Domain Name System Operations   P. Vixie
Internet-Draft   Farsight Security, Inc.
Intended status: Informational   V. Schryver
Expires: April 14, 2017   Rhyolite Software
                                                        October 11, 2016

Response Policy Zones
draft-vixie-dns-rpz-00

Abstract

This document describes a method for expressing DNS response policy inside a specially constructed DNS zone, and for processing the contents of such response policy zones (RPZ) inside recursive name servers. These "DNS Firewalls" are widely used in fighting Internet crime and abuse.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

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].

This document describes DNS Firewalls or a method of expressing DNS [RFC1034] policy information inside specially constructed DNS zones, allowing DNS reputation data producers and subscribers to cooperate in the application of policies to real time DNS responses. Using this policy information, DNS resolution for low-reputation DNS data can be made to deliberately fail or to return local data such as an alias to a "walled garden". A full description of the expressible policies is given in Section 2.

Configuration examples using ISC BIND Version 9 (BIND9) [ISC-ARM] are given, because work to add RPZ to that platform was started in 2009. The RPZ specification itself is free to implement and free to use in operation, and so we expect other recursive DNS implementations to also implement DNS Firewalls as described by the RPZ specification.

2. Zone Format

A DNS Response Policy Zone (RPZ) is a DNS zone. As such its contents can be transferred between servers DNS in whole (AXFR) [RFC5936] or incrementally as changes occur (IXFR) [RFC1995], authenticated and protected by TSIG transaction signatures [RFC2845], and expedited by real time change notifications (NOTIFY) [RFC1996], all subject to familiar DNS access controls. An RPZ need not support query access since query access is never required. It is the zone transfer of RPZ content from producers to subscribers which effectively publishes the policy data, and it is the subscriber's server configuration which promotes RPZ payload data into DNS control plane data.

To be a valid DNS zone, an RPZ is required to have an SOA record and at least one NS record at its apex. The SOA record is real, with a serial number used for NOTIFY and IXFR and timers used for AXFR and IXFR. The MNAME field or domain name of the primary source of the zone and the RNAME field or mailbox of the person responsible for the zone are often used by RPZ providers to label their policy zones.

Because query access is never required, an RPZ's apex NS record will never be used and no parent delegation is required. The zone name of the NS record need not be a unique fully qualified domain name. By convention, a single NS record having the deliberately bogus value "LOCALHOST" is used as a placeholder. The zone's name can be fully qualified to show the identity of its producer or maintainer.

Like any DNS zone, an RPZ consists of RRsets or sets of resource records (RRs) with a common owner name and type. The owner names or left hand sides of RRsets other than the SOA and NS express policy triggers or characteristics of DNS response that require action. The action enjoined by a policy trigger is encoded in the rdata or right hand sides of the records.

All POLICY described here are from RPZ Format 1 unless otherwise noted. Policy triggers from a higher format number than a recursive name server's implementation level are expected to be invisible to that implementation. Policy actions from a higher format number are likely to be misinterpreted as CNAME local data by older implementations.

3. Policy Actions

An RPZ resource record can specify any of five different results or actions.

NXDOMAIN

A single resource record (RR) consisting of a CNAME whose target is the root domain (.) will cause a response of NXDOMAIN to be returned. This is the most commonly used RPZ action.

NODATA

A single RR consisting of a CNAME whose target is the wildcard top-level domain (*.) will cause a response of NODATA (ANCOUNT=0) to be returned regardless of query type.

PASSTHRU

[Format 3] It is sometimes necessary to exempt some DNS responses from the response policy rule that covers an entire domain or a large IP address block. Exempting some clients of a DNS resolver from all RPZ rewriting can also be useful for research into attackers and debugging. The PASSTHRU action encoded as a CNAME with a target or right hand side of "rpz-passthru." overrides lower precedence policies.

