Network Working Group
Internet Draft
<draft-levine-smtp-batv-01>
Intended status: Standards Track
Expires: November 2008

J. Levine
Taughannock Networks
D. Crocker
Brandenburg InternetWorking
S. Silberman
Openwave
T. Finch
University of Cambridge
May 2008

Bounce Address Tag Validation (BATV)

draft-levine-smtp-batv-01

Status of this Memo

This Internet-Draft will expire in November 2008.

Copyright Notice

Abstract

The envelope of Internet mail contains an RFC2821.MailFrom command, which may supply an address to be used as the recipient of transmission and delivery notices about the original message. Existing Internet mail permits unauthorized use of addresses in the MailFrom command, causing notices to be sent to unwitting and unwilling recipients. Bounce Address Tag Validation (BATV) defines an extensible mechanism for validating the MailFrom address. It also defines an initial use of that mechanism which requires no administrative overhead and no global implementation.

This document is a revision of draft-levine-mass-batv-02.

Table of Contents

1. Introduction

The envelope for Internet Mail may contain an address that is designated to receive transmission-related notifications. It is specified in the RFC2821.MailFrom command. The field is set by the RFC2822.Sender, acting as an agent of the message author specified in RFC2822.From. However no portion of the MailFrom address is required to have any similarity to any portion of the From or Sender addresses, and valid usage scenarios do call for the MailFrom address to have no visible relationship to the From or Sender values.

Further, existing Internet mail permits unauthorized use of addresses in the MailFrom command, which results in having notices sent to unwitting and unwilling recipients. Therefore, the challenge is to distinguish legitimate uses from these unauthorized uses and to do this with a mechanism that incurs modest administration, operations and performance costs.

Bounce Address Tag Validation (BATV) defines a framework for mechanisms that validate the value in this command. Multiple validation methods are envisioned. So BATV defines a common syntactic framework that enhances the local-part field of the MailFrom address. An initial, specific validation scheme is also defined; it requires no administrative overhead and no global implementation.

The <local-part> of an Internet mail address is a globally opaque string. Hence, the specified modification to the local-part can be deployed in a manner that is entirely transparent to the public Internet mail service, except for mail system components within the scope of the MailFrom domain, and then only for components that process the MailFrom address local-part. The result permits the MailFrom target domain to distinguish notification message addresses that are valid from those that are not. Enhancements would permit processing agents that are along the original message's transfer path to determine whether the MailFrom adress is likely to be valid. This assessment could aid in deciding whether to send a bounce message, thereby reducing the Internet mail infrastructure cost for transmitting notification messages in response to addresses used without permission. It might even be used to detect invalid messages, thereby reducing Internet mail infrastructure cost for original messages.

Terminology: Terminology conforms to [I-D.email-arch]

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 [RFC2119]

2. Model

BATV defines a method for tagging information to be included in the <local-part> of the RFC2821.MailFrom address. This permits encoding information that authenticates the MailFrom. Because the information is placed in MailFrom, rather than in an RFC2822 header, it sometimes is not as publicly visible as an RFC2822 header. Tagging the MailFrom address rather than any of the RFC2822 addresses avoids problems arising from rewriting message headers that may be visible to recipients, and enables the validation process to operate within an SMTP session before the contents of a message are transferred.

2.1 Meta-Syntax

BATV tagging is based on a meta-syntax that defines a field-oriented structure for an address local-part. It permits use of a variety of address authentication methods, while supporting remote extraction of the core portion of the local-part, without having to understand the semantics of any particular scheme.

NOTE: BATV is for the purpose of detecting invalid RFC2821.MailFrom addresses. Any BATV-related modifications that are made to the original MailFrom MUST preserve the result of returning valid bounces to the address originally specified in that MailFrom.

The meta-syntax for MailFrom local-part is defined in Section 3.

2.2 Tagging Schemes

BATV permits alternative schemes. To ensure interoperability among independent participants, other specifications adopting the meta-syntax conventions MUST define and register with IANA a unique, case insensitive <tag-type> element, to identify the specific mechanism that is being used for MailFrom validation.

Private Tagging: If MailFrom validity assessment is performed only within the scope of the domain referenced in the MailFrom address, then its semantic scope is private (closed), encompassing only that domain and the one that generated the validity information. To the rest of the Internet, the tag information is opaque, like a cookie. In these situations, the closed system is free to use any tagging scheme it deems helpful, although a standard format aids other systems that wish to avoid re-tagging addresses that are already tagged, or to strip off the tag for compatibility with legacy systems that key on the MailFrom address of incoming mail. A simple scheme for this is defined in Section 4.

