Internet-Draft
Individual Submission
Expires: August 13, 2002

E. Lewis
NAI Labs

February 15, 2002

# Discussing Application Public Keys in the DNS
# draft-lewis-siked-dnsargs-00.txt

Status of this Memo



Copyright Notice



Abstract

A debate over whether to include public keys for other applications in the Domain Name System has been held ever since the introduction of the DNS Security Extensions. This document records the salient points of the debate.

## 1 Introduction

The Domain Name System [rfc 1034, 1035], lacking sufficient security, lead to the DNS Security Extensions [rfc 2535]. One of the elements of the Extensions is a resource record called the KEY RR. The KEY RR provided the means to store and distribute a public key within the DNS.

One of the ideas promoted early in the development of the DNS Security Extensions is the notion that the KEY RR could be used by applications, that is other than DNS, to store and retrieve keys. This would be

done leveraging the protection now offered to DNS data.

One other resource record that added a bit to the discussion is the CERT RR [rfc 2538]. This record was defined expressly for the use of non-DNS applications. Each CERT RR is defined to hold the bytes of a certificate, leaving the definition of a certificate and the format of the data to the field indicating the type of the certificate. E.g., PGP or X.509.

As the idea of putting so called "application keys" in DNS became more developed, and as the DNS Security Extensions became more developed, the appropriateness of this idea came under scrutiny. In this document the issues that appeared on the mail list are summarized. The reason for doing this is to record the debate in a more concise manner and, more importantly, provide bounds for application key distribution and DNS.

## 2 Reasons For

The reasons for placing application keys into DNS, with protection via DNSSEC are largely based on the assumption that DNS is already there, and will always be there.

### 2.1 DNS As Enabler

In order to have a presence on the network, as in a small organization maintaining a web site, there must be a DNS name server somewhere. DNS is a minimum barrier to entry on the network, so it is a given than it will be run, or run via a contract. Other network services may be offered, but none of them is needed for DNS. DNS is the entry point.

Therefore, placing application keys in DNS is a path of least resistance. Any alternative to DNS means running that service too.

### 2.2 DNS Is As Reliable As Anything

In as much as DNS is the entry point into an organization's resources, when the DNS is down, the network is down. This is a coarse generalization, meaning that if you put data into the DNS, it will be as available as the ability to find the application that would use the key.

While DNS is not perfect when it comes to being a reliable system, using any other means to distribute keys is only going to be as reliable or worse. It is possible that access to a resource is still possible with the DNS down, e.g., decrypting an already received email is possible even if the sender's DNS is down. However, the key is already in hand, accessing the key would be hurt by the DNS outage.

### 2.3 DNS Is Suited For The Job (Code Reuse)

DNS is a look up based system. Data is retrieved based upon a class, name, and type. If the data is there, the data is returned. If the data isn't there, then a no-data answer is returned.

Applications will be able to know exactly where the keys they require are. For example, in SSH, the keys should be located at or near the address record for the server being contacted. The look up for the key should be as simple as the look up for the address.

## 3 Reasons Against

The reasons against placing application keys in DNS are fundamentally rooted in protecting the critical resource that DNS is.

### 3.1 Response Size Overflow

Perhaps the most critical impact of application keys in DNS is a result of their large size. DNS relies on using UDP datagrams for efficient delivery of data. No matter what the size of the datagrams is, loading application keys into DNS will cause the size of certain responses to overflow the UDP boundary. Although steps can be taken to mitigate this, it is inevitable that somewhere the overflow will happen. (Note that DNS overflow may happen for other reasons, but they are beyond the scope of this argument.)

The result of the overflow is a fall back to TCP. Using TCP incurs a much higher overhead than UDP, with this overhead impacting a DNS server. The impact will not be good, at best slowing down a critical server.

### 3.2 Zone Administrator Already Busy

Assuming the security of the application key is provided via DNSSEC verification of the KEY RR holding it, there are two consequences. One is that the zone administrator is now responsible for the zone's health and safety but also the safety of applications running on all the hosts represented in the zone. This can prove to be quite a bit of mission-creep for the administrator. The other consequence is in the next section.

It has been noted that a zone administrator is already responsible for the A record, and a wrong A record would have undesired consequences. The reason KEY records are thought to be more sensitive is that there is an implied liability attached to data meant to secure or, in essence, guarantee data.

### 3.3. Breaking a DNS Key Breaks Application Keys

Assuming the application is trusting DNSSEC to protect the key, then the application is vulnerable to a breaking (exposure, etc) of any of the DNSSEC keys used to validate the data delivered to the application.

### 3.4 Volume of Data

Besides the response size issue, the sheer volume of queries and

responses for application keys is bound to stress the DNS. More stress will lead to more problems and possibly down time.

[Editorial note: During the review of this text, this point was called into question. I'd appreciate if someone felt that this explanation should be strengthened or dropped.]

### 3.5 NAPTR record

DNS shouldn't hold the keys, but it can be a starting place to find them. There already exists a resource record for referring to other services, and it is called NAPTR (rfc 2915). NAPTR isn't a ready solution, but it is a start, and a potentially a paradigm for using DNS to find fragmented data services.

## 4 Summary

This document is intended to capture the results of an already held, heated debate. Comment on the claims here should be directed at correcting inaccuracies, not debating the merits. This document is also not attempting to draw a conclusion from the arguments presented.

## 5 Security Considerations

This document does not have any direct impact on security is as much as the document is just a summary of a discussion.

## 6 IANA Considerations

There are no requests of nor recommendations to IANA in this document.

References

[RFC1034] Mockapetris, P., "Domain Names - Concepts and Facilities", STD 13, RFC 1034, November 1987.

[RFC1035] Mockapetris, P., "Domain Names - Implementation and Specifications", STD 13, RFC 1035, November 1987.

[RFC2535] Eastlake, D., "Domain Name System Security Extensions", RFC 2535, March 1999.

[RFC2538] Eastlake, D., "Storing Certificates in the Domain Name System (DNS)", RFC 2538, March 1999.

[RFC2915] Mealling, M., "The Naming Authority Pointer (NAPTR) DNS Resource Record", RFC 2915, September 2000.

Editor's Address

Edward Lewis
NAI Labs

3060 Washington Rd. (Rte 97)
Glenwood, MD, 21738
USA

EMail: lewis@tislabs.com

## Appendix A. Acknowledgements

The author gratefully acknowledges, in no particular order, the contributions of the following persons:

Jakob Schlyter

members of the namedroppers mailing list

## Full Copyright Statement



## Acknowledgement

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