Network Working Group S. Rose
Internet-Draft NIST
Expires: February 4, 2004 August 6, 2003

DNS Request and Transaction Signatures (SIG(0)s)
draft-srose-rfc2931bis-00

Status of this Memo

Copyright Notice

Abstract

Extensions to the Domain Name System (DNS) can provide data origin and transaction integrity and authentication to security aware resolvers and applications through the use of cryptographic digital signatures. However, these security extensions do not provide authentication at the transaction or message level. This document describes a message authentication scheme (called SIG(0)) that provides message level authentication and integrity checking by means of a meta-RR in the additional section of a DNS message.

Table of Contents

1. Introduction

intro

It is assumed that the reader has some knowledge of the DNSSEC extensions ([6], [7], and [8]) The key words "MUST", "MUST NOT", NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [2].

2. SIG(0) Design Rationale

SIG(0) provides protection for DNS transactions and requests that is not provided by the regular RRSIG, DNSKEY, and NSEC RRs specified in [7]. These services do not cover glue records, DNS message headers, the query section of DNS requests, and do not provide protection of the overall integrity of a DNS message. The RRSIG RR is used to authenticate data resource records (RRs) or authenticatably deny their nonexistence. The SIG(0) RR is a variant of the RRSIG RR that covers the entire DNS message. This would give the same protection levels to the DNS message headers and query section as the RRSIG RR gives to a data RR set.

2.1 Message Authentication

Message authentication means that a requester can be sure it is at least getting the messages from the server it queried and that the received messages have not be tampered with in transit. This is accomplished by optionally adding either a TSIG RR [3] or, a SIG(0) RR at the end of the message which digitally signs the concatenation of the server's response and the corresponding resolver query.

2.2 Request Authentication

Queries and update messages can be authenticated by including a TSIG or a SIG(0) RR at the end of the request. There is little need to authenticate a traditional DNS query, although it may be desired for dynamic updates to a zone, or to provide proof of the identity. In the latter, message authentication may be used as a form of indentification. The presence of a SIG(0) may allow certain access based on the capability of providing a SIG(0) signature. Due to the cost associated with generating a SIG(0) RR, this ability should not be used for general purpose DNS lookups.

Requests with a non- empty additional information section produce error returns or may even be ignored by a few such older DNS servers. However, this syntax for signing requests is defined to be used for authenticating dynamic update requests [5], TKEY requests [4], or possible future requests requiring authentication.

2.3 Keying

The private keys used in transaction authentication belong to the entitiy composing the DNS message, not to the zone involved. Request authentication may also involve the private key of the host or other entity depending on the request authority seeking to be established. The corresponding public key(s) are normally stored in and retrieved from the DNS for verification as KEY RRs with a protocol byte of 3

(DNSSEC).

3. Differences Between TSIG and SIG(0)

There are significant differences between TSIG and SIG(0).

Because TSIG involves secret keys installed at both the requester and server the presence of such a key implies that the other party understands TSIG and very likely has the same key installed. Furthermore, TSIG uses keyed hash authentication codes which are relatively inexpensive to compute. Thus it is common to authenticate requests with TSIG and responses are authenticated with TSIG if the corresponding request is authenticated.

SIG(0) on the other hand, uses public key authentication, where the public keys are stored in DNS as KEY RRs and a private key is stored at the signer. Existence of such a KEY RR does not necessarily imply implementation of SIG(0). In addition, SIG(0) involves relatively expensive public key cryptographic operations that should be minimized and the verification of a SIG(0) involves obtaining and verifying the corresponding KEY which can be an expensive and lengthy operation. Indeed, a policy of using SIG(0) on all requests and verifying it before responding would, for some configurations, lead to a deadly embrace with the attempt to obtain and verify the KEY needed to authenticate the request SIG(0) resulting in additional requests accompanied by a SIG(0) leading to further requests accompanied by a SIG(0), etc. Furthermore, omitting SIG(0)s when not required on requests halves the number of public key operations required by the transaction.

For these reasons, SIG(0)s SHOULD only be used on requests when necessary to authenticate that the requester has some required privilege or identity. SIG(0)s on replies are defined in such a way as to not require a SIG(0) on the corresponding request and still provide transaction protection. For other replies, whether they are authenticated by the server or required to be authenticated by the requester SHOULD be a local configuration option.

