Internet Draft
Intended status: Experimental
Expires: 11 January 2009

D. Groth
ITUA Inc.
15 July 2008

DNS Encryption
draft-groth-dns-encryption-02.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on 11 January 2009.

Abstract

This document requests IANA registration of a new DNS OpCode and ErrorCode type in facilitating encryption of DNS requests and replies and feed back to the client if plain text requests are not acceptable. Once this OpCode is seen the DNS server attempts to decrypt the request using its private OpenPGP key. Inside the encrypted packet is the AES key which the client expects to be used when the server encrypts a response. A server may advertise that it is capable of DNS encryption by returning OpenPGP fingerprints in TXT records using a similar format to Public Key Association (PKA). The full pubic keys are returned from DNS servers by using a CERT request against the host name(s) of the domain's NS records or via OpenPGP key servers.

Table of Contents

1. Introduction

DNS (RFC 1034, RFC 1035) is a global system; NAPTR records (RFC 3401, RFC 3402, RFC 3403, RFC 3404, RFC 3761), a subset of DNS services, is the first of possibly many such DNS services which reveal sensitive information about the querying agent when requests are sent, regardless of any replies returned. This query information alone is of value to entities in a position to monitor network points.

While there is ongoing work with DNSsec to verify the authenticity of DNS replies which would facilitate the detection of tampering, no active effort is focused on protecting the confidentiality of DNS requests and replies.

2. Why not to use X.509

2.1. X.509 and Web Browser Pop-ups

X.509 certificates (RFC 5280) combined with web browser pop-ups have, in hind sight, proven to be bad for security. The adoption of self-signed and invalid or expired SSL certificates for websites out number certificates that would be deemed valid by most web browsers. However the overall adoption of SSL for websites is very low, less then a tenth of a percent according to Netcraft.

2.2. X.509 and SMTP

X.509 usage with SMTP on the other hand seems to buck the trend observed with web browsers in the absence of pop-up warnings. Adoption estimates currently range up to 50% of all legitimate MTA servers on the Internet. Very few servers use commercial certificates as most people see no advantage in spending money on something they perceive to have no additional tangible benefit, and there is no disadvantage in not purchasing one.

2.3. Problems with X.509 and SMTP

Even with the comparatively high adoption rate of X.509 with SMTP there is still problems. Most problems stem from X.509 extensions being incorrectly set, which in most cases prevents the key pair from being used for both client and server purposes.

This can lead to lost email if the MTA fails or refuses to fall back to non-encrypted transfers.

Due to the way X.509 has been implemented in most software there is no clear path to easily increase security across the internet as a whole for SMTP beyond installing a large number of root certificates, self signed certificates or simply accepting all. This is a big

disadvantage in the long term in achieving strong confidence that servers being connected to really are who they claim to be and not a man-in-the-middle attack.

2.4. The Skype Solution

Looking beyond X.509, it is imperative to reach security paradigms that will actually be beneficial for internet users, rather than road blocks. Skype has proven this to some extent by hiding all the encryption from the user and just letting them get on and use it, rather than annoying the user with a constant barrage of pop-up windows.

2.5. Windows and the Mozilla Foundation

Windows Vista on the other hand has demonstrated how a constant barrage of pop-up windows does little, if anything for security, and only serves to confuse or annoy most users who click through regardless of what the pop-up is, most of the time.

Mozilla Foundation also seems to be ignoring the lessons from the present and the past and has gone down a similar path for certificates of unknown origin. It is now easier to install a root certificate than accept a connection to a server with a self signed or invalid certificate.

3. One way to secure DNS

3.1. Using X.509 and SMTP as a basis

To achieve a beneficial outcome we can review similar protocols that achieve a somewhat successful outcome, since Skype doesn't disclose what they do on a technical level we will instead turn our focus to X.509 with SMTP, which seems to be most widely deployed protocol that is openly documented.

One method to enable encryption with SMTP that cannot be directly transferred to DNS is by escalating the TLS session by sending the 'STARTTLS' command. This is because SMTP only uses ASCII, where as DNS is a binary based protocol.

