Network Working Group
Request for Comments: 3445
Updates: 2535
Category: Standards Track

D. Massey
USC/ISI
S. Rose
NIST

December 2002

Limiting the Scope of the KEY Resource Record (RR)

Status of this Memo

Copyright Notice

Abstract

This document limits the Domain Name System (DNS) KEY Resource Record (RR) to only keys used by the Domain Name System Security Extensions (DNSSEC). The original KEY RR used sub-typing to store both DNSSEC keys and arbitrary application keys. Storing both DNSSEC and application keys with the same record type is a mistake. This document removes application keys from the KEY record by redefining the Protocol Octet field in the KEY RR Data. As a result of removing application keys, all but one of the flags in the KEY record become unnecessary and are redefined. Three existing application key sub-types are changed to reserved, but the format of the KEY record is not changed. This document updates RFC 2535.

1. Introduction

This document limits the scope of the KEY Resource Record (RR). The KEY RR was defined in [3] and used resource record sub-typing to hold arbitrary public keys such as Email, IPSEC, DNSSEC, and TLS keys. This document eliminates the existing Email, IPSEC, and TLS sub-types and prohibits the introduction of new sub-types. DNSSEC will be the only allowable sub-type for the KEY RR (hence sub-typing is essentially eliminated) and all but one of the KEY RR flags are also eliminated.

Section 2 presents the motivation for restricting the KEY record and Section 3 defines the revised KEY RR. Sections 4 and 5 summarize the changes from RFC 2535 and discuss backwards compatibility. It is important to note that this document restricts the use of the KEY RR and simplifies the flags, but does not change the definition or use of DNSSEC keys.

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 [1].

2. Motivation for Restricting the KEY RR

The KEY RR RDATA [3] consists of Flags, a Protocol Octet, an Algorithm type, and a Public Key. The Protocol Octet identifies the KEY RR sub-type. DNSSEC public keys are stored in the KEY RR using a Protocol Octet value of 3. Email, IPSEC, and TLS keys were also stored in the KEY RR and used Protocol Octet values of 1,2, and 4 (respectively). Protocol Octet values 5-254 were available for assignment by IANA and values were requested (but not assigned) for applications such as SSH.

Any use of sub-typing has inherent limitations. A resolver can not specify the desired sub-type in a DNS query and most DNS operations apply only to resource records sets. For example, a resolver can not directly request the DNSSEC subtype KEY RRs. Instead, the resolver has to request all KEY RRs associated with a DNS name and then search the set for the desired DNSSEC sub-type. DNSSEC signatures also apply to the set of all KEY RRs associated with the DNS name, regardless of sub-type.

In the case of the KEY RR, the inherent sub-type limitations are exacerbated since the sub-type is used to distinguish between DNSSEC keys and application keys. DNSSEC keys and application keys differ in virtually every respect and Section 2.1 discusses these differences in more detail. Combining these very different types of keys into a single sub-typed resource record adds unnecessary complexity and increases the potential for implementation and deployment errors. Limited experimental deployment has shown that application keys stored in KEY RRs are problematic.

This document addresses these issues by removing all application keys from the KEY RR. Note that the scope of this document is strictly limited to the KEY RR and this document does not endorse or restrict the storage of application keys in other, yet undefined, resource records.

2.1 Differences Between DNSSEC and Application Keys

DNSSEC keys are an essential part of the DNSSEC protocol and are used by both name servers and resolvers in order to perform DNS tasks. A DNS zone key, used to sign and authenticate RR sets, is the most common example of a DNSSEC key. SIG(0) [4] and TKEY [3] also use DNSSEC keys.

Application keys such as Email keys, IPSEC keys, and TLS keys are simply another type of data. These keys have no special meaning to a name server or resolver.

The following table summarizes some of the differences between DNSSEC keys and application keys:

1. They serve different purposes.

2. They are managed by different administrators.

3. They are authenticated according to different rules.

4. Nameservers use different rules when including them in responses.

5. Resolvers process them in different ways.

6. Faults/key compromises have different consequences.

1. The purpose of a DNSSEC key is to sign resource records associated with a DNS zone (or generate DNS transaction signatures in the case of SIG(0)/TKEY). But the purpose of an application key is specific to the application. Application keys, such as PGP/email, IPSEC, TLS, and SSH keys, are not a mandatory part of any zone and the purpose and proper use of application keys is outside the scope of DNS.

2. DNSSEC keys are managed by DNS administrators, but application keys are managed by application administrators. The DNS zone administrator determines the key lifetime, handles any suspected key compromises, and manages any DNSSEC key changes. Likewise, the application administrator is responsible for the same functions for the application keys related to the application. For example, a user typically manages her own PGP key and a server manages its own TLS key. Application key management tasks are outside the scope of DNS administration.

