Network Working Group
Internet-Draft
Intended status: Informational
Expires: July 20, 2014

D. Crocker
Brandenburg InternetWorking
P. Resnick
Qualcomm Technologies, Inc.

January 16, 2014

STRINT Workshop Position Paper: Levels of Opportunistic Privacy Protection for Messaging-Oriented Architectures
draft-crocker-strint-workshop-messaging-00

Abstract

Given a concern for pervasive monitoring, messaging information needing protection includes primary payload, descriptive meta-data, and traffic-related analysis. Complete protection against pervasive monitoring (PM), for traffic through complex handling sequences, has not yet been achieved reliably in real-world operation. Consequently, it is reasonable to consider a range of mechanisms, for protecting differing amounts of information and against monitoring of different kinds. Although channel-based encryption can be helpful, it is not sufficient. This paper considers pursuing different levels of end-to-end protection, referencing examples of component mechanisms that already have encouraging field experience.

Status of This Memo

Copyright Notice

1. Background

Concern for pervasive monitoring motivates the deployment of strong mechanisms that will protect against intrusive disclosure of information. Information needing protection can be primary payload, descriptive meta-data, or traffic-related analysis. Most Internet services operate according to a relatively simple, two-party client/server model, with the server holding primary data and performing primary actions, and the user having a direct relationship with the service being provided. For these arrangements, concerns over privacy violations tend to focus on wiretapping of the data transfer mechanism and on server compromise.

In contrast messaging architectures, such as for email [MAILARCH], can be highly distributed, with any number of application-level store-and-forward intermediaries. This can produce complex sequences through many independent administrative authorities, possibly unknown to either the user or the recipient. Because multi-hop store-and-forward messaging can involve several systems not under the administrative control of either end of the messaging transaction, compromise of any of the intermediate systems can expose messages to monitoring past the first, or before the last, hop. Therefore end-to-end encryption is still highly desirable. Key distribution and validation is one of the greatest impediments to deployment.

Current multi-hop store-and-forward messaging on the Internet uses primarily two security technologies:

1. Channel encryption between the submitter and its submission server and final recipient and its receiving server, respectively, that encryption generally relying on CAs for authentication; and

2. End-to-end content encryption that relies on pre-authenticated certificates available to the end-points.

The former is used, but does not provide sufficient protection against certain kinds of pervasive monitoring, and the latter is rarely used because of deployment and use barriers. More opportunistic mechanisms might have a higher likelihood of deployment, with minimal effect on services, and therefore should be attempted. Further if these opportunistic mechanisms do gain success, they can be used for further minimization of some forms of abuse.

Complete protection against pervasive monitoring (PM), for traffic through complex handling sequences, has not yet been achieved reliably in real-world operation. Consequently, it is reasonable to consider a range of mechanisms, for protecting differing amounts of information and against monitoring of different kinds. The premises are that one or more of these might prove more effective than others and that some protection is better than none.

Given the scale and urgency of community need for this protection, mechanisms should be based on established technologies, where possible. While innovation is needed, it should be kept as modest as possible. So the major challenge should be system design, rather than component invention, where possible and practical.

There are four types of data to be considered for protection in a distributed messaging architecture:

- Message Content

- Header Content

- Envelope meta-data

- Handling meta-data

Message content is considered the primary payload; for email this is the body of the message. However messaging often contains additional content in a header, such as the names and addresses of authors and recipient, content summary, such as a Subject field, date of posting, and so on. Envelope meta-data is the information used by the transit service, including recipient and return addresses. Handling information is created during transit, such as for recording processing tags by intermediaries. The placement of these bits of information can vary, so that distinguishing among them can sometimes be confusing. As an example email relay handling meta-data is placed into the message header.

Almost all efforts to protect messages have focused on the primary message content, with two well-known capabilities being standardized. [OPENPGP][SMIME] However after twenty-five years of these efforts to protect messages that are in transit, nearly all such traffic is still sent in the clear.

In the absence of a success scenario for end-to-end payload privacy protection, it is not possible to be certain which barriers are critical, nor how to overcome them. In current discussions, the primary culprits are believed to be key administration and end user interface design and performance complexity. Both are deemed to require too much human effort, and a common view is to essentially remove humans from needing to configure their services or choose to use them.