This policy zone record

```
$ORIGIN RPZ.EXAMPLE.ORG.
ok.domain.com     CNAME   rpz-passthru.
```

would exempt requests for ok.domain.com from NXDOMAIN policy or action of the following record:

```
$ORIGIN RPZ.EXAMPLE.ORG.
domain.com          CNAME   .
*.domain.com        CNAME   .
```

The deprecated [Format 1] encoding of the PASSTHRU action was a CNAME with a target equal to the QNAME field of the DNS request. That encoding could not be used with some desirable triggers.

DROP

The "DROP" policy that consists of discarding the request and response is specified by a CNAME whose target is "rpz-drop". For example, with

```
$ORIGIN RPZ.EXAMPLE.ORG.
example.com         CNAME   rpz-drop.
```

nothing is sent to a DNS client trying to resolve example.com, not even a DNS error response.

TCP-Only

The "TCP-Only" policy is specified by a CNAME whose target is "rpz-tcp-only". It changes UDP responses to short, truncated DNS responses that require the DNS client to try again with TCP. It is used to mitigate distributed DNS reflection attacks and is similar to the "slip" parameter of DNS Response Rate Limiting (RRL) [ISC-RRL].

Local Data

Any other RRset that is not the RPZ encoding of NXDOMAIN (.) or NODATA (*.) and not a CNAME with a target domain name starting with "rpz-", specifies local data used to generate synthetic DNS responses. If any local data policy actions are present, then any questions for RR types that are not present in the local data will be answered as NODATA (ANCOUNT=0) as if this DNS server were authoritative for the query name. The most common local data is a CNAME RR pointing to a local walled garden. Such CNAME RRs are distinguishable from other rpz actions because the CNAME target name will not be the root (.), root wildcard (*.), or be a subdomain of a top level domain that starts with "rpz-".

[Format 3] A special form of local data involves a CNAME RR with a wildcarded target name. Wildcards are not valid as CNAME targets in ordinary DNS zones. This special form causes the QNAME to be prepended to the wildcarded target to communicate the triggering QNAME value to the the walled garden DNS server. For example a policy action of "CNAME *.EXAMPLE.COM" and a query name of "EVIL.ORG." will result in a synthetic response of "EVIL.ORG CNAME EVIL.ORG.EXAMPLE.COM." The purpose for this special form is query logging in the walled garden's DNS server.

4. Policy Triggers

All five types of RPZ triggers are encoded by RRset owner names in an RPZ.

Three of the types of triggers are based on target data (RDATA). Those policies are conceptually applied after recursion, so that the recursive DNS resolver's cache contains either nothing or "truth" even if this truth is hidden by current policy. If the policy changes, the original data is available for processing under the changed policy. The other types of policy trigger are independent of cache contents or recursion results.

QNAME

The QNAME policy trigger applies to requested domain names (QNAME). The owner name of an RPZ QNAME policy RRset is the relativized name of the domain name about which policy is being

expressed. For example, if the RPZ apex name is RPZ.EXAMPLE.ORG, an RRset at DOMAIN.COM.RPZ.EXAMPLE.ORG would affect responses to requests about DOMAIN.COM. Wildcards also work, and so the owner name "*.DOMAIN.COM.RPZ.EXAMPLE.ORG" would trigger on queries to any subdomain of DOMAIN.COM. To control the policy for both a name and its subdomains, two policy RRsets must be used, one for the domain itself and another for a wildcard sub-domain. In the following example, queries for both DOMAIN.COM and all subdomains of DOMAIN.COM will result in synthetic NXDOMAIN responses.

```
$ORIGIN RPZ.EXAMPLE.ORG.
DOMAIN.COM          CNAME   .
*.DOMAIN.COM        CNAME   .
```

Response IP address [Format 2]

The response IP policy trigger is based on target data (RDATA). It matches IP addresses that would otherwise appear in A and AAAA records in the "answer" section of a DNS response. IP addresses the in "authority" and "additional" sections are not considered. Response IP policy RRsets have owner names that are sub-domains of "rpz-ip" relativized to the RPZ apex name, and an encoded IP address or block of addresses.

