DNS Operations
Internet-Draft
Expires: September 5, 2007

S. Krishnaswamy
SPARTA Inc.

March 4, 2007

Split-View DNSSEC Operational Practices
draft-krishnaswamy-dnsop-dnssec-split-view-04.txt

Status of this Memo

Copyright Notice

Abstract

The security extensions to the Domain Name System (DNSSEC) allow for integrity protection, whereby it is possible to make a determination of the verity of data returned from the Domain Name System in response to a query. Current operation of the Domain Name System also allows for the creation of multiple views of data, where the answer returned in response to a query is dependent on the origin of the query. Data integrity and the ability to return possibly conflicting values as in split-views may be construed to be mutually conflicting goals; but this apparent dichotomy is resolvable in

practice through careful configuration. This document provides recommendations for configuring a manageable split-view DNSSEC environment in a representative enterprise network.

Table of Contents

1. Introduction 3
2. Considerations for split-views 4
2.1. Split-Views and Name Hiding 4
2.2. Representative Network Structure 5
2.3. Special Capabilities in Network Devices 5
3. Generic Split-View DNS Architecture 5
3.1. Query Channeling 6
3.2. Controlling Errant Queries 8
3.3. Name Server Requirements for Split-Views 8
3.3.1. Internal Recursive Forwarder 8
3.3.2. Boundary Recursive Name Server 8
3.3.3. Authoritative Internal and External-View Name Servers 9
4. Split-View DNSSEC 9
4.1. Approaches 9
4.1.1. Same Key Signing 10
4.1.2. Partial Decoupling of Authentication Chains 11
4.1.3. Multiple DS Records 13
4.1.4. Complete Decoupling of Authentication Chains 14
4.1.5. No Internal Validation 16
4.2. Name Server Requirements for Split-View DNSSEC 17
5. IANA Considerations 17
6. Security Considerations 17
7. Acknowledgements 18
8. References 18
8.1. Normative References 18
8.2. Informative References 19
Appendix A. Packet Filtering Considerations 19
A.1. Inner Packet Filter 19
A.2. Outer Packet Filter 20
Author's Address 21
Intellectual Property and Copyright Statements 23

1. Introduction

Split-view DNS is the term used to describe the behavior where the DNS returns different responses to the same query based upon the query source address. It is also a network management technique that can be used to restrict DNS names to only those segments or views of the network that need to see these names.

The security extensions to DNS (commonly labeled "DNSSEC") [1] provide for origin authenticity and data integrity. These properties are determined by validating the authentication chain from some trust anchor configured at the validating resolver to a signed record.

The combination of DNSSEC with split-views raises some potential issues and concerns. In the case of split-view DNS every authentication chain in every view must validate properly. Names that are common in different views may contain different data for the same resource record type in each of these views. Cache contamination is a possibility when data from the wrong view is returned in response to a query. For "private" views, or views of the DNS that contain data not available through the public DNS, building an authentication chain from a public trust anchor to data in the private view of a zone can be especially problematic, caching problems notwithstanding.

This document describes a configuration that allows for the co-existence of split-views with DNSSEC. Section 3 describes a generic two-level recursive scheme where an Internal Recursive Forwarder resolves internal answers from internal authoritative name servers and marshalls all other queries to a Boundary Recursive Name Server. TSIG between the Internal Recursive Forwarder and the Boundary Recursive Name Server protects against errant queries.

Section 4 provides multiple choices for configuring the trust anchor while using DNSSEC. This list of choices provides trade-offs between configuration overhead and validation overhead. Trading-off in favour of minimal operator overhead is useful for overall maintainability of the system, especially when split-view DNS is considered in the context of validating stub resolvers that are mobile across the two views.

In cases where the different views of DNS information correspond to different physical networks, the name servers authoritative for the internal and external views of data are often separated by a firewall. This document provides recommendations for reducing the impact of errant queries in the split-view DNS setup and also makes recommendations in Appendix A for DNS-related packet filtering rules required to support the proper operation of the suggested

configuration.

Although closely tied to it, this document must not be viewed as an endorsement of split-views technique in itself; this document only provides recommendations for DNSSEC in such environments while avoiding some of the common pitfalls that are possible in this setup. The approach given in this document is adjustable for various operational and security consideration levels. Network architects may implement other mechanisms for DNSSEC split-views however they must carefully consider the ramifications of any changes with respect to the base architecture given in this document.

2. Considerations for split-views

2.1. Split-Views and Name Hiding

Although primarily meant to be a network management technique, the use of split-views is also seen by some as an approach for improving an organization's security posture - by preventing the exposure of internal host names, knowledge of whose existence is deemed to be sensitive.

