DNSIND WG
INTERNET DRAFT
May Update: RFC 2535
Catagory: I-D

Edward Lewis
TIS Labs
Jerry Scharf
ISC

April 1, 1999

The Zone Key Referral

<draft-ietf-dnsind-keyreferral-00.txt>

Status of this Memo

Comments should be sent to the authors or the DNSIND WG mailing list <namedroppers@internic.net>.

This draft expires on October 1, 1999

Copyright Notice

Notes on this document

This section will only appear in the -00.txt edition of this draft.

This document originated in the DNSSEC working group in June 1998. The discussion of the issues in this draft were tabled until the publication of the then current DNSSEC drafts as RFCs.

The first version of this document lists a third author, John Gilmore. He is listed as an author because he was one of the initiators of what is proposed. In this and following versions he is only listed in the Acknowledgements (as opposed to being an author) as he has not been involved in the writing/editing of the draft. This has been done to avoid assigning his name to a document he may not have a chance to read, this is not intended as a slight on his efforts.

When commenting on this draft, please be aware that some terms used here are up for negotiation before progressing - such as "thief" and "road block" appearing later in the draft. Comments which are left justified were added during the re-issuing of the draft, they add context that may have been lost over time.

Abstract

A new type of key is defined to address the problems of performance in large delegeted zones and issues of liability of registrars with regards to the storing of public keys belonging to zone cuts. This new key type also brings DNSSEC more in line with the DNS treatment of zone cuts and speeds recovery in handling privatekey exposure.

The new type of key is a referral record that is stored, signed, at the parent zone's place for the delegation point. A resolver receiving this record is being informed that there are genuine public keys at the child's authoritative name servers. The parent no longer needs to store the child's public keys locally.

1 Introduction

There are a number of different reasons for the proposal of this new key type. Reasons include:
o the performance impact that RFC 2535 has on name servers
o the problem of updating a widely delegated parent zone on demand
o statements in RFC 2181 on authoritative data at delegations
o perceived liability of the operator of a name server or registry

To address these issues, which are expanded upon below, a new key type is proposed - a "zone key referral" - to join the user key, host key, and zone key types defined in RFC 2535.

1.1 Performance Issues

A sample zone will be used to illustrate the problem. The example will part from reality mostly in the length of zone names, which changes the size of the owner and resource record data fields.

```
# $ORIGIN test.
# @          IN SOA   <SOA data>
#            IN SIG   SOA <by test.>
#            IN KEY   <1024 bit zone key>
#            IN SIG   KEY <by test.>
#            IN SIG   KEY <by .>
#            IN NS    ns.test.
```

```
    #           IN SIG   NS <by test.>
    #           IN NXT   my-org.test. NS SOA SIG KEY NXT
    #           IN SIG   NXT <by test.>
    #
    # my-org    IN KEY   <1024 bit zone key>
    #           IN KEY   <1024 bit zone key>
    #           IN SIG   KEY <by test.>
    #           IN NS    ns1.my-org.test.
    #           IN NS    ns2.my-org.test.
    #           IN NXT   them.test. NS SIG KEY NXT
    #           IN SIG   NXT <by test.>
    #
    # them      IN KEY   0xC100 3 1
    #           IN SIG   KEY <by test.>
    #           IN NS    ns1.them.test.
    #           IN NS    ns2.them.test.
    #           IN NXT   test. NS SIG KEY NXT
    #           IN SIG   NXT <by test.>
```

In this zone file fragment, "my-org" is a delegation point of interest with two registered public keys. Presumably, one key is for signatures generated currently and the other is for still living and valid but older signatures. "them" is another delegation point, with a NULL key. This signifies that this zone is unsecured.

To analyze the performance impact of the storing of keys, the number of bytes used to represent the RRs in the procotol format is used. The actual number of bytes stored will likely be higher, accounting for data structure overhead and alignment. The actual number of bytes transferred will be lower due to DNS name compression.

The number of bytes for my-org's two 1024-bit keys, two NS records, NXT and the associated signatures is 526. The bytes needed for them (with the NULL key) is 346. Currently, there are close to 2 million entries in com., so if we take my-org as a typical domain, over 1GB on memory will be needed for com.

The zone keys used in the example are set to 1024 bits. This number may range from as low as 512 bits upwards to over 3000 bits. To scale the above numbers to a different key size, multiply the difference in key sizes by 4 for my-org and by 2 for them, and adjust the numbers accordingly.

