Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 16, 2014

P. Wouters
Red Hat

July 15, 2013

Using DANE to Associate OpenPGP public keys with email addresses
draft-wouters-dane-openpgp-00

Abstract

OpenPGP is a message format for email (and file) encryption, that lacks a standarized secure lookup mechanism to obtain OpenPGP public keys. This document specifies a standarized method for securely publishing and locating OpenPGP public keys in DNS using a new OPENPGPKEY DNS Resource Record.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on January 16, 2014.

Table of Contents

1. Introduction 3
1.1. Terminology 3
2. The OPENPGPKEY Resource Record 4
2.1. Location of the OpenPGPKEY record 4
2.2. The OPENPGPKEY RDATA Format 5
3. OpenPGP public key considerations 5
3.1. Public Key UIDs and email addresses 5
3.2. Public Key UIDs and IDNA 5
3.3. Public Key UIDs and synthesized DNS records 5
3.4. Public Key size and DNS record size 6
4. Security Considerations 6
4.1. Email address information leak 7
4.2. OpenPGP security and DNSSEC 7
4.3. MTA behaviour 7
4.4. MUA behaviour 8
4.5. Email client behaviour 8
4.6. Subject: line encryption 9
5. IANA Considerations 9
5.1. OPENPGPKEY RRtype 9
6. Acknowledgements 9
7. References 9
7.1. Normative References 9
7.2. Informative References 10
Author's Address 10

1. Introduction

To encrypt a message to a target recipient using OpenPGP [RFC4880], possession of the recipient's OpenPGP public key is required. To obtain that public key, two problems need to be solved by the sender's email client, MUA or MTA. Where does one find the recipient's public key and how does one trust that the found key actually belongs to the intended recipient.

Obtaining a public key is not a straightforward process as there are no standarized locations for publishing OpenPGP public keys indexed by email address. Instead, OpenPGP clients rely on "well known key servers" that are accessed using the web based HKP protocol or manually by users using a variety of different front-end web pages.

Currently deployed key servers have no method of validating any uploaded OpenPGP public key. The key servers simply store and publish. Anyone can add public keys with any name or email address and anyone can add signatures to any other public key using forged malicious identities. For example, bogus keys of prominent dissidents have been uploaded to these well-known key servers in attempts to capture encrypted email. Furthermore, once uploaded, public keys cannot be deleted. People who did not pre-sign a key revocation and who have lost access to their private key can never remove their public key from these key servers.

The lack of association of email address and public key lookup is also preventing email clients, MTAs and MUAs from encrypting a received message to the target receipient forcing the software to send the message unencryped. Currently deployed MTA's only support encrypting the transport of the email, not the email contents itself.

This document describes a mechanism to associate a user's OpenPGP public key with their email address, using a new DNS RRtype. This is similar to the SSHFP [RFC4255] RRType, except that this method associates keys with users, not hosts.

The proposed new DNS Resource Record type is secured using DNSSEC. This trust model is not meant to replace the "web of trust" model. However, it can be used to encrypt a message that would otherwise have to be sent out unencrypted, where it could be intercepted by a third party in transit or located in plaintext on a storage or email server.

1.1. Terminology

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].

This document also makes use of standard DNSSEC and DANE terminology. See DNSSEC [RFC4033], [RFC4034], [RFC4035], and DANE [RFC6698] for these terms.

2. The OPENPGPKEY Resource Record

The OPENPGPKEY DNS resource record (RR) is used to associate an end entity OpenPGP public key with an email address, thus forming a "OpenPGP public key association".

The type value allocated for the OPENPGPKEY RR type is [TBD]. The OPENPGPKEY RR is class independent. The OPENPGPKEY RR has no special TTL requirements.

2.1. Location of the OpenPGPKEY record

Domain names are prepared for requests in the following manner.

1. The user name (the "left-hand side" of the email address, called the "local-part" in the mail message format definition [RFC2822] and the "local part" in the specification for internationalized email [RFC6530]), is encoded with Base32 [RFC4648], to become the left-most label in the prepared domain name. This does not include the "@" character that separates the left and right sides of the email address.

2. The string "_openpgpkey" becomes the second left-most label in the prepared domain name.

3. The domain name (the "right-hand side" of the email address, called the "domain" in RFC 2822) is appended to the result of step 2 to complete the prepared domain name.

For example, to request an OPENPGPKEY resource record for a user whose address is "hugh@example.com", you would use "d1qmeq0._openpgpkey.example.com" in the request. The corresponding RR in the example.com zone might look like:

d1qmeq0._openpgpkey.example.com. IN OPENPGPKEY <encoded public key>

Design note: Encoding the user name with Base32 allows local parts that have characters that would prevent their use in domain names. For example, a period (".") is a valid character in a local part, but would wreak havoc in a domain name. Similarly, RFC 6530 allows non-ASCII characters in local parts, and encoding a local part with non-

ASCII characters with Base32 renders the name usable in the DNS.

