Network Working Group
Internet-Draft
Category: Informational
Expires: July 2008

M.Pearson
State Services Commission
F. Hendrikx

M. Hunt
Catalyst IT

January 10, 2008

Applicability Statement for SecureMail: A framework for increasing email security

<draft-pearson-securemail-01.txt>

Status of this Memo

This document is an Internet-Draft and is subject to all provisions of Section 3 of RFC 3978.

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 July 2008.

This document is an individual submission. Comments are solicited and should be addressed to the author(s).

Copyright Notice

Abstract

This document provides an Applicability Statement for Securemail, a framework proposal for secure transmission and better authentication of email based on current Internet standards. The SecureMail framework proposes the use of Transaction Layer Security (TLS), the Sender Policy Framework (SPF) and Sender ID to support secure email communication between internet servers with some assurance of the authenticity of the message sender.

Table of Contents

1. Introduction 4
   1.1. Terminology 4
2. Background 4
3. Motivation 5
4. SecureMail 5
   4.1. Confidentiality - TLS 6
      4.1.1. Why TLS? 6
      4.1.2. Why Anonymous Key Exchange? 6
   4.2. Authentication - SPF And Sender ID 6
5. Implementation Standards For All Mail Servers 7
6. Implementation Standards For SecureMail Servers 7
   6.1. Discovery mechanisms 7
   6.2. Cryptography Standard 8
7. Security Considerations 9
   7.1. TLS 9
   7.2. Man In The Middle 9
   7.3. DNS Attack 9
8. Other Considerations 9
   8.1. Store And Forward 10
   8.2. Mixing Secure And Insecure Receiving 10
   8.3. Mixing Secure And Insecure Sending 10
9. Email Distribution 10
10. Timekeeping Requirements 11
11. Future Development 11
12. IANA Considerations 11
13. Acknowledgements 11
14. References 12
   14.1. Normative References 12
   14.2. Informative References 12
15. Authors' Addresses 13

1. Introduction

This document provides an Applicability Statement for Securemail, a pragmatic framework to increase email security based on current internet standards. SecureMail secures the transport of email in a way analagous to the postal system and paper mail.

The SecureMail framework is being proposed as a replacement for the New Zealand Government's existing proprietary secure email system, SEEMail. It is expected to provide a useful framework for secure transmission of email generally and makes use of common technologies to achieve a greater degree of transmission security and authentication than standard internet email.

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 BCP 14, RFC 2119 [RFC2119].

2. Background

The New Zealand government has been using secure email since 1999.

An initial pilot used secure email clients with individual users being issued Public Key Infrastructure (PKI) digital certificates on smart cards. This worked, but had a number of issues:

- Content: All content was encrypted to individuals; therefore agencies were unable to enforce inappropriate content policies

- Accessibility: Vendors could not guarantee a continued long-term technical ability to decrypt material; therefore agencies were unable to maintain the material in encrypted form for long-term access

- Client software variability: The four trial agencies between them had 7 different email clients; therefore users found email clients behaved differently, creating user support issues

- Inconvenience: Users found it inconvenient to unlock the smartcard with a PIN after a 30 second timeout.

The project then successfully piloted server-to-server PKI-based secure email, with each server being issued a domain-based digital certificate and securing all messages to other participating servers.

This infrastructure, called SEEMail, is currently used by more than 60 government agencies to securely exchange email (and attachments) over the Internet. However, the agencies have some issues with the infrastructure:

- Cost: Commercial secure email software is licensed on a per mailbox basis, making it prohibitively expensive for larger agencies, wanting to use commercial software, where not all staff need secure email. In addition, the software often has management intensive processes associated with setting up secure accounts.

- Experience: Running a PKI application is technically challenging. As staff change, there is a loss of experience associated with PKI implementation and maintenance.

- Robustness: When a PKI-based secure email system goes wrong, it can disrupt communication. For instance, whenever the Certificate Revocation List (CRL) is unavailable, email applications may halt email delivery until the information is available again - and yet, email is about speedy delivery. In addition, the behaviour of email applications in the event of conditions such as certificate expiry is not always well understood. Very few commercial certificate authorities offer a service to generate broken, corrupt, or expired certificates, to test the behaviour of vendor products.

3. Motivation

Government agencies and other organisations want to be able to communicate securely with their customers using an email system that is equivalent, in terms of security, to postal mail.

In the ideal situation - where government customers' ISPs supported SEEmail - the government agencies would utilise the existing SEEMail infrastructure to conduct secure communications with their customers.

However, the ISPs providing email infrastructure for agency clients are concerned with:

- Cost: Who will pay for the software?

- Experience: Who will implement and maintain it?

- Robustness: Will it cause problems and will it scale?

Clearly, SEEMail is not going to be easily scalable to the Internet as a whole.

4. SecureMail

The NZ Government's experience with server-to-server secure email is that it can work exceedingly well. SecureMail is an application of existing standards to achieve secure email without the limitations of SEEMail.