The increased size of the data held for the zone cuts will have two impacts at the transport and below layers. Bandwidth beyond that currently needed will be used to carry the KEY records. The inclusion of all of the child's keys will also push DNS over the UDP size limit and start using TCP - which could cause critical problems for current heavily used name servers, like the roots.

Another impact, not illustrated by the example, is the frequency of updates. If each time a public key for my-org is added or deleted, the SOA serial number will have to increase, and the SOA signed again. If an average zone changes its keys(s) once per month, there will be on average 45 updates per minute in a zone of 2 million delegations.

(The multiple algorithms issue is an extension of multiple keys. The example should be updated to show at least a DSS key as well as an RSA key.)

1.2 Security Incident Recovery (w/ respect to keys only)

Once a zone administrator is alerted that any key's private counterpart has been discovered (exposed), the first action to be taken is to stop advertising the public key in DNS. This doesn't end the availability of the key - it will be residing in caches - but is the closest action resembling revokation available in DNS.

Stopping the advertisement in the zone's name servers is as quick as altering the master file and restarting the name server. Having to do this in two places will will only delay the time until the recovery is complete.

For example, a registrar of a top level domain has decided to update its zone only on Mondays and Fridays due to the size of the zone. A customer/delegatee is the victim of a break in, in which one of the items taken is the file of private keys used to sign DNS data. If this occurs on a Tuesday, the thief has until Friday to use the keys before they disappear from the DNS, even if the child stops publishing them immediately.

If the public key set is in the parent zone, and the parent zone is not able to make the change quickly, the public key cannot be revoked quickly. If the parent only refers to there being a key at the child zone, then the child has the agility to change the keys - even issue a NULL key, which will force all signatures in the zone to become suspect.

1.3 DNS Clarifications

RFC 2181, section 6, clarifies the status of data appearing at a zone cut. Data at a zone cut is served authoritatively from the servers listed in the NS set present at the zone cut. The data is not (necessarily) served authoritatively from the parent. (The exception is in servers handling both the parent and child zone.)

Section 6 also mentions that there are two exceptions created by

DNSSEC, the NXT single record set and the KEY set. This proposal addresses the exception relating to the KEY set, limiting its severity (but falling short of removing it altogether). By limiting the exception, we will be simplifying DNS.

1.4 Liability

Liability is a legal concept, so it is not wise to attempt an engineering solution to it. However, the perceived liability incurred in using DNSSEC by registrars may prevent the adoption of DNSSEC. Hence DNSSEC should be engineered in such a away to address the concern.

One source of liability is the notion that by advertising a public key for a child zone, a parent zone is providing a service of security. With that comes responsibility. By having the parent merely indicate that a child has a key (or has no key), the parent is providing less in the way of security. If the parent is wrong, the potential loss is less. Instead of falsely authenticated data, configuration errors will be apparent to the resolving client.

2 The Proposal

The proposal is to introduce a new key type which indicates whether the delegated zone is running secured or not. Running secured is either a zone signed with at least one key, an experimental zone, or a zone with only NULL keys published.

The Zone Referral Key will resemble the NULL key in syntax. There will be a flags field, an algorithm field, and a protocol field, but no public key material. The Referral Key is signed by the parent zone, as was the public key set in RFC 2065. There is only one Referral Key RR present.

The Referral Key flags field will have the following values:

Field	Bit(s)	Value	Meaning
A/C	0- 1	0b01	indicates a key will be found
		0b11	indicates a key will not be found
		0b?0	error (referral cannot encrypt)
XT	2	0	no extended flags are needed
Z	4- 5	0	must be zero for all keys
NAMTYP	6- 7	0b11	this is a referral to a zone key
Z	8-11	0	must be zero for all keys
SIG	12-15	0	must be zero for a referral key

The legal values of the flags field are (in summary):

Hex Value	Indicates
0x4300	The delegation has a key record set

0xC300   The delegation has no key record

Other values are not valid for Referral Keys (but may be valid for other keys).

The Protocol field must be set to 3, the DNSSEC protocol value.

The Algorithm field must be 0.

The algorithm is not important at this point. So long as the searcher knows to expect a key set at the delegated zone's apex, a secure chain is possible. One the key set is retrieved and verified, then the algorithms used in the delegated zone are known. (The issue is that a zone may be signed in algorithm 1 and not 3, 3 and not 1, both, etc., and a secure resolver must know this in order to set signature arrival expectations.

