., "The South Korean Network Blocking List."

Appendix A. Change Log

TOC

NOTE TO RFC EDITOR: This section may be removed upon publication of this document as an RFC.

A.1. Changes since -asrg-dnsbl-06

TOC

Change forbidden example from EXAMPLE to INVALID.
Remove SOA encoded email addresses.
Change IPv6 test addresses.

A.2. Changes since -asrg-dnsbl-05

TOC

Pervasive edits to standard language, including RFC2119 terms.
Test entries clarified for IPv4, invented for IPv6 and domains.

Author's Address

TOC

	John Levine
	Taughannock Networks
	PO Box 727
	Trumansburg, NY 14886
Phone:	+1 831 480 2300
Email:	standards@taugh.com
URI:	http://www.taugh.com