<draft-ietf-dnsind-simple-secure-update-02.txt>

Updates: RFC 2535, RFC 2136, [TSIG]
Replaces: RFC 2137, [update2]

Simple Secure Domain Name System (DNS) Dynamic Update

Status of this Memo

Abstract

This document proposes a method for performing secure Domain Name System (DNS) dynamic updates. The method described here is intended

to be flexible and useful while requiring as few changes to the protocol as possible. The authentication of the dynamic update message is separate from later DNSSEC validation of the data. Secure communication based on authenticated requests and transactions is used to provide authorization.

1 - Introduction

This document defines a means to secure dynamic updates of the Domain Name System (DNS), allowing only authorized sources to make changes to a zone's contents. The existing unsecured dynamic update operations form the basis for this work.

Familiarity with the DNS system [RFC1034, RFC1035] and dynamic update [RFC2136] is helpful and is assumed by this document. In addition, knowledge of DNS security extensions [RFC2535], SIG(0) transaction security [RFC2535], and TSIG transaction security [TSIG] is recommended.

This document updates portions of RFC 2535, in particular sections 2.3.6 and 3.1.2. This document obsoletes RFC 2137, an alternate proposal for secure dynamic update, due to implementation experience.

1.1 - Overview of DNS Dynamic Update

DNS dynamic update defines a new DNS opcode and a new interpretation of the DNS message if that opcode is used. An update can specify insertions or deletions of data, along with prerequisites necessary for the updates to occur. All tests and changes for a DNS update request are restricted to a single zone, and are performed at the primary server for the zone. The primary server for a dynamic zone must increment the zone SOA serial number when an update occurs or before the next retrieval of the SOA.

1.2 - Overview of DNS Transaction Security

Transactions including TSIG [TSIG] or SIG(0) [RFC2535] records allow two DNS entities to authenticate DNS requests and responses sent between them. A TSIG MAC (message authentication code) is derived from a shared secret, and a SIG(0) is generated from a private key whose public counterpart is stored in DNS. In both cases, a record containing the message signature/MAC is included as the final resource record in a DNS message. Keyed hashes, used in TSIG, are inexpensive to calculate and verify. Public key encryption, as used in SIG(0), is more scalable as the public keys are stored in DNS.

1.3 - Comparison of data authentication and message authentication

In some cases, DNSSEC SIG records could be used to protect the integrity of individual RRs or RRsets in the update message. There are several problems with this, though. First, SIG records do not cover the message header (which includes record counts). Therefore, malicious tampering in the header or the removal of records might not be detected. A SIG record could be required in the zone section, but adds no protection since this SIG is present in DNS. SIG records could be created to protect data in the prerequisite section, but this would imply that the SIG is a prerequisite, and in some cases, the SIG already is present in DNS. In the update section, signing addition requests is straightforward, as the SIG would cover the full set. If records are deleted, though, the final set may be empty and thus impossible to sign.

Message based authentication, using TSIG or SIG(0), avoids these problems, since only one signature/MAC is computed for the message, and this signature/MAC protects the integrity of the entire message. This is also a less expensive operation, since it is only performed once per update.

1.4 - Disallowing non-zone key SIG records in DNS validation

Updating RFC 2535, the DNSSEC validation process performed by a resolver MUST ignore all keys that are not zone keys unless local policy dictates otherwise. This requires that when performing secure dynamic update, all zone data modified in a signed zone MUST be signed by a relevant zone key. There are several reasons for this change.

First, the primary reason to allow host and/or user keys to generate material DNSSEC signatures is to allow dynamic update without online zone keys. Online zone keys are necessary, though, to sign NXT and SOA sets. These online zone keys can sign any incoming data, thus removing the need for host/user key signatures. This also simplifies the validation process. If data must be signed by a zone key, determining whether a key is authorized to sign an RRset requires finding the enclosing zone of the RRset, and following a chain of trusted zone keys to a known trusted key (which may be the DNS root key). If host and user keys were permitted to generate material signatures, following a chain of trust to a trusted DNSSEC key could involve any number of non-zone keys and a non-trivial amount of work to determine if all such keys have the proper authority. Finally, there is no additional flexibility granted by allowing host/user key generated material signatures.

The primary usefulness of host and user keys, with respect to DNSSEC, is to authenticate messages, including dynamic updates. Thus, host and user keys MAY be used to generate SIG(0) records to authenticate updates and MAY be used in the TKEY [TKEY] process to generate TSIG shared

secrets. In both cases, no SIG records (except SIG(0) records) generated by non-zone keys will be used in a DNSSEC validation process unless local policy dictates.

This completely disassociates authentication of an update request from authentication of the data itself. Authentication of the update message can be done with either TSIG shared secrets or DNSSEC host or user keys. Authentication of the data, once it is present in DNS, only involves DNSSEC zone keys and signatures generated by them.

1.5 - Signatory strength

[RFC2535] defines the signatory field of a key as the final 4 bits of the flags field, but does not define its value. This proposal leaves this field undefined. Updating RFC 2535, this field SHOULD be set to 0 in KEY records, and MUST be ignored.

2 - Authentication

TSIG or SIG(0) records MUST be attached to all secure dynamic update messages. This allows the server to verifiably determine the originator of a message. If the message contains authentication in the form of a SIG(0), the identity of the sender (that is, the principal) is the owner of the KEY that generated the SIG(0). If the message contains a TSIG generated by a statically configured shared secret, the principal is the same as the shared secret name. If the message contains a TSIG generated by a dynamically configured shared secret, the principal is the same as the one that authenticated the TKEY process; if the TKEY process was unauthenticated, no information is known about the principal.

SIG(0) signatures MUST NOT be generated by zone keys, since the transaction is initiated by a host or user, not a zone.