Channel encryption is low-hanging fruit when it comes to messaging security, though it only offers minimal protections against pervasive monitoring in its current use. Right now, messaging-related channel encryption is almost exclusively used between end clients and their directly-associated servers, mostly for purposes of protecting the login credentials from monitoring. It does result in clear message contents also being protected from snooping on the channel between the end client and server, and it protects envelope information (which is not otherwise protected by end-to-end content encryption.) However this protection only operates for the first and last message hops and leaves intermediate hops unprotected. So the addition of channel security at every hop is still desirable. Authentication can be recorded in the envelope if it takes place, presumably in a way that allows the recipient to confirm that the authentication took place, but authentication is not necessary for a large increase in security. For intermediate hops opportunistic encryption would be a significant improvement and would be deemed sufficient for most cases. The intermediate servers can simply do key exchange in-band.

2. Incremental End-to-End Protection

Channel encryption can not protect against some of the PM activities that have been documented. So the more challenging concern is protection against collaborating or compromised intermediate nodes and even source and destination servers. Ideally protection therefore must be end-to-end, defined in terms of the author’s and recipient’s independent user agents. The difficulty of achieving this is exacerbated by the degree of existing Internet messaging activity that has all user agent behavior on, or controlled by, end-system web servers, rather than by independent software that is solely under the control of the author or recipient. Hence the best end-to-end protection that will be achievable for many users is between originating server and receiving server.

This highlights the need for incremental mechanisms that provide increasing protection. Greater user independence should be able to permit greater user protection. Another benefit of this incremental approach is that it is likely to provide some useful protection while still permitting exposure information necessary to legitimate management. Of course, balancing between protecting against problematic monitoring and facilitating legitimate monitoring (management) requires agreement on the trade-offs and explicit choices amongst them. The discussion and agreement remain an open and challenging task.

An observation about focusing on PM protection is that use of encryption for that purpose does not necessarily carry the usual, accompanying requirement for strong authentication of one or both principals in the interaction. In the extreme, this might mean that typical man-in-the-middle scenarios are not a concern, but it also can mean that authentication related to an agent -- rather than to the user -- is sufficient.

This well might permit opportunistic privacy protection without direct user involvement, possibly with unauthenticated encryption and no human configuration, and for authentication to take place as a separate piece of user interface when that is desirable. To the extent that human involvement is needed for the basic setup, it might be limited to service administrators, rather than end users. The obvious appeal of this is that there are orders of magnitude fewer administrators than there are users, and administrators typically have far more technical skill.

Key discovery is the most significant challenge during operation of a protection mechanism. A promising approach that already has some field experience achieves key distribution through the [DNS]. The core requirement, of course, is determining what domain name to query. The most obvious choice in a messaging service is the domain name of the recipient’s address. Enhancing this to permit DNS queries on an entire email address would be the refinement to attempt.

A DNS-based mechanism would facilitate query, but would not deal with key administration. Although there is activity in this space, easy key generation remain an open issue for the Internet. However note that by making the critical actors for this service be operators, the scale of this challenge is dramatically smaller than if end users need to be involved.

Given a basic key-discovery ability, the question then is what to encrypt? Simply encrypting a message body is appealing, but leaves exposed all of the message header, as well as associated handling and envelope information. This is where the "levels" reference in the paper’s title comes in. Additional mechanisms or services can protect increasing amounts of message-related information. However, a pragmatic basis for choosing different levels is likely to prove challenging, since users cannot be relied on to make such decision. Still it will be worth pursuing an activity to describe the choices. Essentially, they are:

- Content
- Content + Header
- Content + Header + Envelope

For email, one challenge in encrypting the message header is that the header is modified in transit. A plausible approach is to encapsulate the original message as a [MIME] attachment, so that the visible message header is only a form of envelope.

In order to obscure the origin/receiver envelope information, the message in transit needs to use different envelope data. Given that the information is essential to message transit, this will require an overlay relay service, designed to hide actual author/recipient information. It is worth considering enhancements, to integrate it more seamlessly for well-motivated users.