It uses Transport Layer Security (TLS) for confidentiality and integrity of the message during transport, and Sender ID [RFC4406] and the Sender Policy Framework (SPF) [RFC4408] to authenticate the sender.

It is intended to secure IN-CONFIDENCE email communications between government, business and citizens.

4.1. Confidentiality - TLS

SecureMail uses TLS to create an encrypted connection that plaintext messages are passed through. SecureMail connections are negotiated server-to-server using anonymous key exchange. The connections are set up as required using anonymous Diffie-Hellman key exchange, rather than through a pre-arranged agreement or approval list.

4.1.1. Why TLS?

TLS is a gateway based model, operating between SMTP servers.

- Cost: There are no significant capital costs. TLS is available with most email systems

- Experience: It is already active (slightly more than 10% of 4000 New Zealand domains tested already had TLS enabled on their SMTP servers)

- Robustness: TLS is a mature standard and operates transparently to secure transport protocols

4.1.2. Why Anonymous Key Exchange?

This removes the need for any centralised Public Key Infrastructure, and resolves several of the issues discovered using SEEMail.

4.2. Authentication - SPF And Sender ID

The SecureMail framework uses two sender authentication standards: SPF and Sender ID.

SPF operates at the session layer rather than on the email's content. The advantage of this is that it can validate the address before the message is accepted for delivery by the receiving server. However, this also means that the "From:" address that the recipient user sees is not necessarily that which was authenticated.

SenderID mitigates this risk as, in its PRA mode [RFC4407], it checks the sender information contained in the content of the email message against the published information for the domain.

5. Implementation Standards For All Mail Servers

For sending:

- Mail server domain MUST have an SPF record so that the server can be authenticated as an approved sender of the message

- Mail server SHOULD try to send messages securely using a TLS connection

For receiving:

- Servers with "securemail" as the left-most part of their hostname name SHALL only accept email if a TLS connection is established

- Other servers, SHOULD attempt to accept messages securely via a TLS connection, but otherwise allow an insecure connection

- Server SHALL enforce the mail sending policy specified by a sending domain's SPF record (if any)

- Server SHALL enforce the mail sending policy specified by a sending domain's Sender ID record (if any)

6. Implementation Standards For SecureMail Servers

6.1. Discovery mechanisms

Given that a SecureMail server will only ever receive email securely, it cannot be considered a genuine MTA (according to [RFC3207]). This RFC clearly states that publicly-referenced MTAs must not require TLS connections.

A SecureMail server cannot therefore be listed in the MX records for a domain. Instead, we propose a standard naming convention for servers that implement the SecureMail framework.

For receiving:

- SecureMail servers have a standard naming convention, with "securemail" as the leftmost part of the domain name, for example, securemail.example.com

- SecureMail servers MUST refuse to accept email from senders under any of the following conditions:

  - the sender's SPF record
    does not exist; or
    does not prohibit all other senders "-all"; or
    upon evaluation, returns any result other than "Pass"

- the sender's Sender ID record
  does not exist; or
  does not prohibit all other senders "-all"; or
  upon evaluation, returns any result other than "Pass"

- the sender's TLS connection
  does not exist; or
  does not meet the minimum cryptography standards

For sending

- SecureMail servers MUST have a valid Sender ID record specifying valid senders and prohibiting all other senders ("-all"), so that the message envelope and sender information in the content can be authenticated.

- SecureMail servers MUST refuse to send email (and return it to the sender) under any of the following conditions:

  - the receiver's TLS connection does not exist

  - the receiver's TLS connection does not meet the minimum cryptography standards.

6.2. Cryptography Standard

The minimum cryptography standards are defined by the commonly available implementations of TLS. SecureMail servers MUST support Diffie-Helman key exchange, 256-bit AES encryption and SHA1 message digest. In future these requirements are expected to require ECDSA key exchange and SHA-256 message digest. This move is dependent on the work in progress on TLS Version 1.2 [I-D.ietf-tls-rfc4346-bis] and support for Elliptic Curve Cryptography and alternate MAC algorithms described in TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode [I-D.ietf-tls-ecc-new-mac].

Server crypto modules SHOULD be evaluated to FIPS140-2 and SHOULD be combined with a Common Criteria evaluation of the product to EAL3, or higher, by the Australasian Information Security Evaluation Programme (AISEP), or equivalent.

7. Security Considerations

7.1. TLS

Although TLS is provided as a library (e.g., OpenSSL), the MTA still needs to be able to use it correctly. Administrators need to ensure they use an implementation that has been tested.

7.2. Man In The Middle

Before the TLS session is established, the SecureMail approach is vulnerable to a man-in-the-middle (MITM) attack. The MITM sets up secure TLS sessions with the sending and receiving servers, who believe they are communicating with each other. Both links could appear to be fully authenticated if the DNS records are modified or if the attacker can force packets between the two servers' IP addresses to pass through the attacker's device (alternatively, the attacker might not bother setting up the attacker to recipient link).