2.2. The OPENPGPKEY RDATA Format

The RDATA (or RHS) of an OPENPGPKEY Resource Record contains a single value consisting of a [RFC4880] formatted OpenPGP public keyring encoded in base32 as specified in [RFC4648].

3. OpenPGP public key considerations

Once an OPENPGPKEY resource record has been found and the OpenPGP public keyring has been base32 decoded, the right public key must be located inside the keyring. For a public key in the keyring to be usable, the public key has to have a key uid as specified in [RFC4648] that matches the email address for which the OPENPGPKEY RR lookup was performed.

3.1. Public Key UIDs and email addresses

An OpenPGP public key can be associated with multiple email addresses by specifying multiple key uids. The OpenPGP public key obtained from a OPENPGPKEY RR can be used as long as the target recipient's email address appears as one of the OpenPGP public key uids. The name part (left of the @) should appear in the native format, not its base32 encoding that was used to lookup the OPENPGPKEY RR.

3.2. Public Key UIDs and IDNA

Internationalized domains that use non-ascii characters (U-label) are encoded in DNS using IDNA [RFC5891] - also referred to as punycode or A-label. When matching OpenPGP public key uids, both the email address specified using U-label and A-label should be considered as valid public key uids.

3.3. Public Key UIDs and synthesized DNS records

CNAME's (see [RFC2181]) and DNAME's (see [RFC6672]) can be followed to obtain an OPENPGPKEY RR, as long as the original recipient's email address appears as one of the OpenPGP public key uids. For example, if the OPENPGPKEY RR query for hugh@example.com (d1qmeq0._openpgpkey.example.com) yields a CNAME to d1qmeq0._openpgpkey.example.net, and an OPENPGPKEY RR for d1qmeq0._openpgpkey.example.net exists, then this OpenPGP public key can be used, provided one of the key uids contains "hugh@example.com". This public key cannot be used if it would only contain the key uid "hugh@example.net".

If one of the OpenPGP key uids contains only a single wildcard as the LHS of the email address, such as "*@example.com", the OpenPGP public key may be used for any email address within that domain. Wildcards at other locations (eg hugh@*.com) or regular expressions in key uids are not allowed, and any OPENPGPKEY RR containing these should be ignored.

3.4. Public Key size and DNS record size

Although the reliability of the transport of large DNS Resoruce Records has improved in the last few years, it is still recommended to keep the DNS records as small as possible without sacrificing the security properties of the public key. The algorithm type and key size of the OpenPGP keypair should not be modified to accomodate this section.

[Should a statement be made on the number of signatures left on the key? Should there be _any_ signatures other than the self-signed one?]

OpenPGP supports various attributes that do not contribute to the security of a key, such as an embedded image file. It is recommended that these properties are not exported to OpenPGP public keyrings that are used to create OPENPGPKEY Resource Records.

4. Security Considerations

The main goal of the OPENPGPKEY resource record is to stop passive attacks against plaintext emails. While it can also twart some active attacks (such as people uploading rogue keys to keyservers in the hopes that others will encrypt to these rogue keys), this resource record is not a replacement for verifying OpenPGP public keys via the web of trust signatures, or manually via a fingerprint verification.

Various components could be responsible for encrypting an email message to a target recipient. It could be done by the sender's email client or software plugin, the sender's Mail User Agent (MUA) or the sender's Mail Transfer Agent (MTA). Each of these have their own characteristics. An email client can interact with the user to make a decision before continuing. The MUA can only accept or refuse a message. The MTA must deliver the message, either as-is, or encrypted. Each of these programs should ensure that an unencrypted received email message will be encrypted whenever possible.

4.1. Email address information leak

DNS zones that are signed with DNSSEC using NSEC for denial of existence are susceptible to zone-walking, a mechanism that allow someone to enumerate all the names in the zone. Someone who wanted to collect email addresses from a zone that uses OPENPGPKEY might use such a mechanism. DNSSEC-signed zones using NSEC3 for denial of existence are significantly less susceptible to zone-walking. Someone could still attempt a dictionary attack on the zone to find OPENPGPKEY records, just as they can use dictionary attacks on an SMTP server or grab the entire contents of existing PGP key servers to see which addresses are valid.

4.2. OpenPGP security and DNSSEC

DNSSEC key sizes are chosen based on the fact that these keys can be rolled with next to no requirement for security in the future. If one doubts the strength or security of the DNSSEC key for whatever reason, one simply rolls to a new DNSSEC key with a stronger algorithm or larger key size.

The same does not apply to OpenPGP encrypted messages. Users have an expectation that their OpenPGP encrypted messages cannot be decrypted for years or decades into the future. Changing to a new OpenPGP keypair is also a costly and manual process that people tend to avoid when possible.