Instead all we need to do is examine the OpCode contained at the third byte of all DNS requests to determine if the DNS request is encrypted, this draft requests that IANA allocate a new OpCode for this purpose. Once this OpCode is detected, name servers supporting this capability will attempt to decrypt from the 4th byte onwards.

3.2. So why use OpenPGP instead of X.509

Unlike X.509, OpenPGP (RFC 4880) is currently widely used, people have been holding key signing parties for more than a decade. Instead of trying to build completely new infrastructure it makes more sense to make use of what's already available in abundance. Other more structured examples of this include CAcert www.cacert.org and The Gossamer Spider Web of Trust www.gswot.org.

3.3. Extended or No Expiry Keys and Certificates

With current threats existing for very short periods, typically hours to days at most, there is no practical reason for keys to expire in 1 or even 5 years, the primary reason most certificates expire with such frequency is due to monetary reason which is detrimental to security.

OpenPGP keys can be cached which is advantageous in preventing or detecting man in the middle attacks. This would make such attacks more costly to operate.

While not directly related to the this topic, internet browsers do not warn or otherwise notify the user when a certificate for a website has changed, making it virtually impossible to detect a man in the middle attack to be discovered, or even notice once it has ceased. Constantly changing certificates seem to be a bad security practise.

4. Using OpenPGP to Compute Key Confidence

4.1. Confidence Introduction

The word trust has long been abused by mathematicians and cryptographers alike to mean how much confidence you have that the key belongs to the people you think it does. No two people use the OpenPGP trust options in an identical manner, just like no two people would rank a room full of people in the same manner with respect to the task of how much confidence they would place in the person really having the OpenPGP User ID they purport to own.

Currently most X.509 certificates are issued in a way that people see virtually no difference between certificate authorities, it's not until you get into the finer points of their issuing practises and policies that you can begin to build a similar confidence in each certificate authority and the certificates they issue.

The confidence system OpenPGP adopted normally has coarse options in which individuals can be grouped, that isn't to say software built around OpenPGP keys can't build its own system in a much more refined way, either with individual exceptions or by being able to

group individuals into groups or classes of users based on the confidence you have in those people to introduce other keys to you.

4.2. 6 degrees of separation in a practical sense

The PGP web of trust is in part based on the 6 degrees of separation principal, that is everyone in the world knows everyone else through 6 other people.

For the purpose of generating a tangible confidence rating that a host controls a particular host key we will be using arbitrary numbers. Default values of 50 points for fully trusted keys and 30 points for marginally trusted keys are good base values although any arbitrary number should work, but may vary based on individual circumstances.

For anyone we don't know directly we will calculate trust paths between keys by decaying points from the second relationship outwards. Again these are arbitrary values and they can be customised based on individual needs. The general case will use a base of 50% for full trust introduction, 25% trust for marginal introduction, -25% for untrustworthy and 0% for don't know.

You follow trust paths between the local key ring and the key of the name server you are intending to request information from, branching out until you get a points value of 0 or less, or find a direct path to the host key. In either case you no longer follow that branch any further.

For the system to be confident about an OpenPGP key you set the minimum points required, again this can be any arbitrary number such as 100.

4.3. Refining Confidence Scores

The system must have the ability for more finely grained control over individual scores. The default method in OpenPGP is too coarse, and doesn't easily allow you to distinguish between the capabilities of different individuals. For example you trust Bob's judgement when verifying other people holding the right keys more than most. You add an exception for Bob so that anything he trusts will be assigned 75 points instead of 50.

Alice on the other hand is gullible. While you trust Alice, you don't trust the verifications she makes. An exception is made for Alice so that anything Alice trusts will only be assigned 10 points.

In this hypothetical example, even with both Alice and Bob trusting a key your system still wouldn't hit the 100 points needed, so you obviously need to get out and make more friends.