Public Tagging: Using a public-key approach for signing the MailFrom's local-part permits intermediaries that process the envelope to validate that address. For example, an intermediary (that otherwise might create a bounce message) would be able to decide that the MailFrom address use is not valid, so they might decide to terminate bounce processing. Such a scheme might use the BATV meta-syntax in the following way:

```
pub3=<crypted>=<loc-core>@example.com
```

If the creator of a bounce could make this assessment, all of earlier intermedite MTAs also could. Hence, every MTA would be able to assess whether a message has an unauthorized RFC2821.MailFrom.

Unfortunately, none of the multiple existing public key services has yet gained wide adoption. Therefore, this specification is not able to provide a single method for public MailFrom validity checking.

2.3 Beyond BATV

BATV defines a framework that retains the original local-part of the MailFrom address within the BATV-encoded form. This permits external inspection of the original local-part, such as for analyzing its use with respect to particular RFC2822.From addresses. Enhancements that go beyond the open information of BATV might replace the original local-part with some form of translation. Examples of such schemes could include:

Alias: The original RFC2821.MailFrom local-part could be replaced with an alternative local-part. The meta-syntax provides a way to flag the difference between the new local-part and the original.

Opaque Pointer: This could be used to consult a database with records of mail sent and bounces received.

Challenge Response: The receiver could make a DNS-query for instructions about processing the RFC2821.MailFrom bounce address.

2.4 Operation

The basic methods for creating and interpreting BATV-encoded MailFrom addresses are very simple.

2.4.1 Tag Creation

The RFC2821.MailFrom address is specified by the RFC2822.Sender. This makes the MailFrom address an end-user string, created by the oMUA or MSA. However it is entirely reasonable to have an outbound MTA, under administrative control of the Sender's domain, perform the necessary signing. What is significant is that this requires a change to only two modules, one in the outbound sequence and one in the corresponding inbound sequence. The change is transparent to all other systems components that transmit the message.

NOTE: If a MailFrom local-part already conforms to the meta-syntax, the string SHOULD be left unchanged, so as not to break forwarding.

NOTE: An MTA MUST ONLY tag addresses in domains whose inbound MTAs can validate the tags. In particular, when an MTA is relaying a message, on behalf of another Administrative Management Domain (ADMD), it must not tag the MailFrom address, even if the original ADMD did not add a tag. In all cases, the MTA must only tag addresses for which it has access to the signing key that

corresponds to the validation key used by the inbound MTA for the address' domain.

2.4.2 Tag Interpretation:

Addresses that contain BATV tags can be interpreted for two different purposes: bounce address validation and bounce delivery.

Address validation:

An MTA MAY validate a BATV-encoded MailFrom address. This requires that the MTA be able to process the specific BATV validation scheme that is specified by the <tag-type> field. If the address is determined to be invalid, the MTA SHOULD process the address as having a permanent failure, for example by returning a 550 response to the SMTP command containing the address.

The MTA MAY also require that the use of the address is appropriate, for example that the message is a bounce as indicated by a null RFC2821.MailFrom; other heuristically determined contexts MAY also be appropriate. For example, messages with MailFroms beginning with "mailer-daemon@" are in practice almost always bounces. Use of a BATV address in inappropriate contexts SHOULD cause a permanent failure as above.

Bounce delivery:

When an MTA within the specified address delivery domain's administration receives a delivery notification directed to a BATV-encoded address, the MTA SHOULD validate that address when that message has a null MailFrom. A receiving server MAY also perform heuristic selection of other incoming mail, such as ones that have a MailFrom starting with "mailer-daemon@". If it determines that the use is not valid, it SHOULD reject the message during the mail transfer connection, such as with SMTP.

If the BATV address passes these checks, the message SHOULD then be delivered to the original RFC2821.MailFrom address. This original MailFrom address would be recovered as a side-effect of validating the BATV address.

3. Local-Part Meta-Syntax

A meta-syntax for the <local-part> of an address creates a public convention for partitioning an address' local-part field (left-hand side) into sub-fields of attributes associated with the <addr-spec> that was the original local-part.