This effectively means that anyone who can obtain a DNSSEC private key of a domain name via coercion, theft or brute force calculations, can replace any OPENPGPKEY record in that zone and all of the delegated child zones, irrespective of the key length strength of the OpenPGP keypair.

Therefore, DNSSEC is not an alternative for the "web of trust" or for manual fingerprint verification by humans. It is a solution aimed to ease obtaining someone's public key, and without manual verification should be treated as "better then plaintext" only. While this twarts all passive attacks that simply capture and log all plaintext email content, it is not a security measure against active attacks.

4.3. MTA behaviour

An MTA could be operating in a stand-alone mode, without access to the sender's OpenPGP public keyring, or in a way where it can access the user's OpenPGP public keyring. Regardless, the MTA SHOULD NOT modify the user's OpenPGP keyring.

An MTA sending an email SHOULD NOT add the public key obtained from

an OPENPGPKEY resource record to a permanent public keyring for future use beyond the TTL.

If the obtained public key is revoked, the MTA MUST NOT use the key for encryption, even if that would result in sending the message in plaintext.

[What is the correct behaviour of an MTA when it receives an encrypted message from a MUA that is encrypted to a different key then the one listed in the recipient's OPENPGPKEY record? Encrypt the encrypted message? Refuse to send out the message? Don't even look up the OPENPGPKEY record and pass unmodified?]

If an OPENPGPKEY resource record is received without DNSSEC protection, it MUST NOT be used. If the DNS request returned an "indeterminate" or "bogus" answer, the MTA should queue the plaintext message and try encryption and delivery again at a later time.

If multiple non-revoked OPENPGPKEY resource records are found, the MTA should pick the most secure RR based on its local policy.

4.4. MUA behaviour

If the public key for a recipient obtained from the locally stored public keyring differs from the recipient's OPENPGPKEY resource record, the MUA SHOULD NOT accept the message for delivery.

If a MUA detects that a locally stored public key is present in an OPENPGPKEY resource record, and the OPENPGPKEY RR version of the public key is revoked, the MUA SHOULD reject the message for delivery.

If multiple non-revoked OPENPGPKEY resource records are found, the MUA should pick the most secure RR based on its local policy.

4.5. Email client behaviour

An email client MAY interact with a user to add the contents from an OPENPGPKEY resource record into the user's permanent public keyring.

If the public key for a recipient obtained from the locally stored public keyring differs from the recipient's OPENPGPKEY resource record, the email client SHOULD ask the user which key to use for encryption. The email cilent SHOULD allow encrypting to both public keys.

An email client that is encrypting a message SHOULD clearly indicate to the user the difference between encrypting to a locally stored and

manually verified public key and encrypting to an automatically obtained public key via an OPENPGPKEY resource record that has not been manually verified.

If a MUA detects that a locally stored and manually verified public key is present in an OPENPGPKEY resource record, and the OPENPGPKEY RR version of the public key is revoked, the MUA SHOULD warn the user and give them the chance to not sent the message at all.

If multiple non-revoked OPENPGPKEY resource records are found, the MUA should pick the most secure RR based on its local policy.

4.6. Subject: line encryption

Often, encrypting an email does not cause its Subject: line to be encrypted. If the email client, MUA or MTA automatically encrypt an email based on the existence of an OPENPGPKEY record, it should clear or replace the Subject: header with a notification that does not expose the original subject line. It should prepend the original Subject: line to the first line of the body of the email message before encryption. This allows a receiving email client to decrypt the message and replace the Subject: line to its original decrypted form when presenting the user with the decrypted email message.

5. IANA Considerations

5.1. OPENPGPKEY RRtype

This document uses a new DNS RR type, OPENPGPKEY, whose value [TBD] has been allocated by IANA from the Resource Record (RR) TYPEs subregistry of the Domain Name System (DNS) Parameters registry.

6. Acknowledgements

This document is based on RFC-4255 and draft-ietf-dane-smime whose authors are Paul Hoffman, Jacob Schlyter and W. Griffin.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S.

Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006.

[RFC4880] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", RFC 4880, November 2007.

[RFC5891] Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", RFC 5891, August 2010.

7.2. Informative References

[RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, July 1997.

[RFC2822] Resnick, P., "Internet Message Format", RFC 2822, April 2001.

[RFC4255] Schlyter, J. and W. Griffin, "Using DNS to Securely Publish Secure Shell (SSH) Key Fingerprints", RFC 4255, January 2006.

[RFC6530] Klensin, J. and Y. Ko, "Overview and Framework for Internationalized Email", RFC 6530, February 2012.

[RFC6672] Rose, S. and W. Wijngaards, "DNAME Redirection in the DNS", RFC 6672, June 2012.

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.

Author's Address

Paul Wouters
Red Hat

Email: pwouters@redhat.com