DNSEXT Working Group
INTERNET-DRAFT

Brian Wellington (Nominum)

October 2000

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

Updates: RFC 2535, RFC 2136
Replaces: RFC 2137

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.

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 RFC 2119 [RFC2119].

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, RFC2931], and TSIG transaction security [RFC2845] is recommended.

This document updates portions of RFC 2535, in particular section 3.1.2, and RFC 2136. 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

Exchanges of DNS messages which include TSIG [RFC2845] or SIG(0) [RFC2535, RFC2931] 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

Message based authentication, using TSIG or SIG(0), provides protection for the entire message with a single signing and single verification which, in the case of TSIG, is a relatively inexpensive MAC creation and check. For update requests, this signature can establish, based on policy or key negotation, the authority to make the request.

DNSSEC SIG records can be used to protect the integrity of individual RRs or RRsets in a DNS message with the authority of the zone owner. However, this cannot sufficiently protect the dynamic update request.

Using SIG records to secure RRsets in an update request is incompatible with the design of update, as described below, and would in any case require multiple expensive public key signatures and verifcations.

SIG records do not cover the message header, which includes record counts. Therefore, it is possibly to maliciously insert or remove RRsets in an update request without causing a verification failure.

If SIG records were used to protect the prerequisite section, it would be impossible to determine whether the SIGs themselves were a prerequisite or simply used for validation.

In the update section of an update request, signing requests to add an RRset is straightforward, and this signature could be permanently used to protect the data, as specified in [RFC2535]. However, if an RRset is deleted, there is no data for a SIG to cover.

1.4 - Data and message signatures

As specified in [signing-auth], the DNSSEC validation process performed

by a resolver MUST NOT process any non-zone keys unless local policy dictates otherwise. When performing secure dynamic update, all zone data modified in a signed zone MUST be signed by a relevant zone key. This completely disassociates authentication of an update request from authentication of the data itself.

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 [RFC2930] process to generate TSIG shared secrets. In both cases, no SIG records generated by non-zone keys will be used in a DNSSEC validation process unless local policy dictates.

Authentication of data, once it is present in DNS, only involves DNSSEC zone keys and signatures generated by them.

1.5 - Signatory strength

[RFC2535, section 3.1.2] 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 [RFC2535], this field SHOULD be set to 0 in KEY records, and MUST be ignored.

2 - Authentication

TSIG or SIG(0) records MUST be included in 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 RR that generated the SIG(0). If the message contains a TSIG generated by a statically configured shared secret, the principal is the same as or derived from 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, and the associated TSIG shared secret MUST NOT be used for secure dynamic update.

SIG(0) signatures SHOULD NOT be generated by zone keys, since transactions are 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 RCODE REFUSED. Other TSIG, SIG(0), or dynamic update errors are returned as specified in the appropriate protocol description.

3 - Policy

All policy is configured by the zone administrator and enforced by the zone's primary name server. Policy dictates the authorized actions that an authenticated principal can take. Policy checks are based on the principal and the desired action, 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 enabled though configuration.

The policy is fully implemented in the primary zone server's configuration for several reasons. This removes limitations imposed by encoding policy into a fixed number of bits (such as the KEY RR'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 these types 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 MUST NOT be created, modified, or deleted by dynamic update, as their update may cause instability in the protocol. This is an update to RFC 2136.

Issues concerning updates of KEY records are discussed in the Security Considerations section.

3.2 - Additional policies

Users are free to implement any policies. 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

Although this protocol does not change the way updates to secure zones are processed, there are a number of issues that should be clarified.

4.1 - Adding SIGs

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 a principal is authorized to update SIG records and there are SIG records in the update, the SIG records are added without verification. The server MAY examine SIG records and drop SIGs with a temporal validity period in the past.

4.2 - Deleting SIGs

If a principal is authorized to update SIG records and the update specifies the deletion of SIG records, the server MAY choose to override the authority and refuse the update. For example, the server may allow

all SIG records not generated by a zone key to be deleted.

4.3 - Non-explicit updates to SIGs

If the updated zone is secured, the RRset affected by an update operation MUST, at the completion of the update, be signed in accordance with the zone's signing policy. This will usually require one or more SIG records to be generated by one or more zone keys whose private

components MUST be online [signing-auth].

When the contents of an RRset are updated, the server MAY delete all associated SIG records, since they will no longer be valid.

4.4 - Effects on the zone

If any changes are made, the server MUST, if necessary, generate a new SOA record and new NXT records, and sign these with the appropriate zone keys. Changes to NXT records by secure dynamic update are explicitly forbidden. SOA updates are allowed, since the maintenance of SOA parameters is 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.

6 - Acknowledgements

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

Harald Alvestrand
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 2535, IBM, March 1999.

[RFC2845] P. Vixie, O. Gudmundsson, D. Eastlake, B. Wellington ``Secret Key Transaction Signatures for DNS (TSIG),'' RFC 2845, ISC & NAILabs & Motorola & Nominum, May 2000.

[RFC2930] D. Eastlake ``Secret Key Establishment for DNS (TKEY RR),'' RFC 2930, Motorola, September 2000.

[RFC2931] D. Eastlake ``DNS Request and Transaction Signatures (SIG(0)s),'' RFC 2931, Motorola, September 2000.

[signing-auth] B. Wellington ``Domain Name System Security (DNSSEC) Signing Authority,'' draft-ietf-dnsext-signing-auth-02.txt, Nominum, October 2000.

8 - Author's Address

Brian Wellington
Nominum, Inc.
950 Charter Street
Redwood City, CA 94063
+1 650 779 6022
<Brian.Wellington@nominum.com>

9 - Full Copyright Statement