3. Exemplars to Demonstrate Feasibility

Although it is easy to offer appealing design ideas, estimating their real-world feasibility and utility can be challenging. This paper is not intended to formulate detailed solutions, but it does need to provide some basis for comfort with the basic approaches it suggests. The discussion in this section is therefore intended to provide some substance, to that end.

Rather than consider whether a detail discussed in this section is good or bad, or whether one approach is better or worse than another, the reader is encouraged merely to review the examples in terms of existing deployment experience and the likely pragmatics of incremental engineering and operations that is described. While it is likely that superior designs can be specified, the requirement now is to develop a reasonable degree of comfort that the basic approaches are plausible.

3.1. Administrators vs. Users

There is considerable field experience with the difference between the administrative skills of professional operators, versus end-users. With respect to key administration, specific examples include [DNSSEC] and [DKIM]. The experience shows that key administration tends to be daunting even for professionals, but is infeasible for most end users.

A related point is the greater deployment and use success that is likely when providing protection between servers rather than between end-users. An exemplar of this approach being successful for a security mechanism is [DKIM] as compared against the problematic deployment histories of [OPENPGP] and [SMIME]. However the obvious concern is that the end-users must rely on the safety of their server operations.

3.2. Key Discovery

Key discovery through the DNS already has several examples, including [DNSSEC], [DANE] and [DKIM]. In the aggregate they demonstrate that this basic approach is operationally reasonable.

3.3. Per-User Keys

The history of per-user key administration is particularly disheartening. To the extent that key discovery via domain names has established a strong proof of concept, it is appealing to consider extending it to the granularity of complete email addresses. Although there have been some attempts at doing this, they gained no large-scale traction.

Historically, there has been a basic incompatibility between email address encoding and domain name encoding. A domain name is an undifferentiated sequence, whereas an email address is structured into two, semantically-distinct parts (separated by the "@" sign.) A recent, popular enhancement to DNS naming is the use of an underscore-based node name, such as [SRVDNS] for information that does not need to be treated as a hostname. The application of this enhancement could produce a query name in the style of:

```
Mailbox._at.example.net
```

Hence, key query would be for a domain name, where the name might be a hostname or might be an encoded email address. Although this would be a new mechanism, it entails no enhancement to infrastructure services and it re-uses a well-established and reasonably inexpensive form of DNS-based mechanism.

3.4. Message Encapsulation

Protecting the message header means that it needs to be hidden during transit, in spite of the header’s being modified in transit, for email. One approach to solving this is to encapsulate the entire message as a MIME attachment; the visible header therefore would only contain handling information. This model of encapsulation only requires adoption by author (or originating server) and recipient (or receiving server) and is transparent to the message-handling infrastructure. Architecturally, it is identical with the way MIME was propagated, in the 1990s, so it's viability has been well demonstrated. Also, encapsulating an entire message as an attachment has already been enabled through [BSMTP].

3.5. Protecting Envelope Meta-Data

If envelope data is to be hidden during transit, it must be encapsulated in a message with different envelope data, and processed by special, trusted relays that hide addressing and transit information, and ensure that none is associated with the message when it is finally delivered. This is in the spirit of [TOR].

4. Security Considerations

Everything in this draft related to security, and especially to confidentiality in the service of privacy protection.

5. IANA Considerations

There are no IANA considerations for this draft.

Note to RFC Editor: Please remove the entire IANA Considerations section, prior to publication

6. References - Informative

[BSMTP] Freed, N., Newman, D., Hoy, M., and , "The Batch SMTP Media Type", RFC 2442, November 1998.

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

[DKIM] Crocker, D., Hansen, T., and M. Kucherawy, "DomainKeys Identified Mail (DKIM) Signatures", RFC 6376, September 2011.

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

[DNS] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[MAILARCH]
Crocker, D., "Internet Mail Architecture", RFC 5598, July 2009.

[MIME] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

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

[SMIME] Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, January 2010.

[SRVDNS] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[TLS] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[TOR] "TOR Project", WEB https://www.torproject.org/, .

Authors' Addresses

Dave Crocker
Brandenburg InternetWorking
675 Spruce Drive
Sunnyvale, CA 94086
USA

Phone: +1.408.246.8253
Email: dcrocker@bbiw.net

Pete Resnick
Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego, CA 92121
US

Phone: +1 858 6511 4478
Email: presnick@qti.qualcomm.com