4. The SIG(0) Resource Record

Note: requests and responses can either have a single TSIG or one SIG(0) but not both a TSIG and a SIG(0).

The structure of the SIG(0) resource records (RRs) is similar to the RRSIG RR [7] with the following differences outlined below. Any conflict between the DNSSEC specification and this document concerning SIG(0) RRs should be resolved in favor of this document.

The owner's name of a SIG(0) MUST be the root. That is, a single zero (0) octet. Likewise, the class code MUST be ANY and TTL value MUST be zero (0).

The type code for the SIG(0) is 24.

The RDATA of the SIG(0) is given as:

```
                     1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          reserved             |  Algorithm    |     Labels    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            reserved                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Signature Expiration                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Signature Inception                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Key Tag            |                               /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+         Signer's Name         /
/                                                               /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               /
/                            Signature                          /
/                                                               /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The fixed sized resevered sections MUST be zero (0).

The Algorithm field is described in Section 6.

The Labels and Key Tag field are constructed the same way as the same filds in the RRSIG RR. See [7]. Since the owner name of the SIG(0) is zero, the labels field MUST also be zero (0).

For all SIG(0)s, the signer name field MUST be a name of the originating host and there MUST be a KEY RR at that name with the

public key corresponding to the private key used to calculate the signature. (The host domain name used may be the inverse IP address mapping name for an IP address of the host if the relevant KEY is stored there.)

4.1 SIG(0) Lifetime and Expiration

The inception and expiration times in SIG(0)s are for the purpose of resisting replay attacks. They should be set to form a time bracket such that messages outside that bracket can be ignored. In IP networks, this time bracket should not normally extend further than 5 minutes into the past and 5 minutes into the future.

4.2 Calculating Request and Transaction SIG(0)s

A DNS query message signed with a SIG(0) places the RR at the end of the additional section. The signature is calculated by using a plaintext (see [7]) of (1) the SIG(0)'s RDATA entirely omitting the signature section itself (19 bytes), (2) the entire DNS message minus the UDP/IP header. The additional section RR count in the DNS message header should NOT include the SIG(0) itself.

That is:

```
plaintext = RDATA | DNSquery - SIG(0)
```

where "|" is concatenation and RDATA is the RDATA of the SIG(0) being calculated omitting the signature field itself.

Similarly, a SIG(0) used to secure a response are calculated by using a plaintext of (1) the SIG(0) RDATA omitting the signature itself (again, 19 bytes), (2) the entire DNS query message that produced this response, but not its UDP/IP header, and (3) the entire DNS response message, but not the UDP/IP header. Again, like the query message, the additional section RR counts do not reflect the the SIG(0) RR itself.

That is

```
plaintext = RDATA | full query | response - SIG(0)
```

where "|" is concatenation and RDATA is the RDATA of the SIG(0) being calculated.

Verification of a response SIG(0) (which is signed by the server host key, not the zone key) by the requesting resolver shows that the query and response were not tampered with in transit, that the

response corresponds to the intended query, and that the original response comes from the queried server.

In the case of a DNS message via TCP, a SIG(0) on the first data packet is calculated with "data" as above and for each subsequent packet, it is calculated as follows:

```
data = RDATA | DNS payload - SIG(0) | previous packet
```

where "|" is concatenations, RDATA is as above, and previous packet is the previous DNS payload including DNS header and the SIG(0) but not the TCP/IP header. Support of SIG(0) for TCP is OPTIONAL. As an alternative, TSIG may be used after, if necessary, setting up a key with TKEY [4].

Except where needed to authenticate an update, TKEY, or similar privileged request, servers are not required to check for a request SIG(0).

4.3 Inclusion of SIG(0) RR in a DNS Message

When SIG(0) authentication on a response is desired, that SIG RR MUST be considered the highest priority of any additional information for inclusion in the response. If the SIG(0) RR cannot be added without causing the message to be truncated, the server MUST alter the response so that a SIG(0) can be included. This response consists of only the question and a SIG(0) record, and has the TC bit set and RCODE 0 (NOERROR). The client should retry the request using TCP.

4.4 Processing Responses and SIG(0) RRs

A SIG(0) SHOULD be placed as the last RR in the additional section of a DNS message. If it is located in any other section, it MUST NOT be considered valid. For TKEY responses, it MUST be checked and the message rejected if the checks fail unless otherwise specified for the TKEY mode in use. For all other responses, it MAY be checked and the message rejected if the checks fail.

If a response's SIG(0) check succeed, such a transaction authentication signature does NOT directly authenticate the validity any data-RRs in the message. However, it authenticates that they were sent by the queried server and have not been altered. (Only a proper SIG(0) RR signed by the zone or a key tracing its authority to the zone or to static resolver configuration can directly authenticate data-RRs, depending on resolver policy.) If a resolver or server does not plan to implement transaction and/or request SIG(0), it MUST ignore them without error where they are optional and treat them as failing where they are required.

5. Security Considerations

A more detailed description of the threats against the DNS are given in [9].

Because requests and replies are highly variable, message authentication SIGs can not be pre-calculated. Thus it will be necessary to keep the private key on-line. This will cause the DNS entity to rely on the system security for keeping the key secure.

The inclusion of the SIG(0) inception and expiration time under the signature improves resistance to replay attacks. The benefit of using private and public key pairs allows for the distribution of the public verification key while keeping the private signing key secure. This is an advantage of SIG(0) message authentication schemes over the TSIG RR schemes, which use a shared secret that must be distributed securely.

SIG(0) signature scheme cannot be used to authenticate source data, only to authenticate a resolver request and/or a server response. The DNS security mechanisms described in [7] should be used to provide coverage of the original source data.

6. IANA Considerations

In order to allow DNS Security and SIG(0) to use different sets of algorithms, the existing "DNS Security Algorithm Numbers" registry is renamed as the "SIG(0) Algorithm Numbers" registry and a new "DNS Security Algorithm Numbers" registry is established. The initial algorithm values are:

VALUE	Algorithm [mnemonic]	RFC
0	Reserved	-
1	Reserved (Obsolete)	RFC 2537
2	Diffie-Hellman [DH]	RFC 2539
3	DSA [DSA]	RFC 2536
4	available for assignment	
5	RSA/SHA1 [RSA/SHA1]	RFC 3110
6-252	available for assignment	-
253	private [PRIVATE_DNS]	-
254	private [PRIVATE_OID]	-
255	reserved	-

As support for SIG(0) is not mandatory to the DNS protocol, there are no mandatory to implement algorithms for SIG(0). It is suggested, but not required, that new algorithms usable by both DNS Security and SIG(0) be assigned the same number in both registries.

7. Acknowledgements

The author would like to acknowledge the author of the original SIG(0) draft, Donald Eastlake, as well as the express graditude for those that helped prod the author into producing this draft.

Nornative References

[1] Elz, R. and R. Bush, "Serial Number Arithmatic", RFC 1982, August 1996.

[2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

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

[4] Eastlake, D., "Secret Key Establishment for DNS (TKEY RR)", RFC 2930, September 2000.

[5] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", RFC 3007, November 2000.

Informative References

[6] Arends, R., Austein, R., Larson, M., Massey, D. and S. Rose, "DNS Security Introduction and Requirements", draft-ietf-dnsext-dnssec-intro-05 (work in progress), February 2003.

[7] Arends, R., Austein, R., Larson, M., Massey, D. and S. Rose, "Resource Records for DNS Security Extensions", draft-ietf-dnsext-dnssec-records-04 (work in progress), February 2003.

[8] Arends, R., Austein, R., Larson, M., Massey, D. and S. Rose, "Protocol Modifications for the DNS Security Extensions", draft-ietf-dnsext-dnssec-protocol-00 (work in progress), Februari 2003.

[9] Atkins, D. and R. Austein, "Threat Analysis Of The Domain Name System", draft-ietf-dnsext-dns-threats-02 (work in progress), November 2002.

Author's Address

Scott Rose
National Institute for Standards and Technology
100 Bureau Drive
Gaithersburg, MD 20899-8920
USA

EMail: scott.rose@nist.gov

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.