A standardized meta-syntax for local-part permits attributes to be present in the address, without requiring that public processing of the address have any understanding of the attributes' semantics. The semantics of <local-part> are strictly local to the domain administering the <local-part> field. This separation between local and global semantics has been a powerful benefit to Internet mail. It affords considerable operational flexibility. The meta-syntax permits public information in an address to be richer, while maintaining the local/global separation.

The generic element syntax for the structured fields defined for a BATV <local-part> is:

```
local-part       = tag-type "=" tag-val "=" loc-core

tag-type         = 1*( DIGIT / ALPHA / "-" )
                   ; specific, registered validation scheme

loc-core         = {original local-part value}

tag-val          = 1*( DIGIT / ALPHA / "-" )
                   ; the validation data
```

This syntax is chosen so that software that needs, for legacy compatibility reasons, to recover the original bounce address can do so by checking for the presence of the tag-type, and if it is present, discarding the local-part up through the second equal sign.

4. Simple Private Signature (prvs)

The Simple Private Signature (PRVS) scheme signs the original MailFrom by using a simple shared-key to add a hash of the address and some time-based randomizing information.

4.1 Syntax

This scheme is identified as:

```
tag-type       = "prvs"
                   ; simple private signature

tag-val        =  K DDD SSSSSS

K              =  1DIGIT
                   ; key number, to allow key rotation

DDD            =  3DIGIT
                   ; day number, low three digits of
                   ; the number of days since 1970
                   ; when the address will expire

SSSSSS         =  6HEXDIG
                   ; hex of the first three bytes of the
                   ; SHA-1 HMAC of <hash-source> and a key

hash-source    =  K DDD <orig-mailfrom>

orig-mailfrom  =  <original RFC2821.MailFrom address>
```

4.2 Operation

4.2.1 Signature Creation

PRVS creates a package around an existing <local-part>, comprising the PRVS label and the signature hash on the left. The hash is extremely simple and not very robust, because the requirements for BATV do not entail strong protection. The mechanism provides very weak protection against replay, in order to keep the effort to create or validate the signature small.

4.2.2 Signature Checking

The checking of private signatures is only performed within the domain specified in the MailFrom command. The first component that processes the MailFrom's local-part must be able to interpret the meta-syntax. It MAY also perform validation.

The scheme described here permits algorithmic validation. It does not require maintaining a database of information about recently sent messages.

The DDD part of the <tag-val> allows a domain to limit the lifetime of PRVS addresses to give very basic protection against replay attacks. If the expiry time has passed the address SHOULD be considered invalid even if the HMAC is OK. The address lifetime SHOULD be 7 days, to allow for long delivery delays before a bounce occurs. Since it is valid and often useful for a single message to provoke multiple bounces, it is specifically not a goal of BATV to prevent them. Note that the DDD

is the low three digits of the day number, so comparisons MUST use unsigned subtraction mod 1000 or the equivalent to handle wraparound correctly.

5. Interoperability

BATV seeks to retrofit a standardized syntactic structure onto the <local-part> of an RFC2821.MailFrom email address. Although it is based on an existing, standard structure, it will be used in new environments. Because this field has previously been opaque to these environments, it is likely to create some usage problems with some existing services. Problems are most likely in some services that operate in the scope of the delivery stage of processing, rather than in intermediaries between independent user services. In particular serious problems are likely to be with third-party services that constrain local-part beyond the Internet standards. Hence they restrict interoperability, even without concern for BATV.

As an example, such systems incorrectly identify the sender of the message by using the MailFrom address, rather than the RFC2822.Sender address. Examples are listed below. Further, they require that this address be the same for all future postings from the RFC2822.From address. Problems arise because messages authored by a particular RFC2822.From address are like to vary the associated MailFrom address over time, particularly when BATV encoding is used.

Such systems SHOULD fix the underlying problem, at a minimum by using the RFC2822.Sender address to identify the sender. However, note that Internet mail does not require that the value of the Sender address be related to a From address, and there are many legitimate reasons for it to vary.

Some systems MAY continue to require correlation between MailFrom and From. For example the system might operate on the envelope before the message data has been transmitted, so software might strip off the meta-syntax to recover the <loc-core> which can then be used as the MailFrom address's original <local-part>. For such validation processing this altered address MUST NOT be used for further mail-delivery processing. Rather the MailFrom string MUST be preserved as it was received.

The benefit of a standardized meta-syntax for adding validation attributes is that it permits such mechanisms to detect the "attribute" portions of the local-part and extract only the core portion, without having to understand any of the details of the attributes.