IPv4 addresses are encoded as "prefix.B4.B3.B2.B1.rpz-ip". The prefix length, "prefix", must be between 1 and 32. All four bytes, B4, B3, B2, and B1, must be present and written in decimal ASCII. B4 is the least significant octet of the IP address and B1 is the most significant octet, just as in the IN-ADDR.ARPA naming convention.

IPv6 addresses are encoded in a format similar to the standard IPv6 text representation, "prefix.W8.W7.W6.W5.W4.W3.W2.W1.rpz-ip". Each of W8,...,W1 is a one to four digit hexadecimal ASCII number representing 16 bits of the IPv6 address with no leading zeroes and reversed as in IP6.ARPA. All 8 words must be present unless a "zz" label is present. "Zz" is analagous to the double-colon (::) in the standard IPv6 address representation. The "zz" label is expanded to zero-fill the middle portion of the IPv6 address. The prefix length must be between 1 and 128. For example, to force an NXDOMAIN response whenever a truthful response would contain an "answer" section A RRset having an address in 192.168.1.0/24 unless address 192.168.1.2 is present, the RPZ would contain the following:

```
$ORIGIN RPZ.EXAMPLE.ORG.
24.0.1.168.192.rpz-ip   CNAME   .
32.2.1.168.192.rpz-ip   CNAME   rpz-passthru.
```

In another example, to answer NODATA (ANCOUNT=0) whenever a truthful response would contain an answer AAAA RRset having an address 2001:0002::/48 unless address 2001:0002::3 was present, the RPZ would contain these records:

```
$ORIGIN RPZ.EXAMPLE.ORG.
48.zz.2.2001.rpz-ip     CNAME   *.
128.3.zz.2.2001.rpz-ip  CNAME   rpz-passthru.
```

Client IP address

The IP addresses of DNS clients sending requests can also be triggers. This can be useful for disabling RPZ rewriting for DNS clients used to test or investigate. Client IP address triggers are encoded like response IP address triggers except that they are subdomains of rpz-client-ip instead of rpz-ip. For example, the following would drop all requests from clients in 192.168.1.0/24 and answer truthfull requests from a client at 2001:2::3.

```
$ORIGIN RPZ.EXAMPLE.ORG.
48.zz.2.2001.rpz-client-ip          CNAME *.
128.3.zz.2.2001.rpz-client-ip   CNAME rpz-passthru.
```

NSDNAME [Format 2]

The NSDNAME policy trigger matches name server names (NS RR) of any name server which is in the data path for an RRset present in the answer section of a DNS response. The data path is all delegation points from (and including) the root zone to the closest enclosing NS RRset for the owner name of the answering RRset.

In other words, an NSDNAME trigger is checked by first considering the named servers (domain names in the NS records) for the query domain (QNAME), then the name servers for the parent of the query domain, and so on until the name servers for the root (.) have been checked. This process stops immediately for a given trigger when the trigger is hit.

This process can be expensive, especially when it comes to checking the many NS records for the top level domains and the root. Because the name servers for the root and the TLDs are rarely used as RPZ triggers, common RPZ implementations have a "min-ns-dots" parameter that stops NSDNAME and NSIP checking early.

Despite their costs, NSDNAME triggers can be more effective than QNAME and IP triggers. Malefactors can more easily change their direct domain names and IP addresses detected by QNAME and IP

triggers than they can their change NS names and addresses in parent domains such as TLDs.