3. DNSSEC zone keys are used to authenticate application keys, but by definition, application keys are not allowed to authenticate DNS zone keys. A DNS zone key is either configured as a trusted key or authenticated by constructing a chain of trust in the DNS hierarchy. To participate in the chain of trust, a DNS zone needs to exchange zone key information with its parent zone [3]. Application keys are not configured as trusted keys in the DNS and are never part of any DNS chain of trust. Application key data is not needed by the parent and does not need to be exchanged with the parent zone for secure DNS resolution to work. A resolver considers an application key RRset as authenticated DNS information if it has a valid signature from the local DNS zone keys, but applications could impose additional security requirements before the application key is accepted as authentic for use with the application.

4. It may be useful for nameservers to include DNS zone keys in the additional section of a response, but application keys are typically not useful unless they have been specifically requested. For example, it could be useful to include the example.com zone key along with a response that contains the www.example.com A record and SIG record. A secure resolver will need the example.com zone key in order to check the SIG and authenticate the www.example.com A record. It is typically not useful to include the IPSEC, email, and TLS keys along with the A record. Note that by placing application keys in the KEY record, a resolver would need the IPSEC, email, TLS, and other key associated with example.com if the resolver intends to authenticate the example.com zone key (since signatures only apply to the entire KEY RR set). Depending on the number of protocols involved, the KEY RR set could grow unwieldy for resolvers, and DNS administrators to manage.

5. DNS zone keys require special handling by resolvers, but application keys are treated the same as any other type of DNS data. The DNSSEC keys are of no value to end applications, unless the applications plan to do their own DNS authentication. By definition, secure resolvers are not allowed to use application keys as part of the authentication process. Application keys have no unique meaning to resolvers and are only useful to the application requesting the key. Note that if sub-types are used to identify the application key, then either the interface to the resolver needs to specify the sub-type or the application needs to be able to accept all KEY RRs and pick out the desired sub-type.

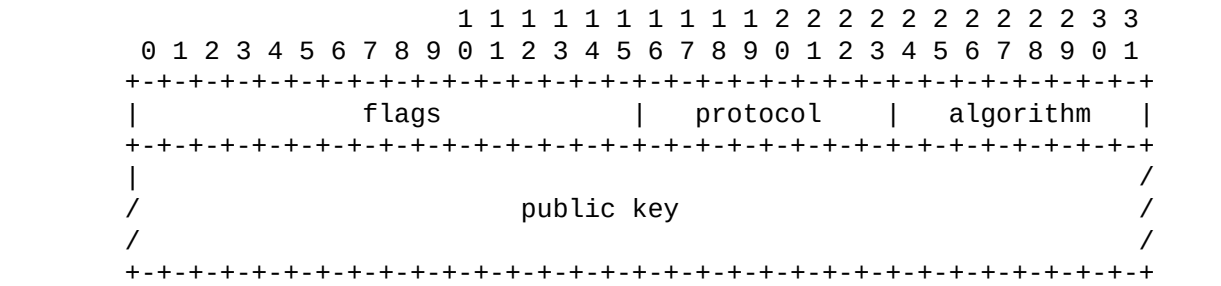
6. A fault or compromise of a DNS zone key can lead to invalid or forged DNS data, but a fault or compromise of an application key should have no impact on other DNS data. Incorrectly adding or changing a DNS zone key can invalidate all of the DNS data in the zone and in all of its subzones. By using a compromised key, an

attacker can forge data from the effected zone and for any of its sub-zones. A fault or compromise of an application key has implications for that application, but it should not have an impact on the DNS. Note that application key faults and key compromises can have an impact on the entire DNS if the application key and DNS zone keys are both stored in the KEY RR.

In summary, DNSSEC keys and application keys differ in most every respect. DNSSEC keys are an essential part of the DNS infrastructure and require special handling by DNS administrators and DNS resolvers. Application keys are simply another type of data and have no special meaning to DNS administrators or resolvers. These two different types of data do not belong in the same resource record.

3. Definition of the KEY RR

The KEY RR uses type 25 and is used as resource record for storing DNSSEC keys. The RDATA for a KEY RR consists of flags, a protocol octet, the algorithm number octet, and the public key itself. The format is as follows:



KEY RR Format

In the flags field, all bits except bit 7 are reserved and MUST be zero. If Bit 7 (Zone bit) is set to 1, then the KEY is a DNS Zone key. If Bit 7 is set to 0, the KEY is not a zone key. SIG(0)/TKEY are examples of DNSSEC keys that are not zone keys.

The protocol field MUST be set to 3.

The algorithm and public key fields are not changed.

4. Changes from RFC 2535 KEY RR

The KEY RDATA format is not changed.