The known and likely set of problem third-parties are:

Greylisters:

A correct BATV implementation will only result in routine delays in this case. However the result of BATV tagging MUST be a constant local-part, for a given message, and not (say) created at delivery time such that each retry gets

	a different validation string, which would prevent it from ever getting through to a greylisting site.
Mailing Lists:	BATV will cause problems with some mailing lists that identify posters by their bounce address. The list will not recognize the identical MailFrom addresses, because it will interpret the differing BATV attributes as part of the address. These services will either reject postings or pass them all to the moderator.
Challenge-Response Systems:	The problem with these is similar to the those with mailing lists, but the challenged user will have to take special action for every message recipient that auto-sorts mail by bounce address.
Sorting and Duplicate Detection:	Any system that sorts by

bounce address (MailFrom) will interpret the addresses as different, even though they are not. This may include whitelisting services.

BATV requires that the sending and receiving mail software within a domain share the secret key used to create the signature. Usually this is easy to arrange, by creating the signature in a domain's outgoing mail relay and checking it in the inbound MX, if both are run by the same management. But it is not necessary for a domain's inbound and outbound relays to be under the same management; for example it is fairly common for incoming mail for a small business domain to be received by an MTA run by a hosting company, while the outbound mail is sent through the ISP that provides the connection to the company's office. In this case, it may be necessary to sign the outgoing mail in the individual senders' MUAs, to check the signature in the individual recipients' MUAs, or both.

6. Security Considerations

This entire document pertains to the security of email's asynchronous error handling (bounce notification) mechanism, by describing a way to differentiate between valid and invalid bounce addresses. This document does not directly provide a mechanism for authenticating RFC2821.MailFrom addresses at intermediate MTAs. The ability to perform validation across the entire transfer sequence is possible if a standardized public key scheme is defined.

The PRVS scheme described here provides minimal protection of the RFC2821.Mailfrom against forgery, with detection possible at the target (delivery) domain. The scheme does not attempt to protect against a replay attack in which a valid, signed MailFrom is used but the message contents are replaced. The same will be true for any other BATV scheme that does not include some link with the message data; however such protection is only reliable for the recipient of the original message, because the integrity of the link will often be broken when the original message data is mangled into the bounce.

There are two common forms of email address forgery: guessing (e.g. attaching common <local-part>s to a domain) and harvesting (e.g. from the web or usenet). Cryptographic BATV schemes make guessing attacks unfeasibly difficult; however these are relatively minor compared to replay attacks, which deserve closer attention.

MailFrom addresses are not usually exposed in the places from which addresses are usually harvested. Many mailing list systems archive messages sent to a list on the web; however they usually replace the original MailFrom address with one that refers to the mailing list manager. So this case is generally not a problem, although there are exceptions. There are other instances of systems that archive email publicly without altering the MailFrom address, such as bug tracking systems; these are a problem.

A proportion of forgeries are caused by mass mailing viruses. Unlike spammers, these have access to private email stores and are therefore more likely to be able to find and replay BATV addresses. For that matter, they can generate MailFrom addresses that are entirely valid.

The PRVS scheme includes a modest protection against replay attacks, by virtue of its using an expiry time, which prevents very old addresses from being used by attackers. It does not prevent replay attacks of young addresses.

7. References

7.1 References - Normative

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

7.2 References - Informative

[I-D.email-arch] Crocker, D., "Internet Mail Architecture", May 2004.

Authors' Addresses

John Levine
Taughannock Networks
PO Box 727
Trumansburg, NY 14886
EMail: standards@taugh.com

Dave Crocker
Brandenburg InternetWorking
675 Spruce Drive
Sunnyvale, CA 94086
USA
Phone: +1.408.246.8253
EMail: dcrocker@bbiw.com

Sam Silberman
Openwave
EMail: sam_silberman@openwave.com

Tony Finch
University of Cambridge
Cambridge, CB2 1TN
UK
EMail: dot@dotat.at
URI: http://dotat.at

A. Acknowledgements

This specification was greatly improved by the extensive participation of John Leslie and Douglas Otis, in early design discussions.

B. IANA Considerations

It may be desirable to establish a registry of BATV tagging schemes and tag types.

Full Copyright Statement

Intellectual Property

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[1].

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

[1] mailto:ietf-ipr@ietf.org