NSDNAME policies are encoded as RRsets in sub-domains of "rpz-nsdname" but otherwise much like QNAME policies. For example, to force an NXDOMAIN answer whenever a name server for the requested domain or one of its parents is NS.DOMAIN.COM, the RPZ would contain the following:

```
$ORIGIN RPZ.EXAMPLE.ORG.
NS.DOMAIN.COM.rpz-nsdname   CNAME .
```

Some implementations of DNS RPZ will exhaustively discover all ancestral zone cuts above the query name and will learn the NS RRset at the apex of each delegated zone. Other implementations will know only the zone cut information which has naturally come into the cache, which will often include only parent delegation name server names or "glue." Since apex ("below the cut") NS RRsets and delegation NS RRsets need not exactly match, this can lead to instability in DNS RPZ behavior in the presence of zone cuts having differences between the NS RRsets above and below a zone cut. This potential inconsistency must be taken into account when designing a security policy or testing DNS RPZ.

The BIND9 and Unbound RPZ implementations use whatever NS RRsets that are in their caches unless there are none, in which case they recurse. In practice this is best, because the authoritative apex NS RRsets of domains operated by malefactors is often peculiar. For example, RRs like "example.com NS ." claiming that the root is authoritative are popular choices in malefactor apex NS RRsets.

NSIP [Format 2]

The NSIP policy trigger matches name server addresses (an A or AAAA RR that's been referenced by an NS RRset). NSIP is much like NSDNAME (described above) except that the matching is by name server address rather than name server name. NSIP policies are expressed as sub-domains of "rpz-nsip" and have the same sub-domain naming convention as described for response IP policy triggers above.

In other words, an NSIP trigger is checked by first considering all of the IP address for all the named servers (or domain names in the NS records) for the query domain (QNAME), then the IP addresses for name servers for the parent of the qname, and so on until the name servers for the root (.) have been checked. This process stops immediately when the trigger is hit.

This process can be very expensive, especially when it comes to checking the many NS and IP RRs for the TLDs and the root. Because the IP addresses of name servers for the root and the TLDs are rarely used as RPZ triggers, common RPZ implementations have a "min-ns-dots" parameter that stops NSIP and NSDNAME checking early.

Nevertheless, NSIP triggers can be more effective than QNAME and IP triggers. Malefactors can more easily change their direct domain names and IP addresses that QNAME and IP triggers detect than they can their change NS names and addresses that are in parent domains such as TLDs.

As with NS Domain Name or NSDNAME triggers, some implementations of DNS RPZ will exhaustively discover all IP addresses (V4 and V6) associated with each name server name. Other DNS RPZ implementations will only know the subset of IP addresses which have entered the cache naturally. This can lead to instabilities in the DNS RPZ behavior since the natural entry of IP addresses into the cache is itself unstable. This instability must be taken into account when designing a security policy or testing DNS RPZ.

Also like NSDNAME triggers, the BIND9 and Unbound RPZ implementations use whatever A or AAAA RRsets that are in their caches for NS domains unless there are none, in which case they recurse. In practice this is best, not only because the authoritative A and AAAA RRsets of name servers for the domains of malefactors are often imaginitive, but because even when they are reasonable, the authoritative DNS servers for their NS domains are often extremely slow or broken.

5. Subscriber Behavior

RPZs must be primary or secondary zones at subscriber recursive resolvers. They can only be searched in a recursive server's own storage, because additional network transactions for DNS resolvers are extremely undesirable.

By default, policies are applied only on DNS requests that ask for recursion (RD=1) and which either do not request DNSSEC metadata (DO=0) or for which no DNSSEC metadata exists.

Policies are checked at each stage of resolving a domain name defined by a CNAME or DNAME record, stopping at the first CNAME in the chain with an applicable policy. CNAME targets in a CNAME chain are checked as if they were query names.

If a policy trigger results in a modified answer, then that modified answer will include in its "authority" section the SOA RR of the DNS RPZ whose policy was used to generate the modified answer. This SOA RR will tell both the fully qualified name of the DNS RPZ and the serial number of the policy data which was connected to the DNS control plane at the time the answer was modified.