Relying solely on split-view DNS to protect sensitive hosts from attacks has proven to be insufficiently adequate in the past and is not recommended. Attack vectors in Internet exploits have been able to successfully infect hosts with or without their IP addresses being published in the DNS. Conversely, publishing the IP addresses of hosts that are otherwise secured does not necessarily increase their vulnerability to these attacks.

Name hiding through split-view DNS is primarily useful as part of a more comprehensive defense-in-depth strategy to provide one line of defense against name-based attacks. Split-views are not recommended by themselves as a name-hiding approach. A better alternative for protecting a namespace of sensitive hosts is to have that entire namespace reside within a private delegation. By doing so, it is possible to have the protection given to the name server that serves these names commensurate with the protection given to the hosts themselves. Since hosts in the private branch are explicitly marked as such by virtue of their domain name, this method also allows the network administrator to better classify hosts as being public or private and lessens the opportunity for sensitive hosts to be inadvertantly placed in public domains. Private delegations are useful when name hiding is the only reason for namespace separation. They do not, however, allow for transparency during name resolution; queries have to be made for specific names in specific views.

2.2. Representative Network Structure

The architecture for split-views in this document is modeled around a representative enterprise structure as described below.

The two views are for the internal and external segments of the network, with the external segment residing within the boundary network. The external view of the DNS is also the "global" view of the enterprise to the outside world. The two networks are separated by a packet filtering firewall. A packet filtering firewall also separates the boundary network from the external Internet. Name hiding is not an objective of this split-view setup, but avoiding cache contamination is. Although the two concepts are related, split-views is not recommended for hiding sensitive names because of the ease with which names can be leaked out (through email headers, for example). Again, if name hiding is the main objective, the approach of using a private delegation for sensitive names is strongly encouraged.

2.3. Special Capabilities in Network Devices

Some name server implementations directly support split-view DNS as a configuration feature. However, the options for placement of such devices in the representative architecture suggested above are rarely optimal.

Split views can also be constructed using a single multi-homed device in lieu of the multiple machines suggested in this document -- an approach typically used in a proxy-firewall setting. The device is configured as the authoritative name server for both views of data by running multiple instances of the name server process and answering differently based on the query source.

This document describes a generic architecture for split-view DNS without relying on any of the above special capablities from any network device or name server implementation. Ergo, the number of distinct devices that provide the name resolution function is increased. Networks that use a reduced set of network devices for providing the split-view functionality may still draw from recommendations given in this document, provided that the ramifications of any change (if any) with respect to the base architecture given in this document are well understood.

3. Generic Split-View DNS Architecture

Split-views can be implemented using an approach known as "query channeling". Query channeling involves carefully controlling the

security-aware validating name server path for fetching responses from different name servers so that cache contamination is avoided. Figure 1 illustrates this setup.