4.4. Out of band fingerprint verification

Just as people already hold key signing parties to verify each others OpenPGP user ids, variations on this would start to appear depending on the level each party needs or wants to secure their resources. It is a reasonable assumption that not all domains need strong protection, and it is up to both the administrators of domains and those making DNS requests to have the right level of security for their needs.

For example the domain of a bank would be at more at risk and hence worth protecting more than a personal domain for someone's blog that gets 10 hits a month. Banks already have a relationship with their customers and it would be easy for them to provide the fingerprint of their user ids on business cards and other stationary items.

This process is commonly used to verify personal keys but there is no reason this concept couldn't be extended so people could also sign host keys.

The worst level of security would be no different to most mail servers using self signed certificates for SMTP-TLS.

4.5. OpenPGP information and commercial services

It is possible to leverage existing OpenPGP web of trust meta information to draw similar security decisions about X.509 certificates issued by commercial Certificate Authorities.

While the focus of this draft is on individuals making their own security choices, there is nothing preventing commercial entities from offering signing services against host keys. The standard practise is for OpenPGP user ids to be signed by multiple entities, and this practise could be utilised by multiple commercial entities, which would potentially increase security.

5. Structure of Host Information in OpenPGP Keys

5.1. FQDN

OpenPGP was designed specifically for text based communication and file encryption, so most of the user id sections of keys contain a name, an email address and possibly a comment. This field can contain any valid UTF-8 string, so computer based systems can easily parse the information present in this string there needs to be a fixed format adhered to, unlike computers humans can more easily cope with variations. The client will compare the host name of the system it connects with to all host names appearing in user ids. All host names MUST BE prefixed with 'dns:';

dns:nameserver.example.com

5.2. The Use of Wild Cards

Wild card host names are allowed, however only one level is allowed, so *.example.com would match nameserver.example.com and a.example.com but would not match this.nameserver.example.com. Multiple wild card characters per host name are not allowed, *.*.example.com

5.3. Extended Information in User Ids

Extended information in OpenPGP user ids such as the information that can be contained in X.509 certificates (RFC 3280) is desirable.

These fields must be only used for informational purposes only. All prefixes must be lower case and the 'dns' prefix is mandatory and must always exist in each host user id however all other prefixes may be absent or must only appear once per user id, for the purposes of this internet draft the only valid prefixes in OpenPGP user ids are;

c: can be used as the prefix for any valid 2 letter ISO country code, e.g. c:ccTLD
st: can be used as the prefix for state, province or territory designation, e.g. st:State Name
l: can be used as the prefix for location, such as town, suburb or city name, e.g. l:Town Name
o: can be used as the prefix for organisation or company name, e.g. o:Example Company
ou: can be used as the prefix for organisation unit, or department in the organisation the information applies to, e.g. ou:Server Administration
uri: can be used as the prefix for valid URIs, e.g. uri:http://www.example.com

5.4. Separation of Fields

The pipe character '|' must be used to separate the different sections, this character must not be used as part of the information contained within any section. URIs must use hex encoding if the pipe character is needed.

The following is an example of a valid OpenPGP user id for the purpose of a DNS name server host name;

dns:example.com|dns:*.example.com|c:ccTLD|st:No State|l:No Place|\
o:Example Company|ou:Server Administration|uri:http://example.com

5.5. Name Servers with Multiple Host Names

A single name server may be authoritative for multiple host names and/or IPs, the 'dns' prefix is the only prefix allowed to exist multiple times on the same user id. If the organisation information is different you could use multiple user ids, one per entity, or multiple OpenPGP keys. The information contained in one user id MUST NOT be mixed or used with host name(s) on other user ids of the same OpenPGP key. Alternatively multiple OpenPGP keys could be used to facilitate this.