Response policy zones are loaded in the usual way. For primary zones this may mean loading the contents of a local file into memory, or connecting to a database. For secondary zones this means transferring the zone from the primary server using zone transfer such as IXFR [RFC1995] or AXFR [RFC5936]. It is strongly recommended that all secondary zone transfer relationships be protected with transaction signatures (DNS TSIG) and that real time change notification be enabled using the DNS NOTIFY protocol [RFC1996].

DNS resolvers often have limited or no notion of a DNS zone. They sometimes have special local zones, but generally have no implementations of IXFR, AXFR, or NOTIFY. Therefore, an external module or daemon that maintains local copies of policy zones can be useful.

To connect the name server's control plane to the DNS RPZ data plane, an ordered list of RPZs should be supplied. For each DNS RPZ in this list, it should be possible to specify an overriding policy action to be used for any policy triggers found in that RPZ. These override policies should include NXDOMAIN, NODATA, PASSTHRU, GIVEN, and CNAME. GIVEN just explicitly reaffirms the default which is to respect all policy actions found in this DNS RPZ. CNAME is an instance of "local data" which probably points to a walled garden service.

It is possible for more than one policy trigger among the various DNS RPZs connected to the name server's control plane to match a given DNS query or DNS response. The precedence rules for multiple matches are as follows:

RPZ Ordering
   Policies from RPZs defined earlier ordered set of DNS RPZs are applied instead of those defined later.

Within An RPZ
   Among policies from a single RPZ, QNAME policies are preferred over IP, IP policies are preferred over NSDNAME, and NSDNAME policies are preferred over NSIP.

Name Length
   Among applicable QNAME or NSDNAME policies within a DNS RPZ, choose the policy that matches the smallest name.

Prefix Length
Among applicable IP or NSIP policies, use the policy with the longest prefix length.

Tie Breaker
Given equal prefix lengths, use the policy that matches the smallest IP address.

By default, when a QNAME or client IP address policy is triggered and the trigger is of such high precedence that it dominates all other possible triggers (e.g. it is in the first configured policy zone), the resolver continues to check and fill its cache by recursion as if it did not already know the answer. This denies operators of authority servers for listed domains data about whether they are listed.

6. Producer Behavior

A DNS RPZ producer should make every effort to ensure that incremental zone transfer (IXFR [RFC1995]) rather than full zone transfer (AXFR [RFC5936]) is used to move new policy data toward subscribers. Also, real time zone change notifications (DNS NOTIFY [RFC1996] should be enabled and tested. DNS RPZ subscribers are "stealth slaves" as described in RFC 1996, and each such server must be explicitly denoted in the master server's configuration. Because DNS NOTIFY is a lazy protocol, it may be necessary to explicitly trigger the master server's "notify" logic after each update to the DNS RPZ. These operational guidelines are to limit policy data latency, since minimal latency is critical to both prevention of crime and abuse, and to withdrawal of erroneous or outdated policy.

In the data feed for disreputable domains, each addition or deletion or expiration can be handled using DNS UPDATE [RFC2136] to trigger normal DNS NOTIFY and subsequent DNS IXFR activity which can keep the subscribing servers well synchronized to the master RPZ. Alternatively, on some primary name servers (such as ISC BIND) it is possible to generate an entirely new primary RPZ file and have the server compute the differences between each new version and its predecessor. In ISC BIND this option is called "ixfr-from-differences" and is known to be performant even for million-rule DNS RPZ's with significant churn on a minute by minute basis.

It is good operational practice to include test records in each DNS RPZ to help that DNS RPZ's subscribers verify that response policy rewriting is working. For example, a DNS RPZ might include a QNAME policy record for BAD.EXAMPLE.COM and an IP policy record for

127.0.0.2. A subscriber can verify the correctness of their installation by querying for BAD.EXAMPLE.COM which does not exist in real DNS. If an answer is received it will be from the DNS RPZ. That answer will contain an SOA RR denoting the fully qualified name of the DNS RPZ itself.