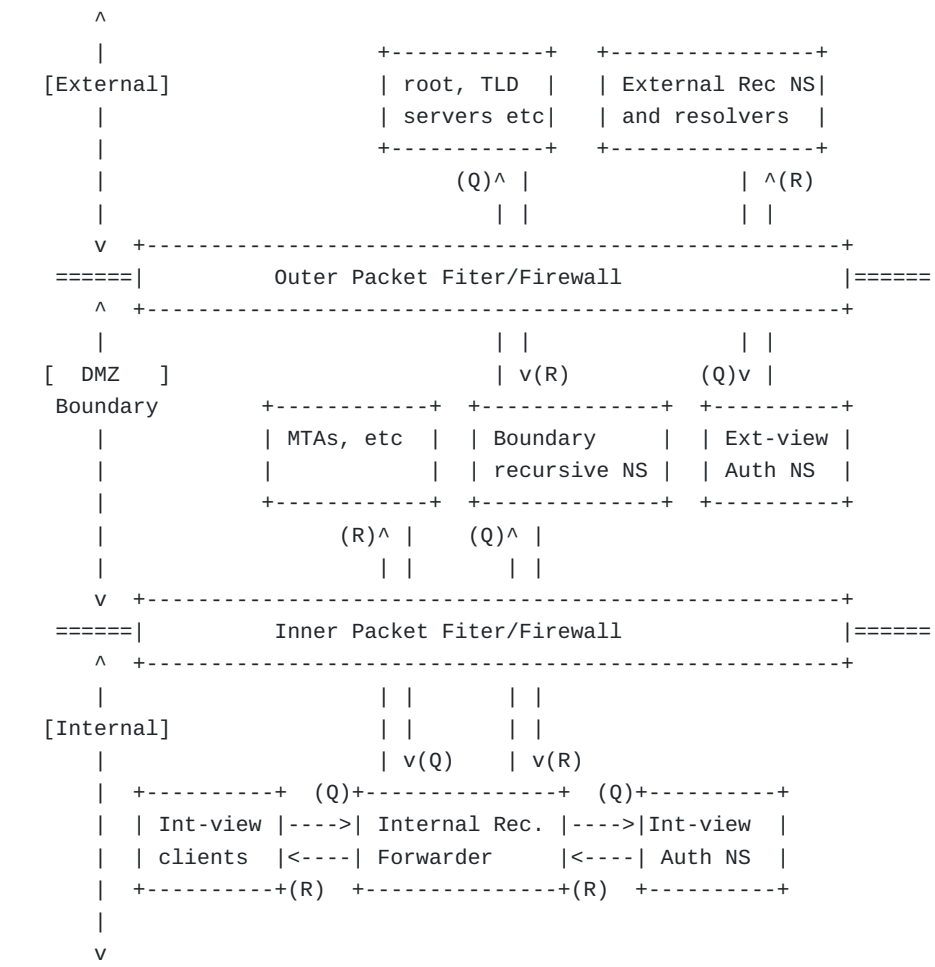

Figure 1: Split-View Architecture

3.1. Query Channeling

Resolution of external data by clients in the external segment of the network or in the global Internet is straightforward since queries follow their normative paths. The configuration in Figure 1 uses a combination of multiple name servers and query forwarding to return answers for clients in the internal segment of the network.

One server functions as an Internal Recursive Forwarder and is responsible for answering queries for clients on the internal segment of network. Some appliances in the boundary network (such as Mail

Transfer Agents, MTAs) may directly query the Internal Recursive Forwarder for the internal view of certain names.

None of the name servers in this architecture are simultaneously recursive and authoritative. Separation of authoritative and recursive name servers not only provides simple role separation, but is also widely recognized as a security measure in DNS for protecting authoritative name servers against compromised caches.

The Internal Recursive Forwarder forwards any query for internal data to its corresponding Internal-view Authoritative Name Server while recursively obtaining any other data from a Boundary Recursive Name Server. The Internal-View Authoritative Name Servers do not perform recursion. In cases where queries can only be forwarded per zone, the Internal Recursive Forwarder recursively obtains answers for any names that lie within delegations under the forwarded zones, so queries for child names do not require further forwarding. However, if forwarding is only possible per domain, the Internal Recursive Forwarder must explicitly forward queries for every internal delegation to its respective authoritative name server. This rule can be relaxed while forwarding queries to name servers that are simultaneously authoritative for the child as well as the parent zone.

The Boundary Recursive Name Server is a simple caching name server that obtains answers from the outside, but only if asked by the Internal Recursive Forwarder. Separation of the Internal Recursive Forwarder from the Boundary Recursive Name Server allows queries to and responses from the outside network to be channeled (and mediated) through the boundary network.

The Internal Recursive Forwarder and the Internal-view Authoritative Name Servers reside in the internal network; the Boundary Recursive Name Server and the External-view Authoritative Name Servers reside in the boundary network. The Boundary Recursive Name Server and the Internal Recursive Forwarder may alternatively be implemented as an integral part of the device that provides packet-filtering or firewalling services between the internal network and the boundary network, provided that they execute as distinct and well-separated processes within this device.

The two-level recursive scheme controls the name servers to which queries are directed to. Since queries for internal data are sent to authoritative name servers which are not recursive, this scheme also controls the data reception path. In this way internal data can be kept totally separate from external data, thus preventing cache contamination.

3.2. Controlling Errant Queries

DNS queries that are sent from the Internal Recursive Forwarder for global DNS data should only be directed towards the Boundary Recursive Name Server. Since the Boundary Recursive Name Server has no knowledge of internal-view data, internal clients must not use it directly for resolving any queries. Only properly configured Internal Recursive Forwarders that are approved to send queries to this name server must do so.

It is useful to note that the Internal Recursive Forwarder must not attempt to recursively answer queries if the Internal-view Authoritative Name Server fails to respond. If it did so, external data could be returned in such circumstances and lead to cache contamination.

TSIG [3] is the recommended method for controlling which name servers are approved for sending queries to the Boundary Recursive Name Server. Alternatively, configuring the Boundary Recursive Name Server such that it only answers queries for the Internal Recursive Forwarder(s), and supporting this configuring with an appropriate set of rules in the packet filter is also possible.