5.6. Multiple Host Name Example

The following is an example of a valid OpenPGP key with multiple user ids for the purpose of a DNS name server host name;

```
dns:example.com|dns:*.example.com|c:ccTLD|st:No State|l:No Place|\
o:Example Company|ou:Server Administration|uri:http://example.com
dns:example.net|dns:*.example.net|c:ccTLD|st:No State|l:No Place|\
o:Example Company|ou:Server Administration|uri:http://example.net
```

6. Storing OpenPGP keys in DNS

6.1. Storing Stripped Public Keys

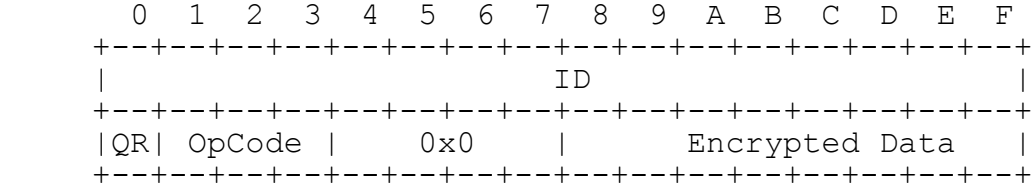
The usual method of storing OpenPGP keys in DNS is to strip all meta information except for the user id(s) and transmit in binary format. Dig and other utilities output this information in base64 encoding.

6.2. Retrieving Meta Information

To be able to calculate confidence scores the full host keys will need to be retrieved from PGP key servers, this can be a timely process and will need to be periodically re-run to ensure signatures are still valid.

7. DNS Packet Structure

7.1. Unencrypted DNS Packet Structure



where:

ID The first 2 bytes must be random but distinct from the real query ID inside the encrypted packet, this is to minimise

	the risk of spoofed error replies.
QR	A one bit field for backward compatibility, it must always be 0x0 for questions and 0x1 for replies.
OPCODE	A new OpCode needs to be allocated by IANA for this purpose to be compatible with existing DNS infrastructure.
DATA	RFC 3766 indicates that 2048 bit RSA and 128 bit AES should be secure until 2016, at which point 4096 bit RSA and 256 bit AES MUST BE used however these key sizes may be prior to this date as well.

7.2. Encrypted DNS Packet Structure

When decoding the encrypted packet the first 4 bytes of the DNS request should be discarded, the rest of the DNS request should be encrypted using the public key of the DNS server.

Once the packet has been decrypted, the next 32 bytes is the AES key and possibly null padding if the AES key is less than 256 bit. The AES key can be 16, 24 or 32 bytes in length depending if it is a 128, 192 or 256 bit key being sent. The client is expecting the reply to be returned encrypted with this AES key.

The packet contains the DNS request from the 33rd byte which can then be processed in the same manner as any other DNS request except that the reply must be encrypted using the AES key.

If the AES key is not 256 bit or 32 bytes there must be null padding used to ensure that no part of the DNS request is found in the first 32 bytes of the DNS request.

8. Advertising Encryption Capability With Additional Records

8.1. Fingerprints in TXT Records

The use of TXT records to associate fingerprints with host names will make it easier to use OpenPGP on subsequent connections as it can simply be loaded from the local key ring. These fingerprints should be returned by authoritative name servers and as glue records from registries and registrars.

8.2. Structure of TXT Records

Unfortunately PKA was designed in a very email centric manner so it isn't possible to use PKA format directly, however using TXT records which follow a similar formating to PKA is possible but with a few minor differences. Usually _pka is placed in the hostname replacing the at symbol, with host names this distinction isn't needed, and the hostnames instead can be prefixed with _fingerprint instead to avoid confusion of this record type and PKA information.

As with PKA, semi-colons are used to separate the three fields. The fields are; v= for the revision number, t= for the type, which can be OpenPGP or PKI, f= for the full 40 byte hexadecimal fingerprint of the public key.

8.3. TXT Record Example

```
_fingerprint.example.com. IN TXT \
    "v=1;t=OpenPGP;f=0123456789ABCDEF0123456789ABCDEF01234567"
```

8.4. Glue Records

If registries and registrars allowed fingerprint glue records in their respective zones and returned these with any IP glue records, this would minimise the number of packets required to facilitate encryption. Each glue record must be per name server host name, not per zone to minimise the disruption caused when IPs for hostnames change.