DNSSEC SIG records (other than SIG(0)) MAY be included in an update message, but MUST NOT be used to authenticate the update request.

If an update fails because it is signed with an unauthorized key, the server MUST indicate failure by returning a message with the REFUSED rcode. Other TSIG, SIG(0), or dynamic update errors are returned unchanged.

3 - Policy

All policy is configured by the zone administrator and enforced by the zone's primary name server. Policy checks are based on principal, where the principal is derived from the message signing key and applied to dynamic update messages signed with that key.

The server's policy defines criteria which determine if the key used to sign the update is permitted to perform the requested updates. By default, a principal MUST NOT be permitted to make any changes to zone data; any permissions MUST be explicitly enabled.

The policy is fully implemented in the server for several reasons. It removes limitations imposed by encoding policy into a fixed number of bits (such as the KEY's signatory field). Policy is only relevant in the server applying it, so there is no reason to expose it. Finally, a change in policy or a new type of policy should not affect the DNS protocol or data format, and should not cause interoperability failures.

3.1 - Standard policies

Implementations SHOULD allow access control policies to use the principal as an authorization token, and MAY also allow policies to grant permission to a signed message regardless of principal.

A common practice would be to restrict the permissions of a principal by domain name. That is, a principal could be permitted to add, delete, or modify entries corresponding to one or more domain names. Implementations SHOULD allow per-name access control, and SHOULD provide a concise representation of the principal's own name, its subdomains, and all names in the zone.

Additionally, a server SHOULD restrict updates by RR type, so that a principal could add, delete, or modify specific record types at certain names. Implementations SHOULD allow per-type access control, and SHOULD provide concise representations of all types and all ``user'' types, where a user type is defined as one that does not affect the operation of DNS itself.

3.1.1 - User types

User types include all data types except SOA, NS, SIG, and NXT. SOA and NS SHOULD NOT be modified by normal users, since they can create or modify delegation points. The addition of SIG records can lead to attacks resulting in additional workload for resolvers, and the deletion of SIG records could lead to extra work for the server if the zone SIG was deleted. Note that these records are not forbidden, but not

recommended for normal users.

NXT records are explicitly forbidden, as their update may cause instability in the protocol.

3.2 - Additional policies

A framework for specifying additional policies is beyond the scope of this document. Policies may be as specific or general as desired, and as complex as desired. They may depend on the principal or any other characteristics of the signed message.

4 - Interaction with DNSSEC

An authorized update request MAY include SIG records with each RRset. Since SIG records (except SIG(0) records) MUST NOT be used for authentication of the update message, they are not required. If the updated zone is signed, the server will generate SIG records for each updated RRset with one or more zone keys (of which the private components MUST be online), unless the update message includes a valid signature by a zone key for the RRset. If multiple zone keys are online and an RRset requires a signature, a SIG MUST be generated by at least one of the zone keys.

If a principal is authorized to add SIG records and there are SIG records in the request, the following rules are applied. If a SIG record was not generated by a zone key for the relevant zone, the SIG is retained. Otherwise, the SIG is verified (the public key must be available if the determination that it is a zone key was made), and is retained if verification succeeds and dropped if verification fails. At the completion of the update process, each updated RRset must be signed in accordance with the zone's signing policy; the SIGs must either be included in the update or generated by the server.

The server MUST also, if necessary, generate a new SOA record and new NXT records, and sign these with the appropriate zone keys. Unlike traditional dynamic update, the client is forbidden from updating NXT records. SOA updates are allowed, since SOA serial advancement policies are outside of the scope of the DNS protocol.

5 - Security considerations

This document requires that a zone key and possibly other cryptographic secret material be held in an on-line, network connected host, most likely a name server. This material is at the mercy of host security to remain a secret. Exposing this secret puts DNS data at risk of masquerade attacks. The data at risk is that in both zones served by the machine and delegated from this machine.

Allowing updates of KEY records may lead to undesirable results, since a principal may be allowed to insert a public key without holding the private key, and possibly masquerade as the key owner. This is more of an issue with dynamic update in general, but additional policies requiring self-signed KEYs may alleviate this problem.

6 - Acknowledgements

The author would like to thank the following people for review and informative comments (in alphabetical order):

Donald Eastlake
Olafur Gudmundsson
Andreas Gustafsson
Bob Halley
Stuart Kwan
Ed Lewis

7 - References

[RFC1034] P. Mockapetris, ``Domain Names - Concepts and Facilities,'' RFC 1034, ISI, November 1987.

[RFC1035] P. Mockapetris, ``Domain Names - Implementation and Specification,'' RFC 1035, ISI, November 1987.

[RFC2136] P. Vixie (Ed.), S. Thomson, Y. Rekhter, J. Bound ``Dynamic Updates in the Domain Name System,'' RFC 2136, ISC & Bellcore & Cisco & DEC, April 1997.

[RFC2137] D. Eastlake ``Secure Domain Name System Dynamic Update,'' RFC 2137, CyberCash, April 1997.

[RFC2535] D. Eastlake, ``Domain Name System Security Extensions,'' RFC 2065, IBM, March 1999.

[TSIG] P. Vixie (ed), O. Gudmundsson, D. Eastlake, B. Wellington ``Secret Key Transaction Signatures for DNS (TSIG),'' draft-

ietf-dnsind-tsig-12.txt, ISC & NAILabs & IBM & NAILabs, December 1999.

[TKEY] D. Eastlake ``Secret Key Establishment for DNS (TKEY RR),'' draft-ietf-dnsind-tkey-01.txt, IBM, May 1999.

8 - Author's Address

Brian Wellington
NAILabs
Network Associates
3060 Washington Road (Rt. 97)
Glenwood, MD 21738
+1 443 259 2369
<bwelling@tislabs.com>