Some devices such as MTAs that reside in the Boundary network may also directly query the Internal Recursive Forwarder for internal data. This must be configured via an appropriate set of rules at the packet filter between the Internal Network and the Boundary Network.

3.3. Name Server Requirements for Split-Views

This section summarizes the list of requirements for the various name servers involved in the split-view configuration.

3.3.1. Internal Recursive Forwarder

- Ability to forward queries for names in specific zones or domains to specific name servers.
- Ability to control forwarding behaviour such that no further attempt to resolve the query is made if the name server(s) that queries are normally forwarded to fails to respond.
- Ability to recursively answer queries.
- Ability to authenticate (and verify the authenticity of) messages using TSIG.

3.3.2. Boundary Recursive Name Server

- Ability to recursively answer queries.
- Ability to authenticate (and verify the authenticity of) messages using TSIG.
- Ability to filter incoming queries based on the TSIG key used to authenticate the message.

3.3.3. Authoritative Internal and External-View Name Servers

- Ability to authoritatively answer queries for a zone.
- Ability to disable all recursive behaviour.

4. Split-View DNSSEC

The DNSSEC concern for split-view is ensuring that both the internal and the external authentication chains validate properly. While validating external data is relatively straightforward, there are multiple approaches that can be used for validating internal data. The method of choice depends on the percieved threat environment for the internal view, the amount of end-resolver configuration overhead and the ability to support administrative separation between the two split-views.

The configuration overhead at end resolvers is mainly associated with the task of managing trust anchors. Having fewer keys is desirable since it makes key management easier. It is also desirable to reduce the amount of reconfiguration required for validating stub resolvers that are mobile across the two views of data, while still being able to tie an answer to a particular view when troubleshooting this setup. In many cases two views of a split zone are administered separately, so the ability to configure and use different zone signing keys for the different views is also useful.

4.1. Approaches

The different options for internal data validation, with recommendations on when these options must (and must not) be used, are further outlined below. In each of these options, the zone that is split is assumed to be a delegation under example.com and unless explicitly mentioned, configuration of trust anchors (and hence validation of internal data) occurs at the Internal Recursive Forwarder and validating stub resolvers. Validating stub resolvers that operate in the internal segment of the network must still resolve DNS queries through the Internal Recursive Forwarder.

4.1.1. Same Key Signing

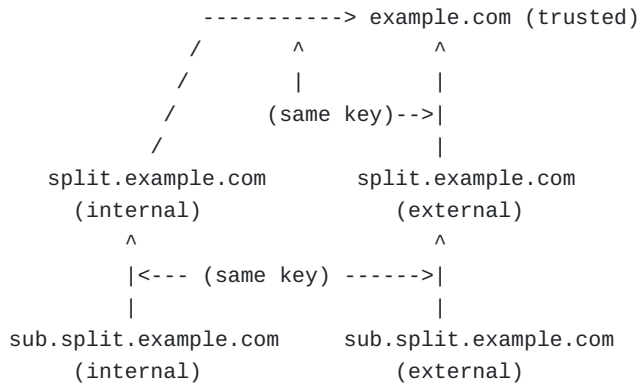

Figure 2: Same Key Signing

DESCRIPTION:

The same keys are used to sign corresponding zone data in both views of the split name space. The DS, NS and glue records at example.com all correspond to the external view data. Since the keys referenced in the DS record at example.com are present in the apex key-set of both views, the authentication chain can always be completed. The trust anchor at the Internal Recursive Forwarder and any validating stub resolver can be configured at any level of the zone hierarchy.

ADVANTAGES:

- Trust Anchor management is simple in this approach. A single trust anchor is sufficient to validate answers in both views. There is also more flexibility in chosing the zone level at which the Trust Anchor is defined.
- Validating stub resolvers on hosts that are mobile across views can validate answers in both views without having to change their trust anchor.
- Key management is straightforward for administrators since the same keys are used for signing the internal and external views.

DISADVANTAGES:

- Although easy to setup, this approach can be difficult to troubleshoot. There is no easy way to identify if the record obtained for a query corresponds to the internal view or the external view.
- Using the same key makes administrative separation of the two views of data difficult.

- Cache contamination caused by the insertion of data from the external view for data with the same name in the internal view (or vice-verca) cannot be automatically detected.

RECOMMENDED USAGE SCENARIO:

- This approach must be only used when all zones covered by the split are administered by the same individual and key management overhead needs to be kept to a minimum.