DNSSEC [RFC4033] or the use of mutually authenticated TLS (instead of anonymous TLS) would mitigate the risk. It would require PKI certificates for each mail server but, unlike S/MIME, the certificates would not need to be pre-positioned as they can be passed in the handshaking phase. A directory would still be required, but only to publish CRLs (Certificate Revocation Lists).

In situations requiring higher levels of assurance, it is recommended that PKI certificates be exchanged between the two parties.

7.3. DNS Attack

If the sending domain's DNS record is compromised and the SPF record is modified to include an attacker's address, that device would appear to be authorised to send mail on the domain's behalf. This type of attack is unlikely as the types of threat agents (spammers, phishers, etc.) are unlikely to want the additional effort and risk of modifying DNS servers to pretend to originate from a SecureMail address. As with the example above, the vulnerability could be minimised by the use of mutually authenticated TLS (i.e., the attacker would also have to get a legitimate key pair and certificate, and the attack would be traceable through that certificate).

8. Other Considerations

SecureMail is intended to provide better security during transmission for email sent over the Internet between two mail servers. It is not intended to specify how the sender or receiver manages their own email security.

8.1. Store And Forward

Organisations that use an intermediate mail server between the sending and recipient servers (e.g., store and forwarded through an ISP or an application-level firewall) can break security. The configuration to make this work could make the SPF look-up process ineffectual and the mail may be transmitted in plaintext at this point.

8.2. Mixing Secure And Insecure Receiving

It is recommended that received SecureMail messages be kept separate from other messages. Otherwise it will be difficult to determine whether the message was authenticated (via SecureMail), or arrived unauthenticated via the normal mail system.

The method proposed to mitigate this risk is to have alternative accounts or inboxes for SecureMail versus other mail. Based on the "To:" address and the mailbox a message is in, the user knows whether the sender has been validated.

Alternatively or additionally, the receiving mail server could mark incoming messages with their authentication level in a similar way to junk mail marking employed in some systems (the normal mail system would have to check/remove similar markings in email that arrived through 'normal' channels).

8.3. Mixing Secure And Insecure Sending

When a user sends a message securely, they have no control or knowledge of how the message will be delivered. Their own system may not be configured to correctly secure the message.

A user can assume that a SecureMail server, identified by "securemail" as the leftmost part of the hostname, will fail-safe and refuse to accept insecure messages sent from the user's domain.

The user can test this, using the free testing tool service at http://tools.secmx.org/.

9. Email Distribution

Users who access a SecureMail server should connect to the server using a secure connection (e.g., using POP3/SSL or a secure internal network). Remote users should only connect to such a mail server utilising equipment which has been appropriately certified and accredited for that purpose.

10. Timekeeping Requirements

SecureMail servers should maintain time synchronisation using Network Time Protocol (NTP).

11. Future Development

In the future, thought will be given to improving the security, through public key technology or other technologies not involving digital certificates, such as Kerberos. Support for DomainKeys Identified Mail (DKIM) Signatures [RFC4871] may be recommended in a future version of this applicability statement.

The implications of the SUBMIT protocol [RFC4409] will be considered in a future version of this applicability statement.

12. IANA Considerations

This document has no actions for IANA.

13. Acknowledgements

The authors would like to acknowledge contributions from Geoff Cant and Hector Santos.

14. References

14.1. Normative References

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

[RFC3207] P. Hoffman, "SMTP Service Extension for Secure SMTP over Transport Layer Security", RFC 3207, February 2002.

[RFC4406] J. Lyon, M. Wong, "Sender ID: Authenticating E-Mail", RFC 4406, April 2006.

[RFC4407] J. Lyon, "Purported Responsible Address in E-Mail Messages", RFC 4407, April 2006.

[RFC4408] M. Wong, W. Schlitt, "Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1", RFC 4408, April 2006.

14.2. Informative References

[RFC4033] R. Arends, R. Austein, M. Larson, D. Massey, S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC4409] R. Gellens, J.Klensein, "Message Submission for -Mail", RFC 4409, April 2006.

[RFC4871] E. Allman, J. Callas, M. Delany, M. Libbey, J. Fenton, M. Thomas, "DomainKeys Identified Mail (DKIM) Signatures", RFC 4871, May 2007.

[I-D.ietf-tls-ecc-new-mac]
E. Rescorla, "TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode", draft-ietf-tls-ecc-new-mac-02 (work in progress), December 2007.

[I-D.ietf-tls-rfc4346-bis]
T. Dierks, E. Rescorla, "The TLS Protocol Version 1.2", draft-ietf-tls-rfc4346-bis-07 (work in progress), November 2007.

15. Authors' Addresses

Michael Pearson
State Services Commission
100 Molesworth Street
Wellington
New Zealand

EMail: mike.pearson@ssc.govt.nz

Ferry Hendrikx
Wellington
New Zealand

EMail: ferry.hendrikx@gmail.com

Matthew Hunt
Catalyst IT
Level 2
150-154 Willis Street
Wellington
New Zealand

EMail: matt@catalyst.net.nz

Copyright Statement

Acknowledgment

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