8.5. Additional Records

TXT fingerprint records must be returned as additional records when a client makes a NS request on the zone that shares the same domain as any name server host name(s). Additionally the same records must be returned for any matching TXT requests.

9. Security Considerations

9.1. DNS is inherently insecure

DNS encryption does not introduce any new security issues beyond any already present in DNS, DNS is inherently insecure, and this draft attempts to solve some of the attacks that can occur with DNS. As DNS is further extended beyond its original uses, it has become more imperative to protect the confidentiality of both the query and the response, however at the cost of efficiency there is a trade off towards information leakage.

In an ideal world if the server responds that the request was corrupt or unable to decrypt the request should be sent to the next name server, once the pool of name servers is exhausted the recursive look-up could fall back to plain text mode to ensure best effort is met. Any software implementing this internet draft must implement the ability to have domains that are exempt from using plain text mode.

9.2. Reducing Information Leaks

During a normal DNS look-up the full host name is sent to each name server, and then either a suitable reply is returned, record not

found or other error, or a NS to submit a new query to. While this method appears to be the most efficient, when switching between systems that can handle encrypted look-ups and systems that can't this could leak too much information about the information being sought after.

DNS clients and resolvers must split the gTLD or ccTLD zone name from the fully qualified host name being requested. The zone information must be used to find relevant NS records and only the relevant name servers that may have the information must receive the full query.

10. IANA Considerations

This internet draft requests that IANA delegate a new OpCode so name servers can distinguish encrypted DNS requests, this is critical that it appear at the 3rd byte and must be allocated in the original OpCode space only.

This internet draft requests that IANA delegate a new ErrorCode so name servers can respond to plain text requests that they only reply to encrypted DNS requests, this isn't critical and only needs to be in EDNS error space.

11. Conclusions

As with other protocols, it is becoming imperative to prevent disclosure of dialogues between the intended client and server in the interest of security and privacy. Even though DNS is a public database, the general public is unaware of how DNS works or that their requests and replies can be intercepted or altered.

If a large number of popular name servers were to adopt strong cryptography, many attacks on DNS would be rendered useless.

Even though this draft is specifically about securing DNS by using OpenPGP key pairs, there is nothing special about the methods used or the way the structure of the information in OpenPGP keys is implemented that would prevent them from being re-utilised for other purposes.

12. Informative References

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

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC3401] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part One: The Comprehensive DDDS", Network Working Group, RFC 3401, October 2002.

[RFC3402] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part Two: The Algorithm", Network Working Group, RFC 3402, October 2002.

[RFC3403] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database", Network Working Group, RFC 3403, October 2002.

[RFC3404] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part Four: The Uniform Resource Identifiers (URI) Resolution Application", Network Working Group, RFC 3404, October 2002.

[RFC3761] Faltstrom, P., "The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)", Network Working Group, RFC 3761, April 2004.

[RFC3766] Orman, H., "Determining Strengths For Public Keys Used", VPN Consortium, RFC 3766, April 2004.

[RFC4398] Josefsson, S., "Storing Certificates in the Domain Name System (DNS)", Network Working Group, RFC 4398, March 2006.

[RFC4880] Callas, et. al., "OpenPGP Message Format", Network Working Group, RFC 4880, November 2007.

[RFC5280] Cooper, et. al., "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", Network Working Group, RFC 5280, May 2008.

13. Acknowledgements

This document was prepared using 2-Word-v2.0.template.dot. Although it seems to mostly work in Open Office its a shame that ODF wasn't specifically supported. This document also used draft-timms-encrypt-naptr-00.txt as inspiration. Big thanks to Philipp Guehring and Ian G. for letting me bounce initial ideas off them, as well other finer points along the way. Suggestions and tips from Paul Vixie, Simon P. Ditner, and many others on mailing lists and forums. Finally a great big thanks to Denise Khoo, without her help none of this would be possible.

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

Author's Addresses

Duane Groth
Internet Telephony Users Association Inc.
P.O. Box 75
Banksia NSW 2216
Australia

Email: support@e164.org

Intellectual Property Statements

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.