4.1.2. Partial Decoupling of Authentication Chains

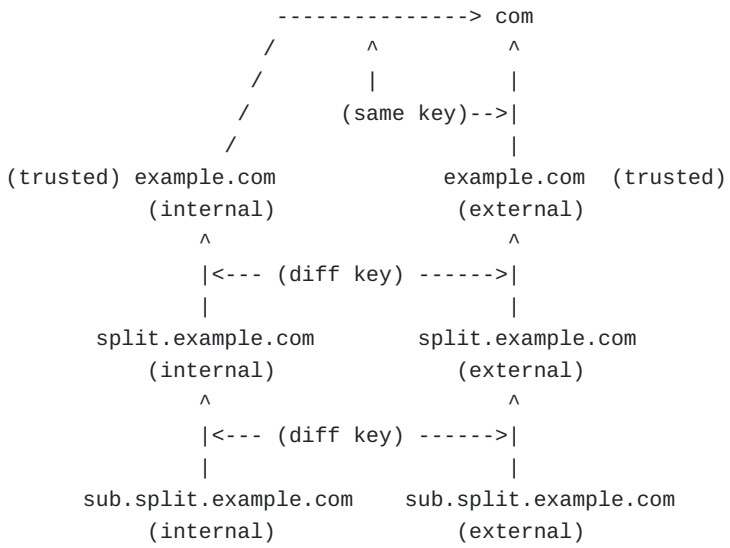

Figure 3: Partial Decoupling of Authentication Chains

DESCRIPTION:

In this approach example.com zone is also split into two views. Different keys are used for signing each view of the split below (but not including) example.com such that the authentication chains for the internal and external zones share no common records that may cause any ambiguity. The trust anchor is configured at the level of example.com or higher. The apex keyset in example.com is the same for both views, but the glue and DS information for delegations under example.com is different across views. An internal name server is configured as the authoritative server for the internal view of the split example.com zone and the Internal Recursive Forwarder is modified to forward all internal queries for this zone to it.

ADVANTAGES:

- Trust Anchor management is simple in this approach. A single trust anchor is sufficient to validate answers in both views.
- Validating stub resolvers on hosts that are mobile across views can validate answers in both views without having to change their trust anchor.
- Having separate keys for the two views of data is useful for troubleshooting and in determining which view a given record belongs to.
- The approach allows for administrative separation between the different views.
- As long as all members of the authentication chain are not inserted from one view to the other, cache contamination caused by the insertion of data from the external view for data with the same name in the internal view (or vice-verca) can be automatically detected.

DISADVANTAGES:

- Key management is onerous since different keys are used to sign data in different views.
- This approach relies on the presence of an additional view of the example.com zone where relevant records are duplicated. The number of such records is typically not very large, but the overhead and complexity in maintaining duplicate records can, in some cases, be a burden. This approach must only be used if this duplication is administratively viable.
- Cache contamination caused by the insertion of data from the external view for data with the same name in the internal view (or vice-verca) cannot be automatically detected when all elements of the authentication chain are inserted.

RECOMMENDED USAGE SCENARIO:

- This approach may be used if internal and external views are administered separately. This approach may also be used when there is a single administrator, if the administrator has a well-defined process in place for managing DNS keys.
- This approach is not recommeded if creating a split view of the parent (example.com) is not administratively viable.

4.1.3. Multiple DS Records

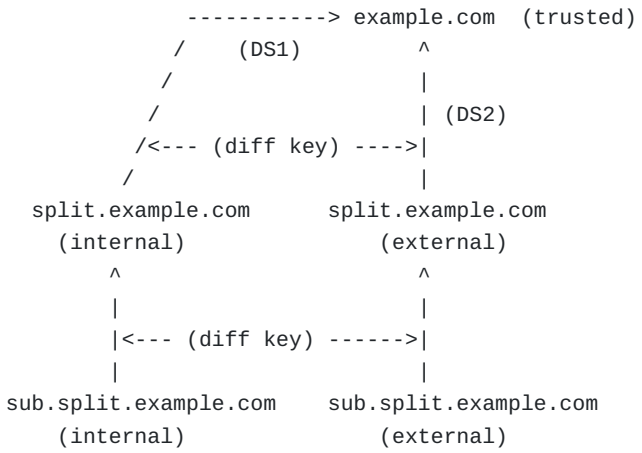

Figure 4: Multiple DS Records

DESCRIPTION:

This approach is simliar to Section 4.1.2. However, in this approach, example.com is not split. DS records corresponding to each of the two different views of split.example.com are published at the delegation point. The glue and NS records at the delegation point still correspond to the external view of split.example.com but this information is never used by the internal zone. The trust anchor is configured at the level of example.com or higher. The authentication chain from this trust anchor to data in either zone is automatically constructed by using one of the two DS records, which ever is applicable at that view.