All flags except for the zone key flag are eliminated:

The A/C bits (bits 0 and 1) are eliminated. They MUST be set to 0 and MUST be ignored by the receiver.

The extended flags bit (bit 3) is eliminated. It MUST be set to 0 and MUST be ignored by the receiver.

The host/user bit (bit 6) is eliminated. It MUST be set to 0 and MUST be ignored by the receiver.

The zone bit (bit 7) remains unchanged.

The signatory field (bits 12-15) are eliminated by [5]. They MUST be set to 0 and MUST be ignored by the receiver.

Bits 2,4,5,8,9,10,11 remain unchanged. They are reserved, MUST be set to zero and MUST be ignored by the receiver.

Assignment of any future KEY RR Flag values requires a standards action.

All Protocol Octet values except DNSSEC (3) are eliminated:

Value 1 (Email) is renamed to RESERVED.

Value 2 (IPSEC) is renamed to RESERVED.

Value 3 (DNSSEC) is unchanged.

Value 4 (TLS) is renamed to RESERVED.

Value 5-254 remains unchanged (reserved).

Value 255 (ANY) is renamed to RESERVED.

The authoritative data for a zone MUST NOT include any KEY records with a protocol octet other than 3. The registry maintained by IANA for protocol values is closed for new assignments.

Name servers and resolvers SHOULD accept KEY RR sets that contain KEY RRs with a value other than 3. If out of date DNS zones contain deprecated KEY RRs with a protocol octet value other than 3, then simply dropping the deprecated KEY RRs from the KEY RR set would

invalidate any associated SIG record(s) and could create caching consistency problems. Note that KEY RRs with a protocol octet value other than 3 MUST NOT be used to authenticate DNS data.

The algorithm and public key fields are not changed.

5. Backward Compatibility

DNSSEC zone KEY RRs are not changed and remain backwards compatible. A properly formatted RFC 2535 zone KEY would have all flag bits, other than the Zone Bit (Bit 7), set to 0 and would have the Protocol Octet set to 3. This remains true under the restricted KEY.

DNSSEC non-zone KEY RRs (SIG(0)/TKEY keys) are backwards compatible, but the distinction between host and user keys (flag bit 6) is lost.

No backwards compatibility is provided for application keys. Any Email, IPSEC, or TLS keys are now deprecated. Storing application keys in the KEY RR created problems such as keys at the apex and large RR sets and some change in the definition and/or usage of the KEY RR would have been required even if the approach described here were not adopted.

Overall, existing nameservers and resolvers will continue to correctly process KEY RRs with a sub-type of DNSSEC keys.

6. Storing Application Keys in the DNS

The scope of this document is strictly limited to the KEY record. This document prohibits storing application keys in the KEY record, but it does not endorse or restrict the storing application keys in other record types. Other documents can describe how DNS handles application keys.

7. IANA Considerations

RFC 2535 created an IANA registry for DNS KEY RR Protocol Octet values. Values 1, 2, 3, 4, and 255 were assigned by RFC 2535 and values 5-254 were made available for assignment by IANA. This document makes two sets of changes to this registry.

First, this document re-assigns DNS KEY RR Protocol Octet values 1, 2, 4, and 255 to "reserved". DNS Key RR Protocol Octet Value 3 remains unchanged as "DNSSEC".

Second, new values are no longer available for assignment by IANA and this document closes the IANA registry for DNS KEY RR Protocol Octet Values. Assignment of any future KEY RR Protocol Octet values requires a standards action.

8. Security Considerations

This document eliminates potential security problems that could arise due to the coupling of DNS zone keys and application keys. Prior to the change described in this document, a correctly authenticated KEY set could include both application keys and DNSSEC keys. This document restricts the KEY RR to DNS security usage only. This is an attempt to simplify the security model and make it less user-error prone. If one of the application keys is compromised, it could be used as a false zone key to create false DNS signatures (SIG records). Resolvers that do not carefully check the KEY sub-type could believe these false signatures and incorrectly authenticate DNS data. With this change, application keys cannot appear in an authenticated KEY set and this vulnerability is eliminated.

The format and correct usage of DNSSEC keys is not changed by this document and no new security considerations are introduced.

9. Normative References

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

[2] Eastlake, D., "Domain Name System Security Extensions", RFC 2535, March 1999.

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

[4] Eastlake, D., "DNS Request and Transaction Signatures (SIG(0)s)", RFC 2931, September 2000.

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

10. Authors' Addresses

Dan Massey
USC Information Sciences Institute
3811 N. Fairfax Drive
Arlington, VA 22203
USA

EMail: masseyd@isi.edu

Scott Rose
National Institute for Standards and Technology
100 Bureau Drive
Gaithersburg, MD 20899-3460
USA

EMail: scott.rose@nist.gov

11. Full Copyright Statement

Acknowledgement

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