2.1 Example

The Referral key for my-org.test. and them.test. would appear as the following in the zone master file:

```
      my-org.test. IN KEY   0x4300 3 0
      them.test.   IN KEY   0xC300 3 0
```

In the example introduced earlier, the master file would change to the following.

```
    # $ORIGIN test.
    # @         IN SOA   <SOA data>
    #           IN SIG   SOA <by test.>
    #           IN KEY   <1024 bit zone key>
    #           IN SIG   KEY <by test.>
    #           IN SIG   KEY <by .>
    #           IN NS    ns.test.
    #           IN SIG   NS <by test.>
    #           IN NXT   my-org.test. NS SOA SIG KEY NXT
    #           IN SIG   NXT <by test.>
    #
    # my-org    IN KEY   0x4300 3 0
    #           IN SIG   KEY <by test.>
    #           IN NS    ns1.my-org.test.
    #           IN NS    ns2.my-org.test.
    #           IN NXT   them.test. NS SIG KEY NXT
    #           IN SIG   NXT <by test.>
    #
    # them      IN KEY   0xC300 3 1
    #           IN SIG   KEY <by test.>
    #           IN NS    ns1.them.test.
    #           IN NS    ns2.them.test.
    #           IN NXT   test. NS SIG KEY NXT
    #           IN SIG   NXT <by test.>
```

3 Analysis

By removing the public keys from the parent's master file, the parent is no longer a road block during an emergency removal of keys. A parent zone is unchanged as a zone changes from NULL keys to experimental keys to fully signed. The parent is also not providing a security service, other than to authentically claim the existence of a KEY record set - akin to the "hints" of the name servers.

The change also improves the prospect for performance. The need for multiple KEY RR's, each one on the order of 100 bytes, is removed and replaced by a single KEY RR of the order of 25 bytes. Saving bytes reduces the need to use TCP to avoid truncated responses. Also, the need for updating the zone drops - no longer will there be updates for each key change.

As far as the statements by RFC 2181 conerning authority levels, the Referral Key is not authortative and would be superseeded by a verified set of the real zone keys. The only caveat is that once the verified set of keys expire (assuming the parent has to learn the keys from another server), the Referal Key must reappear. This is an example of what has been labelled "mount-like semantics."

[No reference for mount-like semantics has yet been found.]

The last point is important. This requires the "mount-like semantics" that have been discussed for the BIND name servers. Once hints are overridden by learned, authoritititative and verified data, the hints are not discarded. Hints in this state are stored and become visible when the learned data expires.

4 IANA Considerations

Other than using a new value in the flags field of the KEY RR, no new number assignments are needed. The flags field is not under the control of IANA as of yet. There are no requirements placed on IANA by this draft.

5 Security Considerations

There has been some debate about whether the Referral key should be treated as a hint - just like the NS records. If so, then there is no need to sign the Referral Key, and an unsigned (hence non-authenticated) security record is of little value. So, is the Referral Key even needed?

Authentication in DNSSEC is done from the data "back" towards a trusted point - e.g., "up" to the root. Since the authentication is done by going repeatedly from child to parent, why bother having the parent indicate the status of the child?

The answer is in the scenario in which a resolver somewhere has obtained data which fails the verification process. Perhaps the signature is wrong, a key in the chain of trust is unavailable, the set should have had a signature, but none is found (or vice versa), or the trail of signed-by names is not acceptable. In this case, the resolver needs to find the authoritative zone, its status and its name server set.

If a zone is being attacked by a masquerader, and parents do not make any statements about the security of child zones, then an easy and successfull attack may occur. An attacker only needs to supply either fake name server records or glue records to redirect queries.

While this attack will not be stopped as far as denial of service, the masquerader can be stopped from being accepted as an authoritative source if the parent of the zone claims the child is secure and signs the public keys of the true child and not the masquerader.

The masquerader cannot successfully claim that the zone is unsigned, because it must have a zone key signed by the parent. NULL or not, the key would not be trusted by the resolver, assuming the parent has not also been duped. The resolver, sensing this, should report an error or security incident, and not accept data.

6 Acknowledgements

John Gilmore originally raised the issues that have led to this document.

7 Author's addresses

Edward Lewis
<lewis@tislabs.com>
3060 Washinton Rd (Rte 97)
Glenwood, MD 21738
+1(443)259-2352

Jerry Scharf
<scharf@vix.com>

8 References

RFC 2181 "Clarifications to the DNS Specification", Elz and Bush
RFC 2535 "Domain Name System Security Extensions", Eastlake

9 Full Copyright Statement

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."

This draft expires on October 1, 1999