ADVANTAGES:

- Trust Anchor management is simple in this approach. A single trust anchor is sufficient to validate answers in both views.
- Validating stub resolvers on hosts that are mobile across views can validate answers in both views without having to change their trust anchor.
- Having separate keys for the two views of data is useful for troubleshooting and in determining which view a given record belongs to.
- The approach allows for administrative separation between the different views.
- As long as all members of the authentication chain are not inserted from one view to the other, cache contamination caused by the insertion of data from the external view for data with the same name in the internal view (or vice-verca) can be

automatically detected.

DISADVANTAGES:

- Key management is onerous since different keys are used to sign data in different views.
- While most of the internal zone contents can be kept private to the internal view, the DS record must still be exposed. Since data hiding is not an objective of the split-view setup this is not really a problem in most instances.
- An attendent problem with multiple DS records is that since the validation algorithm iteratively verifies parent DS records while trying to complete the authentication chain, there is some added computational overhead, which increases as the number of DS records in the delegation point grows.
- Cache contamination caused by the insertion of data from the external view for data with the same name in the internal view (or vice-verca) cannot be automatically detected when all elements of the authentication chain are inserted.

RECOMMEDED USAGE SCENARIO:

- This approach may be used if internal and external views are administered separately. This approach may also be used when there is a single administrator, if the administrator has a well-defined process in place for managing DNS keys.
- Since some coordination between split.example.com and its parent, example.com, is required to publish multiple DS records, this approach is most suitable when the split is made at a level lower than the organization apex.
- This approach must not be used if exposure of the DS record for the internal view of the split view is considered to be a problem.

4.1.4. Complete Decoupling of Authentication Chains

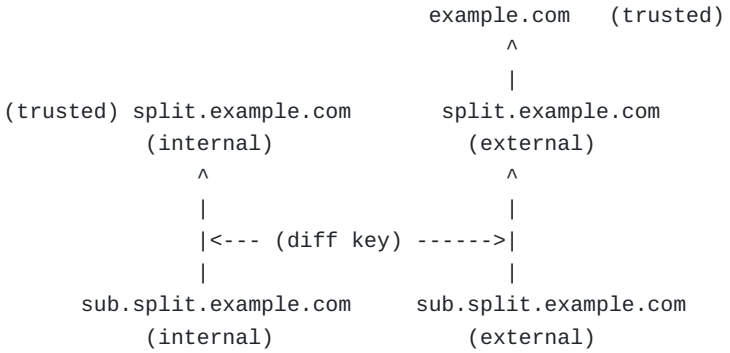

Figure 5: Complete Decoupling of Authentication Chains

DESCRIPTION:

This approach makes it possible to create multiple split views without having to split or modify the parent zone data in any manner. It does so at the expense of increased administrator overhead. In this approach different keys are used for signing each view of the split below (and including) split.example.com such that the authentication chains for the internal and external zones share no common records that may cause any ambiguity. A new trust anchor is configured for each internal view of the split zone in addition to any existing trust anchors for the outer zone data.

ADVANTAGES:

- This approach allows for administrative separation between the different views.
- Having separate keys for the two views of data is useful for troubleshooting and in determining which view a given record belongs to.
- This approach can automatically detect cases where data from the external zone has been inserted into the internal network.

DISADVANTAGES:

- Key management is onerous since different keys are used to sign data in different views.
- Trust Anchor management is onerous since multiple trust anchors need to be configured at validating resolvers.
- Validation on moible hosts containing stub resolvers may not be seamless. The outcome of validation in the external view may be impacted by the local policy on the validator, which decides how multiple trust anchors are evaluated. Further, in a working setup involving multiple trust anchors, cache contamination cannot be detected automatically. In order to have predictable results, the trust anchors configured on such validators must be constantly changed whenever they move across views.

RECOMMEDED USAGE SCENARIO:

- This approach may be used if the two views are signed and the approaches specified in Section 4.1.2 and Section 4.1.3 are not feasable.
- This approach may also be used in scenarios where DNSSEC deployment schedules for the two split views are different and the internal view data is signed and deployed long before the external view data. DNSSEC deployment in the two views can occur independently.

- This option may also be used when accidental or malicious insertion of DNS data across views is a concern.