7. History and Evolution

RPZ was previously described in a technical note from Internet Systems Consortium [ISC-RPZ]. A more up to date description was in chapter 6 of the "BIND 9 Administrator Reference Manual" [ISC-ARM].

RPZ was designed by Paul Vixie and Vernon Schryver in 2009. The initial implementation and first patch adding it to BIND was written by Vernon Schryver in late 2009. Patches for various versions of BIND9 including 9.4, 9.6, and 9.7 were distributed from FTP servers at redbarn.org and rhyolite.com 2010.

If all RPZ triggers and actions had been foreseen at the start in 2009, they would probably have been encoded differently. Instead RPZ grew incrementally, and upward compatibility required continuing support of the original encodings.

Today, with a number of commercial RPZ providers with many users and no functional problems with the encodings, any lack of aesthetic appeal is balanced by the ever increasing weight of the installed base. For example, it is impossible to replace the original QNAME trigger encoding NXDOMAIN and NODATA policy action encodings with encodings that involve rpz-* psuedo-TLDs at RPZ providers without breaking the many existing RPZ subscriber installations. The original, deprecated [Format 1] PASSTHRU encoding of a CNAME pointing to the trigger qname might still be in use in local, private policy zones, and so it is still recognized by RPZ subscriber implementations.

The initial RPZ idea was only to deny the existence of objectionable domain names, and so there were only QNAME triggers and NXDOMAIN actions. Given that single kind of trigger, encoding it as the owner name of a policy record was clearly best. A CNAME pointing to the root domain (.) is a legal and valid but not generally useful record, and so that was the encoding for the NXDOMAIN action. The encoding of the NODATA action as "CNAME *." followed similar reasoning. Requests for more kinds of triggers and actions required a more general scheme, and so they are encoded as CNAMES with targets in bogus TLDs owner names with DNS labels that start with "rpz_".

8. IANA Considerations

No actions are required from IANA as result of the publication of this document.

9. Security Considerations

RPZ is a mechanism for providing "untruthful" DNS results from recursive servers. However, RPZ does not increase the intrinsic DNS vulnerabilites at recursive servers to falsifying DNS data. RPZ merely formalizes and facilitates modifying DNS data on its way from DNS authority servers to clients. Moreover, DNSSEC (see RFC 4033 [RFC4033] and RFC 4034 [RFC4034]) prevents changes to DNS data by RPZ.

By default, DNS resolvers using RPZ do not modify DNS results when DNSSEC signatures are available and requested by the DNS client. When the common "BREAK-DNSSEC" configuration setting is used, RPZ using resolvers ignore DNSSEC. The result of "BREAK-DNSSEC" at DNS clients using DNSSEC is functionally similar to an RPZ NXDOMAIN policy action; the DNS client is blocked from malefactor domains.

The policy zones might be considered sensitive, because they contain information about malefactors. Like other DNS zones in most situations, RPZs are transferred from sources to subscribers as cleartext vulnerable to observation. However, TSIG transaction signatures [RFC2845] SHOULD be used to authenticate and protect RPZ contents from modification.

Recursive servers using RPZ are often configured to complete recursion even if a policy trigger provides a rewritten answer without needing recursion. This impedes malefactors observing requests from their own authority servers from inferring whether RPZ is in use and whether their RRs are listed. "qname-wait-recurse" is a common configuration switch that controls this behavior.

10. References

10.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <http://www.rfc-editor.org/info/rfc1034>.

[RFC1995] Ohta, M., "Incremental Zone Transfer in DNS", RFC 1995, DOI 10.17487/RFC1995, August 1996, <http://www.rfc-editor.org/info/rfc1995>.

[RFC1996] Vixie, P., "A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)", RFC 1996, DOI 10.17487/RFC1996, August 1996, <http://www.rfc-editor.org/info/rfc1996>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2136] Vixie, P., Ed., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, DOI 10.17487/RFC2136, April 1997, <http://www.rfc-editor.org/info/rfc2136>.