4.1.5. No Internal Validation

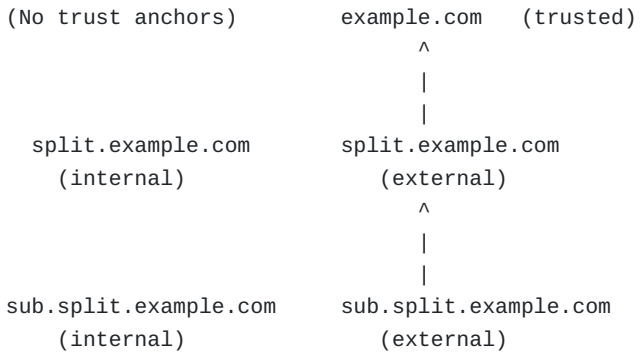

Figure 6: No Internal Validation

DESCRIPTION:

The Internal Recursive Forwarder does not perform any DNSSEC validation for internal view data. DNSSEC is enabled but no trust anchors are configured at the Internal Recursive Forwarder and validating stub resolvers. Instead, trust anchors are configured at the Boundary Recursive Name Server for validating global data queried by clients in the internal view of the network. The Internal Recursive Forwarder does not return answers that do not contain the AD bit from the Boundary Recursive Name Server to internal clients. The organizational network is assumed to be secure, so no additional last-mile protection of DNSSEC results between the Boundary Recursive Name Server and the Internal Recursive Forwarder is needed.

It should be noted that disabling DNSSEC validation in the internal view can also be accomplished by selectively turning "off" DNSSEC at the Internal Recursive Forwarder for all zones under and including the internal view of the split namespace if this feature is available. However, this has the drawback that such features would have to be additionally supported by and configured at every validating stub resolver.

ADVANTAGES:

- Trust Anchor management is simple in this approach. Trust anchors are only needed for validating external view data.
- Key management is straightforward for administrators since keys are not required for the internal view.

DISADVANTAGES:

- Requires additional configuration on validating resolvers to add and remove trust anchors as they move from the internal to external view and vice-versa.

RECOMMENDED USAGE SCENARIO:

- This is an option if the organizations internal and boundary network are considered safe -- the threat environment for the internal zone in this scenario does not include DNS compromise.
- This approach may be used in scenarios where DNSSEC deployment schedules for the two split views are different and the internal view data is signed and deployed long after the external view data.

4.2. Name Server Requirements for Split-View DNSSEC

In order to support split-views with DNSSEC, in addtion to the list of requirements specified in Section 3.3, all name servers involved in the split-view configuration must conform to the specifications given in [2]. DNSSEC support must be enabled on all of these name servers. Additionally, the Internal Recursive Forwarder must support the following features:

- Ability to validate DNSSEC responses.
- Support for configurable DNSSEC trust anchors. It should be possible to configure more than one trust anchor.
- Ability to accept or discard reponses from the Boundary Recursive Name Server based on the AD bit as described in Section 4.1.5.

5. IANA Considerations

This document has no actions for IANA.

6. Security Considerations

The configuration suggested in this document tries to minimize cache contamination, but misconfigurations are still easily possible. Any misconfigurations in the three different types of name servers or the two packet filters can result in cache contamination and cause incorrect or inconsistent results to be returned between views. The validating resolver may or may not detect this depending on the manner in which secure entry points to split zones are defined and which trust anchors are configured. Confidentiality loss caused by leakage of information in this context is not an issue since split-

views by itself is not meant to provide this functionality. In name-hiding is the objective, split-views can (and must) be avoided and the alternative scheme of using different names for internal and external domains must be used instead.

An improperly configured packet filter that allows errant DNS traffic through or denies legitimate responses can lead to aggressive retransmission of queries by resolvers to name servers, leading to increased query load at such name servers.

Each of the validation options outlined in Section 4 also introduce their own security considerations. Not using DNSSEC in the internal view creates the possibility of a malicious entity supplying bogus information in response to queries, without detection. Using a common key between both views of the split does not allow one to differentiate between internal and external data and troubleshooting is greatly encumbered. On the other hand, using a multitude of keys at validating resolvers increases the operator overhead and thus the chances for configuration errors. All approaches that use a common key for validating internal and external data are unable to automatically detect cache contamination, so they may escape the attention of a system administrator until troubleshooting begins. Approaches using a common trust anchor are also susceptible to an insider attack where data from one view injected into a response for queries for data in the other the other view can corrupt the cache within the Internal Recursive Forwarder.

7. Acknowledgements

The contributions, suggestions and remarks of the following persons to this draft are particularly acknowledged: Wes Griffin, John Kelley, Ed Lewis, Russ Mundy, Scott Rose, Mike StJohns, Char Sample, Andrew Sullivan, Howard Eland, Wes Hardaker and Robert Story. The two-level name server scheme described in this document builds upon work that was originally performed by Ed Lewis.

8. References

8.1. Normative References

[1] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[2] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions",

RFC 4035, March 2005.

[3] Vixie, P., Gudmundsson, O., Eastlake 3rd, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, May 2000.

8.2. Informative References

[4] Larson, M. and P. Barber, "Observed DNS Resolution Misbehavior", Work in Progress ietf-dnsop-bad-dns-res, July 2005.

Appendix A. Packet Filtering Considerations

Among some of the frequently observed DNS resolution misbehaviour [4] is the problem of resolvers aggressively retransmitting queries from behind misconfigured firewalls that allow queries out, but drop all returned responses. This problem is exacerbated by a handful of errant queries that are sent by only a subset of internal resolvers, which makes problem isolation extremely difficult. The following subsections define the rules that must be configured in the two packet filters depicted in Figure 1 in order to support the split-view configuration.

A.1. Inner Packet Filter

In order to allow the configuration described in Section 3 to work, any packet filtering system between the internal network and the boundary network must ALLOW all of the following types of packets.

- DNS queries from any internal address to the Boundary Recursive Name Server (finer level access control is done by TSIG):

Proto	SrcIP	SrcPort	DestIP	DstPort	AckBit
UDP	internal	>1023	boundary rec srv	53	N/A
UDP	internal	53	boundary rec srv	53	N/A
TCP	internal	>1023	boundary rec srv	53	Any

- Responses to the above queries from the Boundary Recursive Name Server to any internal address:

Proto	SrcIP	SrcPort	DestIP	DstPort	AckBit
UDP	boundary rec srv	53	internal	>1023	N/A
UDP	boundary rec srv	53	internal	53	N/A
TCP	boudnary rec srv	53	internal	>1023	Set

- Queries from devices in the boundary network (such as MTAs) to any internal name server:

Proto	SrcIP	SrcPort	DestIP	DstPort	AckBit
UDP	boundary device	>1023	internal	53	N/A
TCP	boundary device	>1023	internal	53	Any

- Responses to the above queries from the (any) recursive forwarder to devices that reside in the boundary network:

Proto	SrcIP	SrcPort	DestIP	DstPort	AckBit
UDP	internal	53	boundary device	>1023	N/A
TCP	internal	53	boundary device	>1023	Set

The packet filtering system between the internal network and the boundary network must finally DROP any DNS packets not covered by the above rules.

Proto	SrcIP	SrcPort	DestIP	DstPort	AckBit
UDP	Any	53	Any	Any	N/A
UDP	Any	Any	Any	53	N/A
TCP	Any	53	Any	Any	Any
TCP	Any	Any	Any	53	Any

A.2. Outer Packet Filter

Any packet filtering system configured between the boundary network and the external network needs to ALLOW the following.

- Queries from the Boundary Recursive Name Server network to the external network:

Proto	SrcIP	SrcPort	DestIP	DstPort	AckBit
UDP	boundary rec srv	>1023	Any	53	N/A
UDP	boundary rec srv	53	Any	53	N/A
TCP	boundary rec srv	>1023	Any	53	Any

- Responses to the above queries from the outside to the Boundary Recursive Name Server:

Proto	SrcIP	SrcPort	DestIP	DstPort	AckBit
UDP	Any	53	boundary rec srv	>1023	N/A
UDP	Any	53	boundary rec srv	53	N/A
TCP	Any	53	boundary rec srv	>1023	Set

- Queries from outside resolvers to the external-view authoritative servers:

Proto	SrcIP	SrcPort	DestIP	DstPort	AckBit
UDP	Any	>1023	auth serv(ext view)	53	N/A
UDP	Any	53	auth serv(ext view)	53	N/A
TCP	Any	>1023	auth serv(ext view)	53	Any

- Responses to the above queries from the external-view authoritative server to the outside:

Proto	SrcIP	SrcPort	DestIP	DstPort	AckBit
UDP	auth serv(ext view)	53	Any	>1023	N/A
UDP	auth serv(ext view)	53	Any	53	N/A
TCP	auth serv(ext view)	53	Any	>1023	Set

The packet filtering system between the boundary network and the externally network must finally DROP any DNS packets not covered by the above rules.

Proto	SrcIP	SrcPort	DestIP	DstPort	AckBit
UDP	Any	53	Any	Any	N/A
UDP	Any	Any	Any	53	N/A
TCP	Any	53	Any	Any	Any
TCP	Any	Any	Any	53	Any

Author's Address

Suresh Krishnaswamy
SPARTA Inc.
7110 Samuel Morse Dr.
Columbia, MD 21046
US

Email: suresh AT sparta DOT com
URI: http://www.sparta.com

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).