[RFC2845] Vixie, P., Gudmundsson, O., Eastlake 3rd, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, DOI 10.17487/RFC2845, May 2000, <http://www.rfc-editor.org/info/rfc2845>.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <http://www.rfc-editor.org/info/rfc4033>.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <http://www.rfc-editor.org/info/rfc4034>.

[RFC5936] Lewis, E. and A. Hoenes, Ed., "DNS Zone Transfer Protocol (AXFR)", RFC 5936, DOI 10.17487/RFC5936, June 2010, <http://www.rfc-editor.org/info/rfc5936>.

10.2. Informative References

[ISC-ARM] Internet Systems Consortium, "BIND 9 Administrator Reference Manual, https://ftp.isc.org/isc/bind9/cur/9.10/doc/arm/Bv9ARM.ch06.html#rpz", 2016.

[ISC-RPZ] Vixie, P. and V. Schryver, "DNS Response Policy Zones (DNS RPZ, Format 3), https://ftp.isc.org/isc/dnsrpz/isc-tn-2010-1.txt", 2010.

[ISC-RRL] Vixie, P. and V. Schryver, "DNS Response Rate Limiting (DNS RRL), https://ftp.isc.org/isc/pubs/tn/isc-tn-2012-1.txt", 2012.

Appendix A. Examples

An existing data feed capable of producing an RHSBL can be trivially used to generate a DNS RPZ. If the desired policy is to alias targeted domains to a local walled garden, then for each domain name, generate the following records, one for the name itself and perhaps also one for its sub-domains:

```
bad.domain.com    CNAME   walled-garden.example.net.
*.bad.domain.com  CNAME   walled-garden.example.net.
```

If it is desirable to return NXDOMAIN for each domain (and its sub-domains in this example), try this:

```
bad.domain.com    CNAME   .
*.bad.domain.com  CNAME   .
```

If there are specific walled gardens for mail versus everything else:

```
bad.domain.com    MX      0 wgmail.example.net.
bad.domain.com      A     192.168.6.66
*.bad.domain.com    MX    0 wgmail.example.net.
*.bad.domain.com    A     192.168.6.66
```

An extended example follows:

```
         $ORIGIN rpz.example.com.
         $TTL 1H
         @                  SOA LOCALHOST. named-mgr.example.com. (
                                1 1h 15m 30d 2h) NS LOCALHOST.

         ; QNAME policy records.
         ; There are no periods (.) after the relative owner names.
         nxdomain.domain.com CNAME .               ; NXDOMAIN policy
         nodata.domain.com   CNAME *.              ; NODATA policy

         ; redirect to walled garden
         bad.domain.com    A       10.0.0.1
                           AAAA    2001:2::1

           ; do not rewrite OK.DOMAIN.COM (so, PASSTHRU)
           ok.domain.com       CNAME   rpz-passthru.
           bzone.domain.com    CNAME   garden.example.com.

           ; redirect X.BZONE.DOMAIN.COM to
           ; X.BZONE.DOMAIN.COM.GARDEN.EXAMPLE.COM
           *.bzone.domain.com  CNAME   *.garden.example.com.

           ; rewrite all answers for 127/8 except 127.0.0.1
           8.0.0.0.127.rpz-ip  CNAME   .
           32.1.0.0.127.rpz-ip CNAME   rpz-passthru.

           ; rewrite to NXDOMAIN all responses; for domains for which
           ; NS.DOMAIN.COM is an authoritative DNS server or a server
           ; for a parent) or that have an authoritative server
           ; in 2001:2::/48
           ns.domain.com.rpz-nsdname   CNAME   .
           48.zz.2.2001.rpz-nsip       CNAME   .
```

Authors' Addresses

Paul Vixie
Farsight Security, Inc.

Email: paul@redbarn.org

Vernon Schryver
Rhyolite Software

Email: vjs@rhyolite.com